

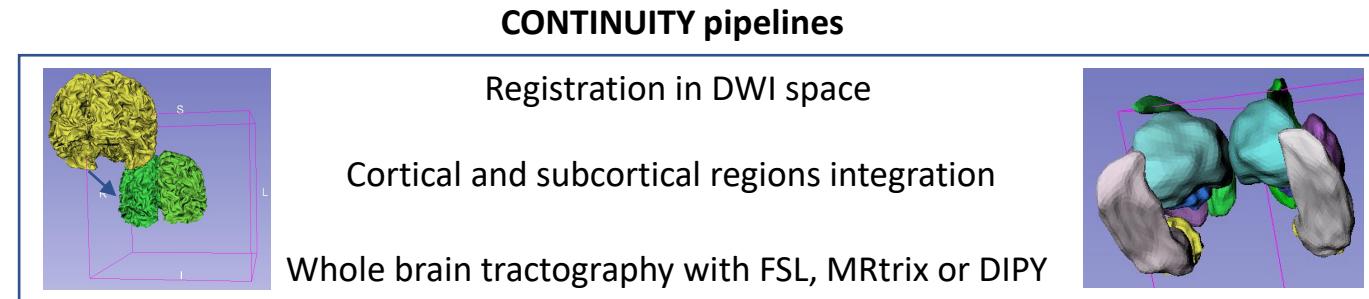
Tutorial  
**CONTINUITY**

**CONnectivity Tool**  
with **INtegration of sUb cortical regions,**  
**regIstration and visualization of**  
**TractographY**

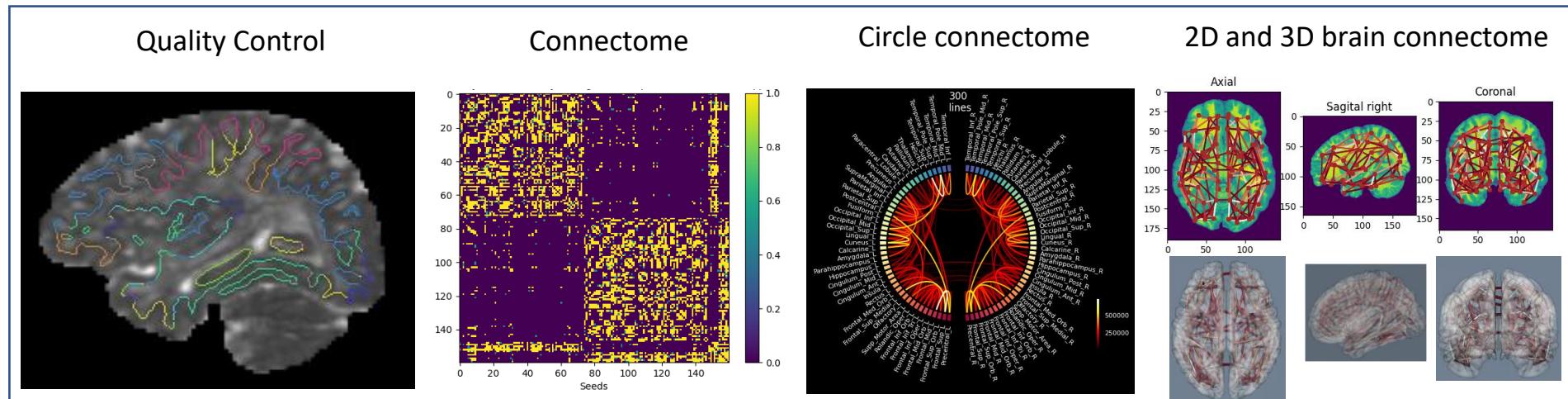
# CONTINUITY:

## CONnectivity Tool with INtegration of sUb cortical regions, registration and visualization of TractographY

Two components:



### CONTINUITY Visualization



# Summary

## CONTINUITY tractography

1. Registration and subcortical regions integration pipeline
2. Tractography pipeline
3. Run application

## CONTINUITY visualization

1. Quality Control with Slicer
2. Connectivity matrix and circle connectome
3. Brain connectome 2D and 3D

# Tutorial **CONTINUITY**

CONTINUITY tractography

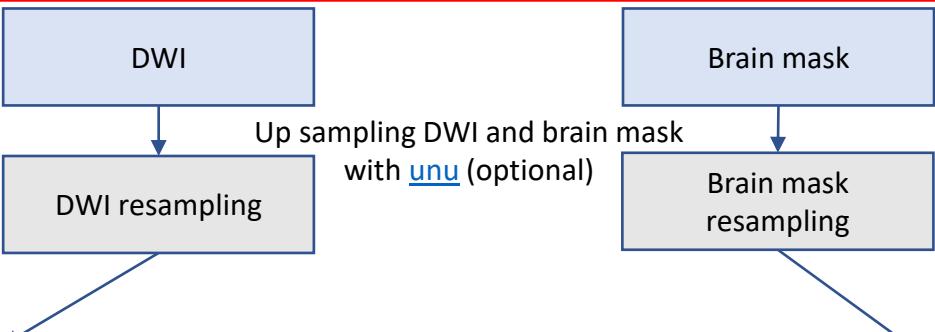
1. Registration  
and subcortical regions  
integration pipeline

2. Tractography  
pipeline

3. Run application

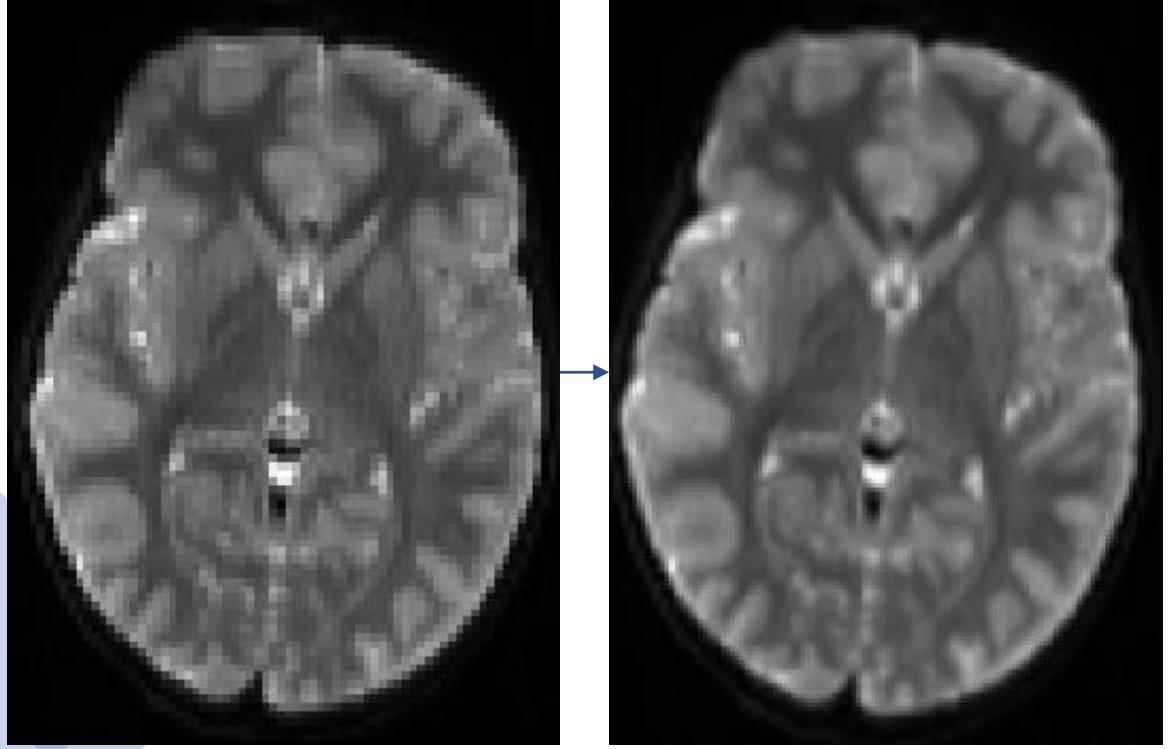
# 1. Registration pipeline: register T1 in DWI space

## Step 1: Preprocessing

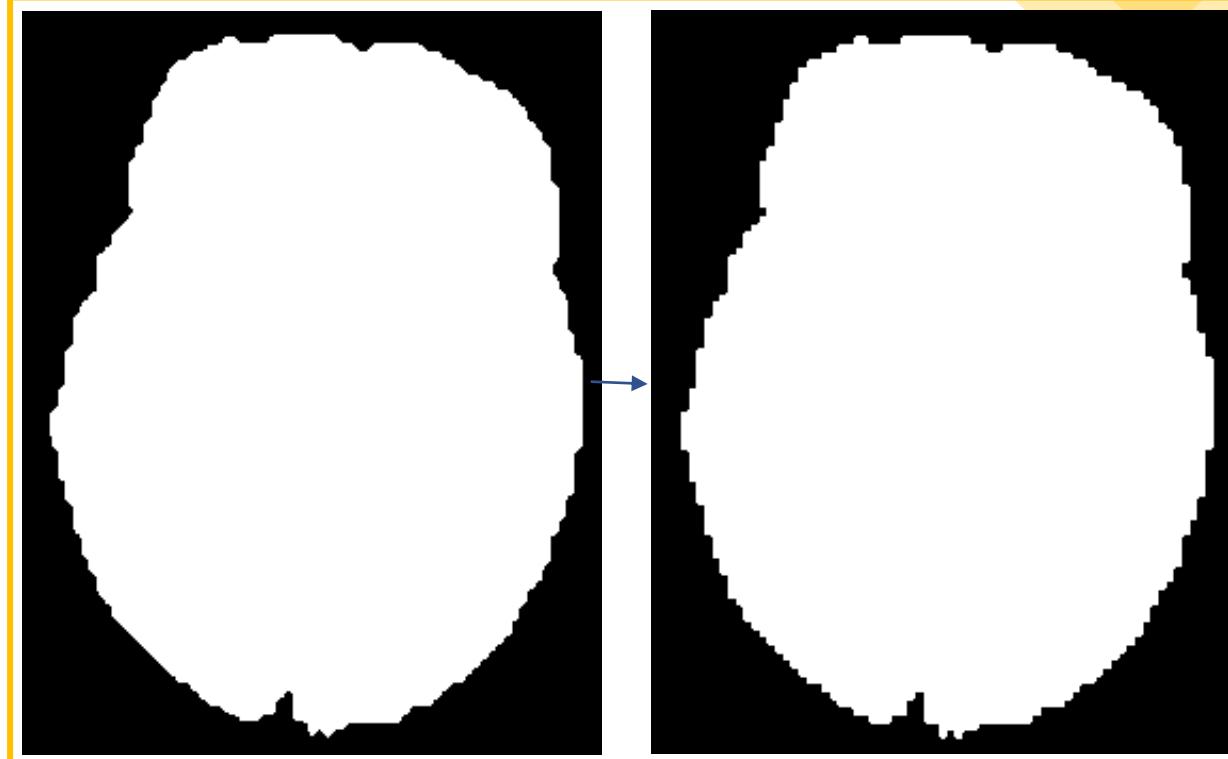


 Input data  
 Intermediate data  
 Key data

Up sampling DWI:

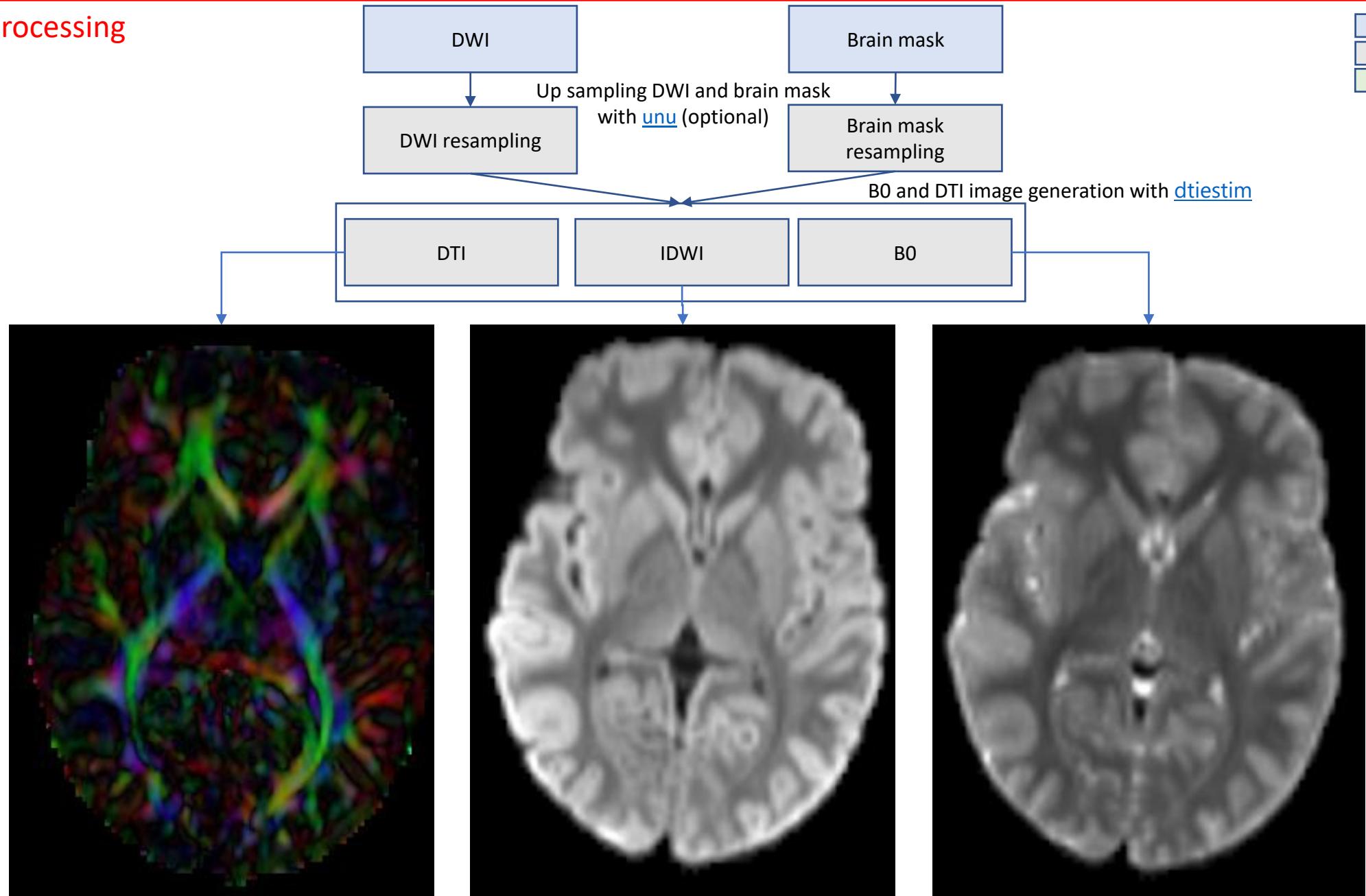


Up sampling Brain mask:



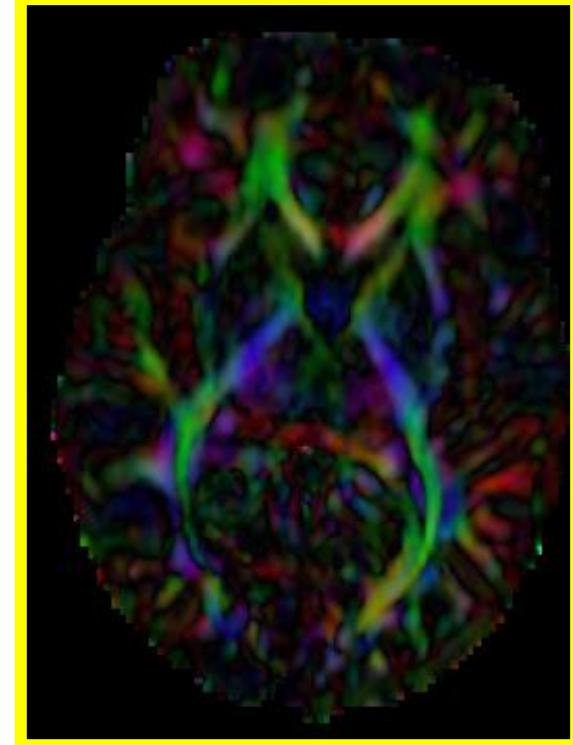
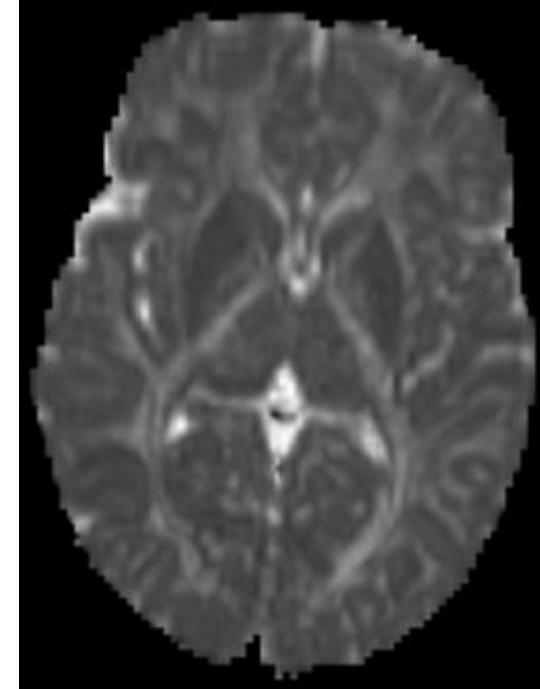
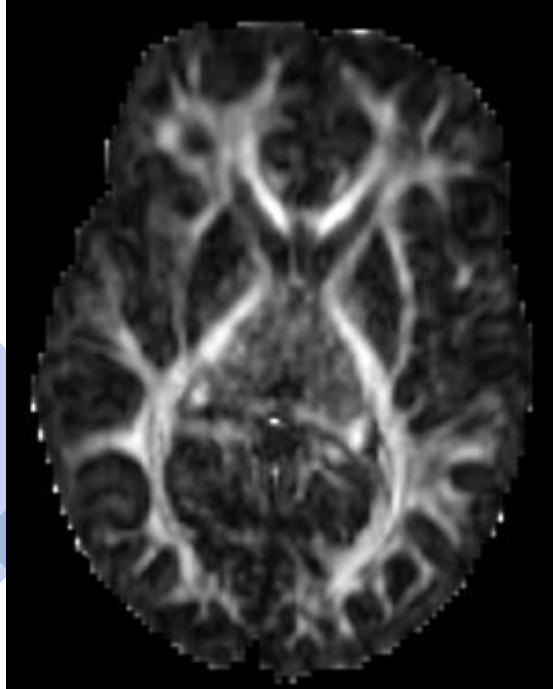
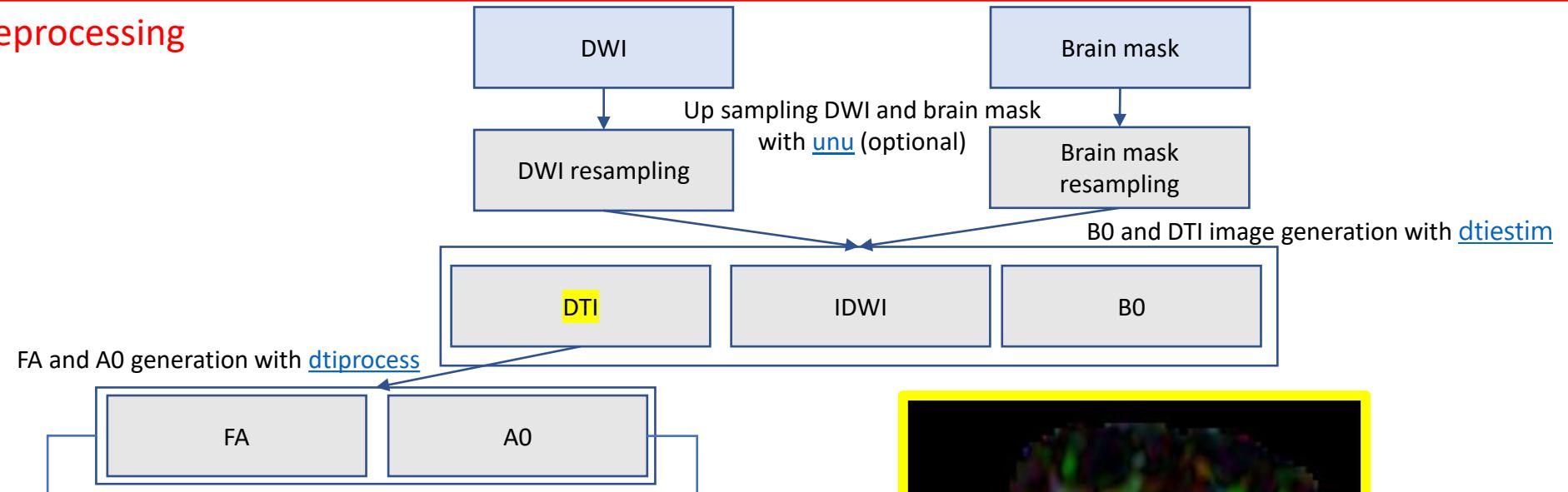
# 1. Registration pipeline: register T1 in DWI space

## Step 1: Preprocessing



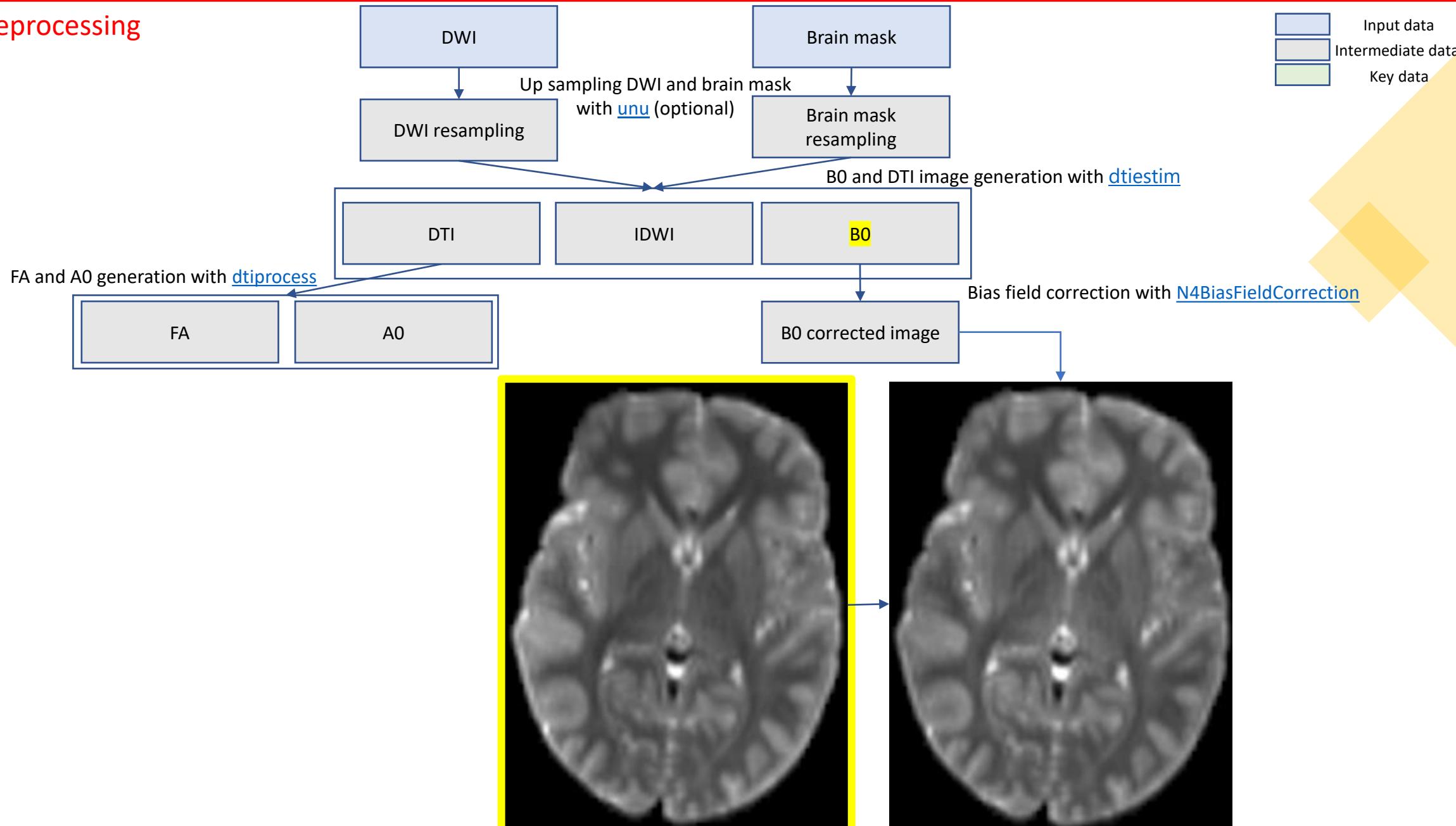
# 1. Registration pipeline: register T1 in DWI space

## Step 1: Preprocessing



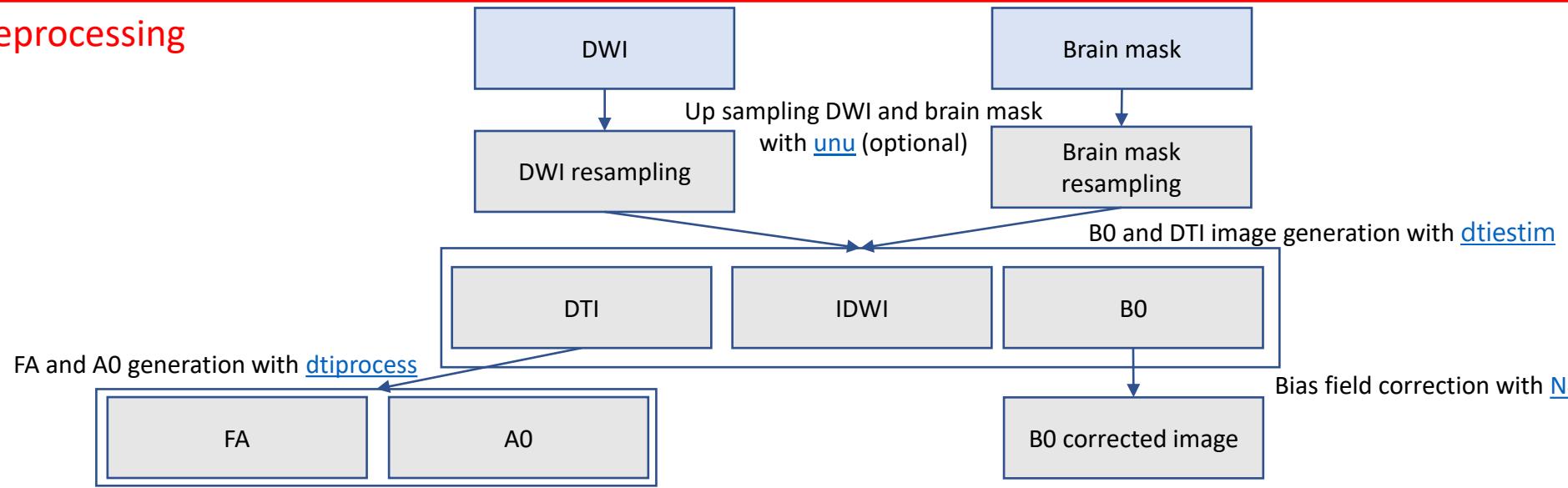
# 1. Registration pipeline: register T1 in DWI space

## Step 1: Preprocessing



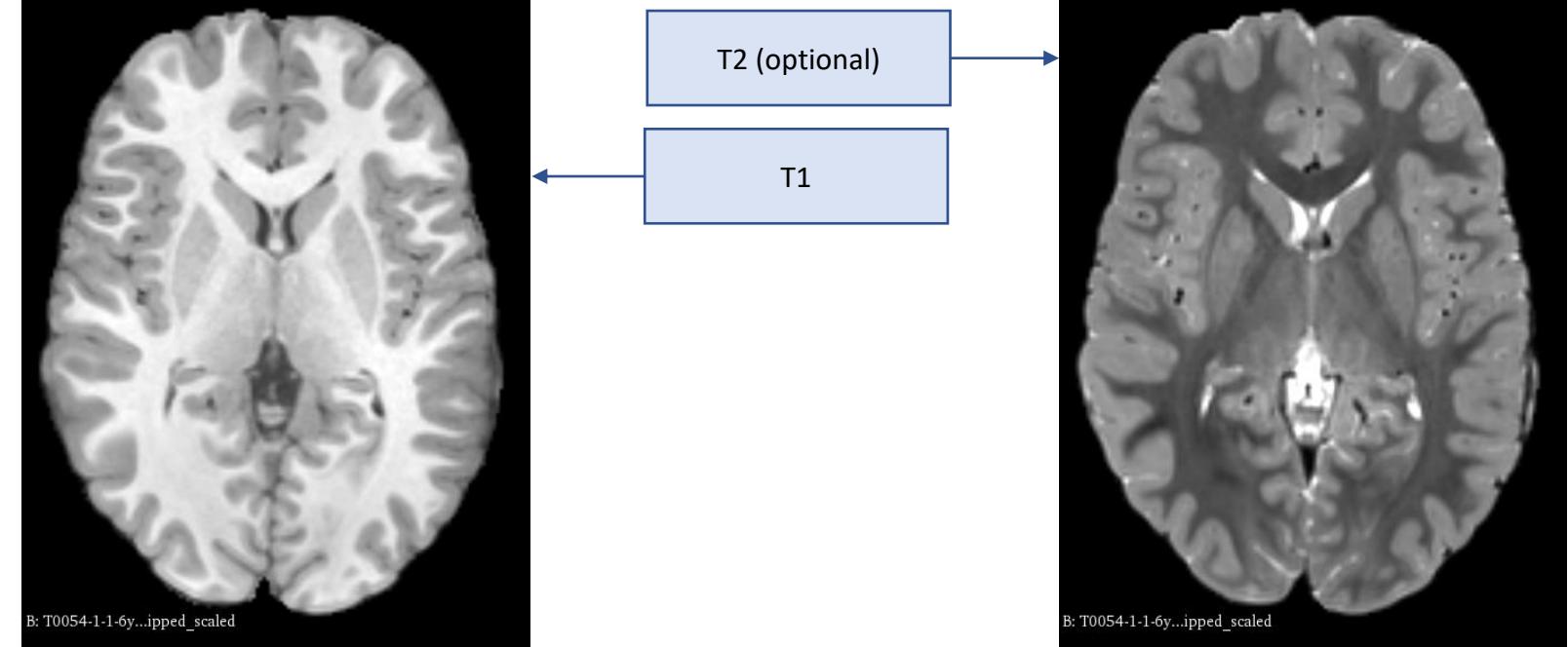
# 1. Registration pipeline: register T1 in DWI space

## Step 1: Preprocessing



  Input data  
  Intermediate data  
  Key data

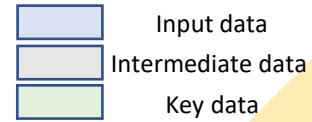
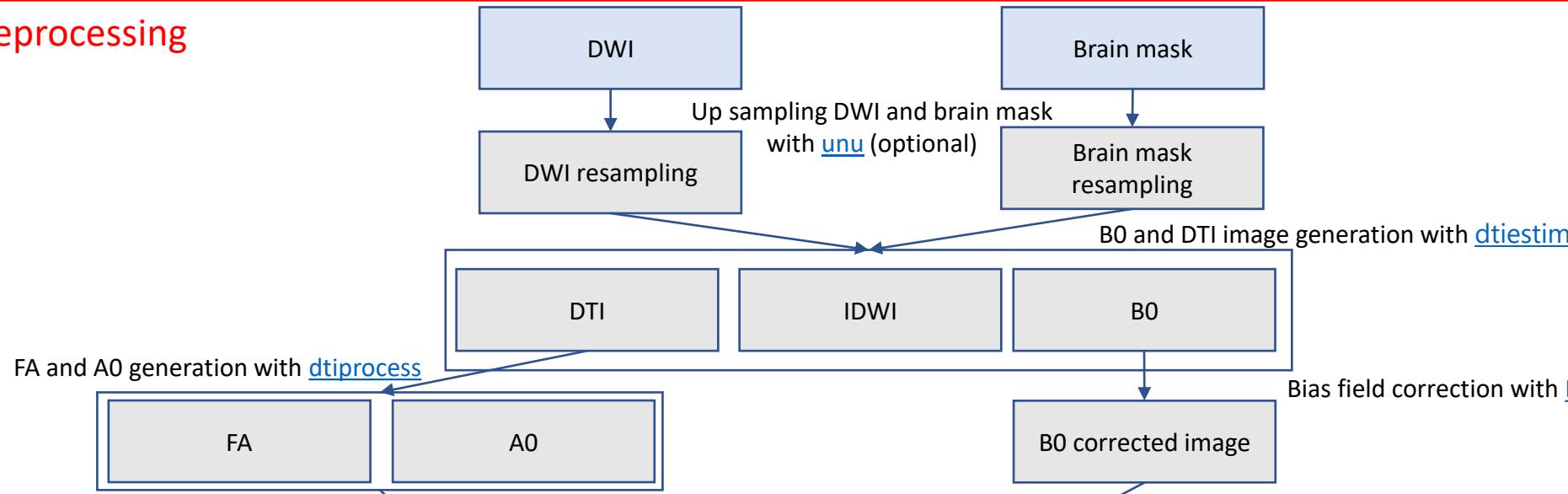
## Step 2: Register T1 in DWI space



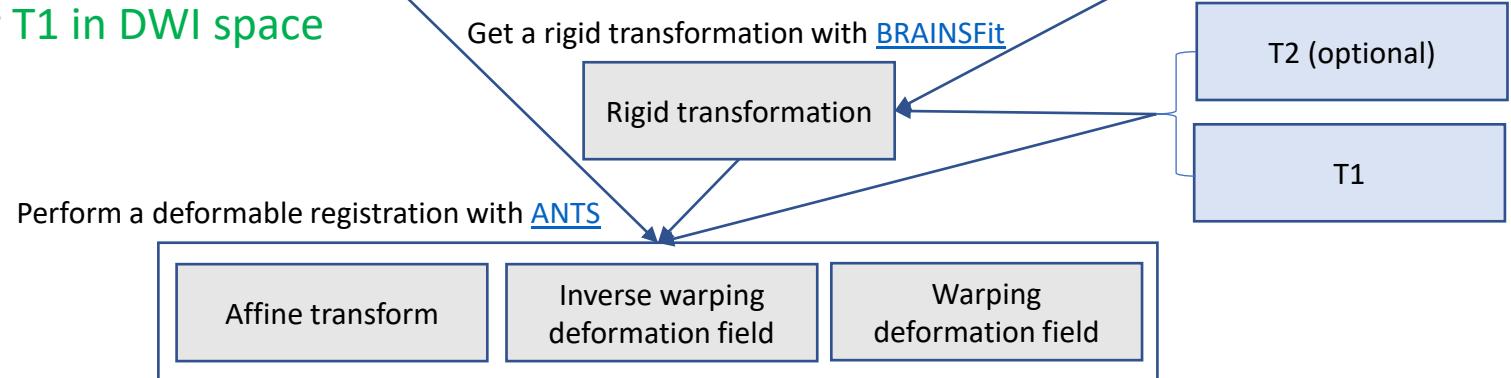
# 1. Registration pipeline: register T1 in DWI space

CONTINUITY

## Step 1: Preprocessing

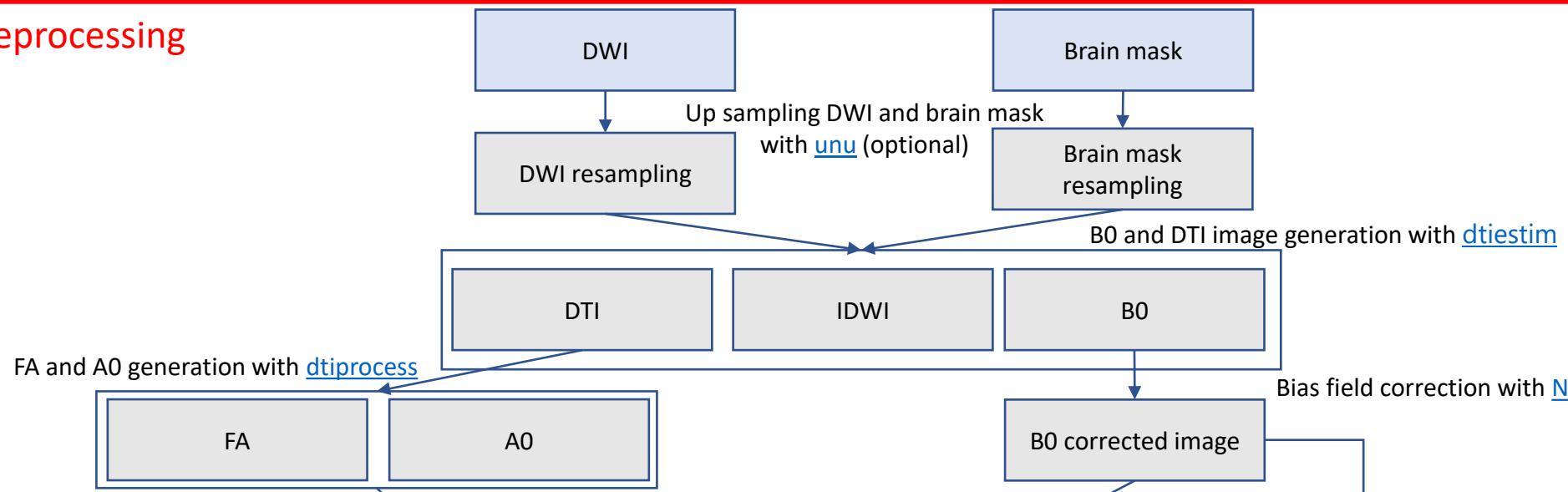


## Step 2: Register T1 in DWI space

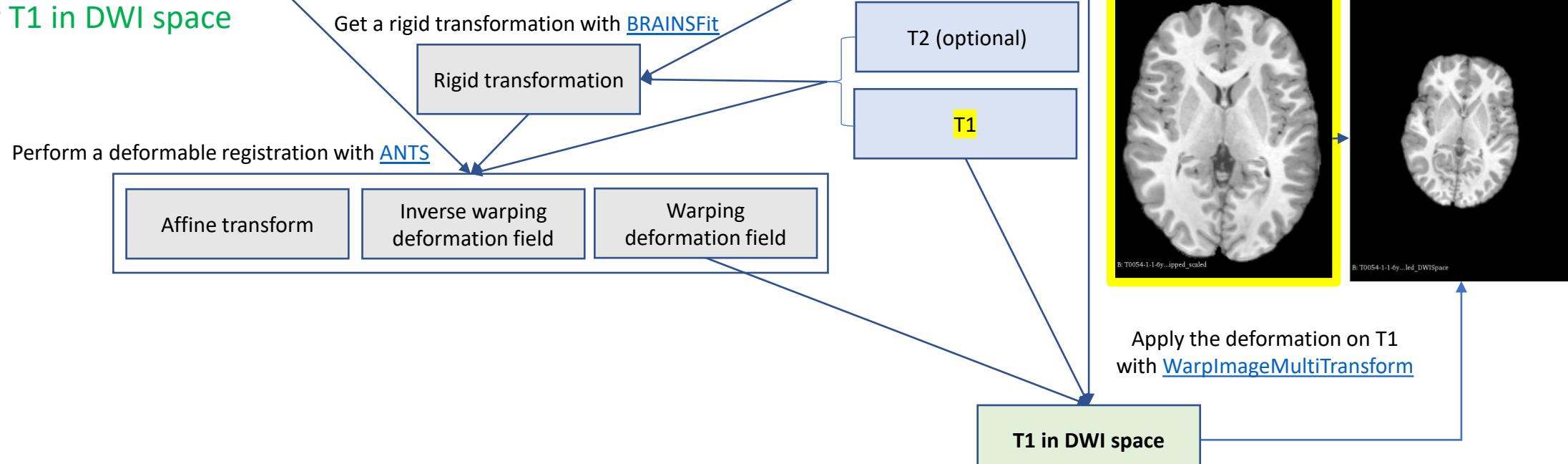


# 1. Registration pipeline: register T1 in DWI space

## Step 1: Preprocessing

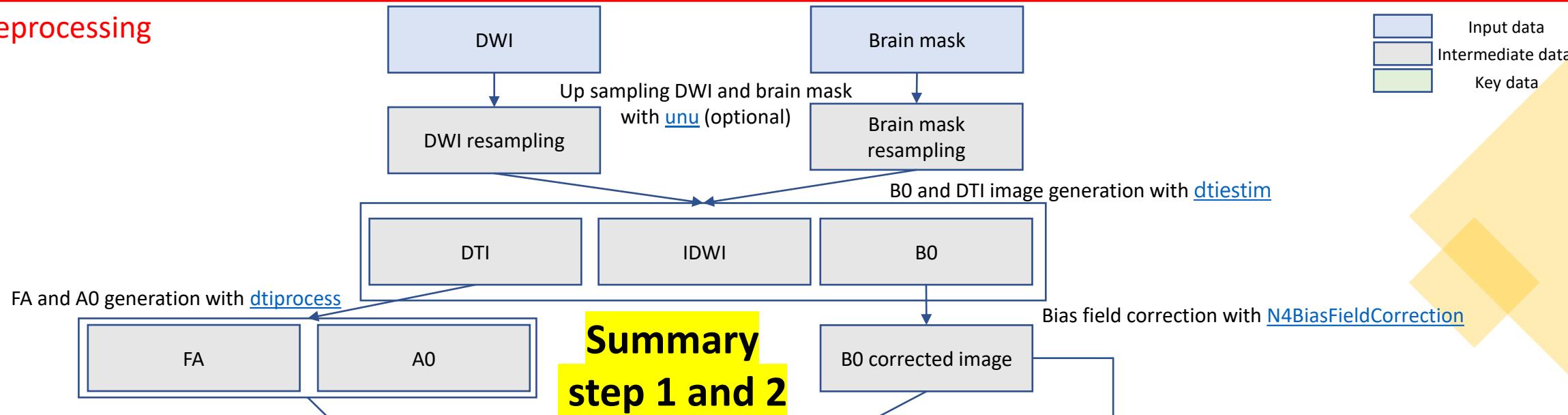


## Step 2: Register T1 in DWI space

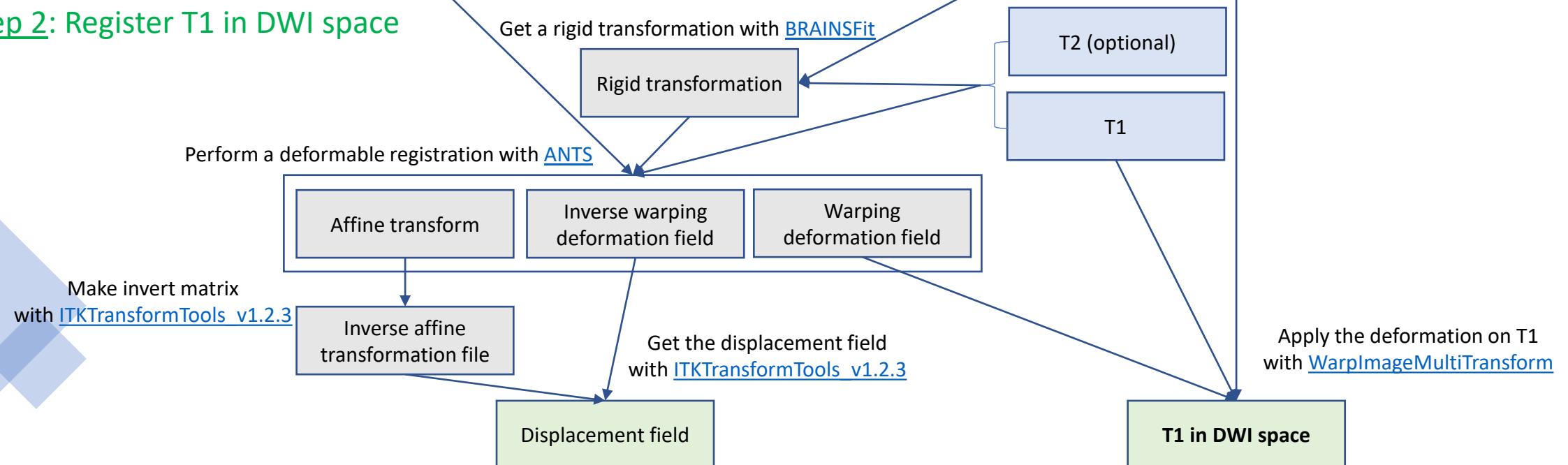


# 1. Registration pipeline: register T1 in DWI space

## Step 1: Preprocessing



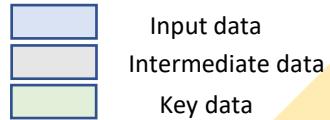
## Step 2: Register T1 in DWI space



# 1. Registration pipeline: cortical and subcortical regions integration

CONTINUITY

## Step 3 : Preprocessing for subcortical regions integration



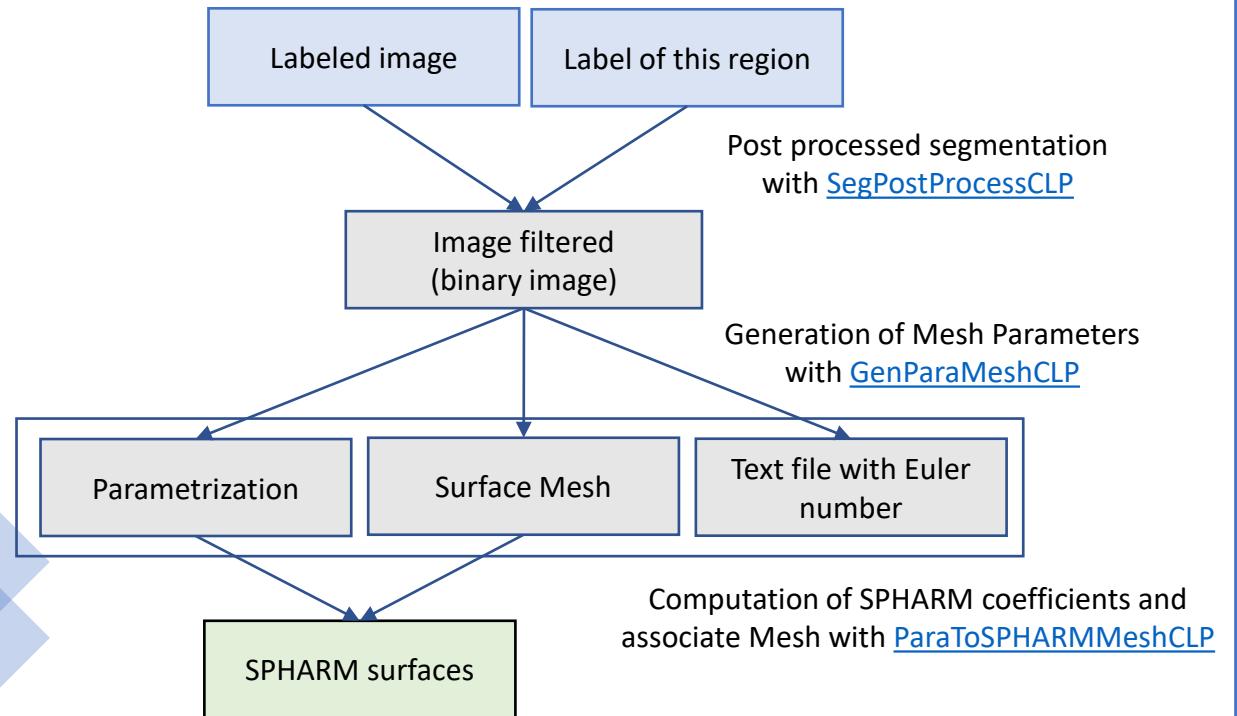
Two different input data options to integrate subcortical regions:

Option 1: SALT (SPHARM surfaces) and KWM folder provided by the user

Option 2: SALT and KWM files computed by CONTINUITY:

For each subcortical region:

### 1- Computation of SALT files



### 2- Write KWM files

For each subcortical region, write a text file with lines containing the label of this region ( number of line = number of points )

```
NUMBER_OF_POINTS=1002
DIMENSION=1
TYPE=Scalar
11176
11176
11176
11176
11176
11176
11176
11176
11176
11176
11176
11176
```

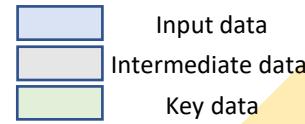
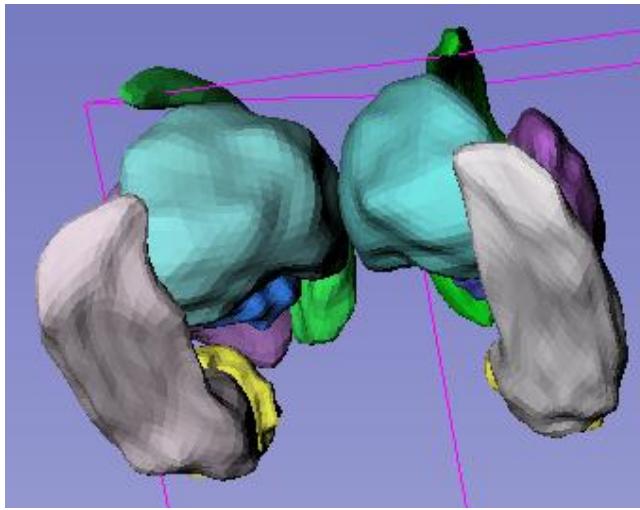
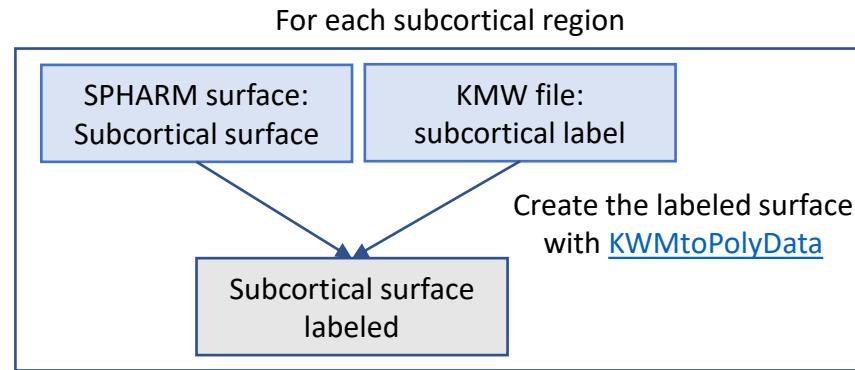
Extract of one of the .txt file

# 1. Registration pipeline: cortical and subcortical regions integration

CONTINUITY

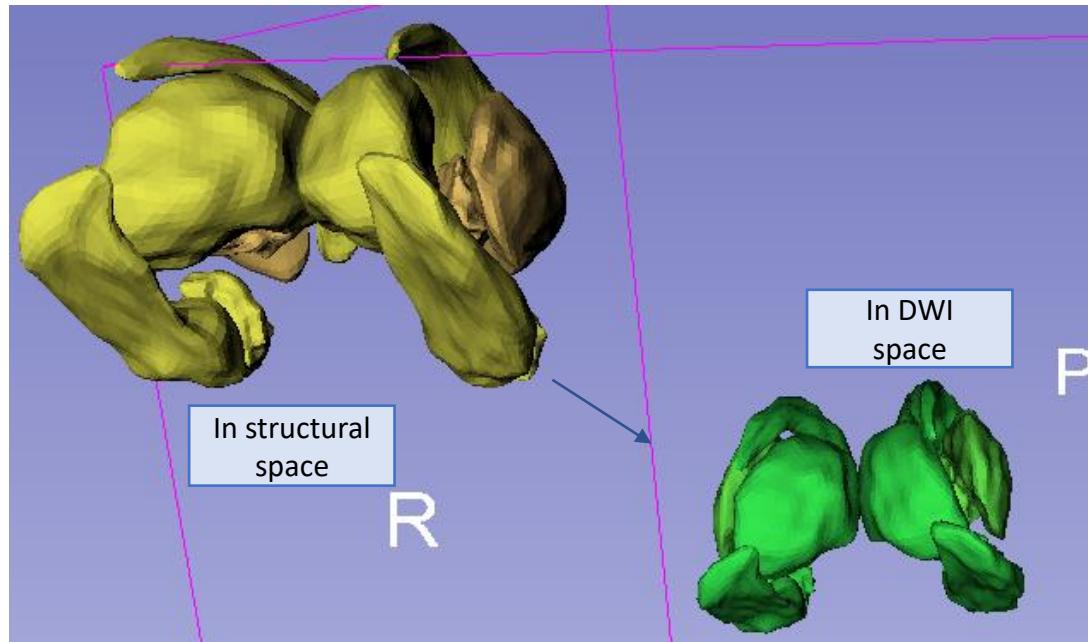
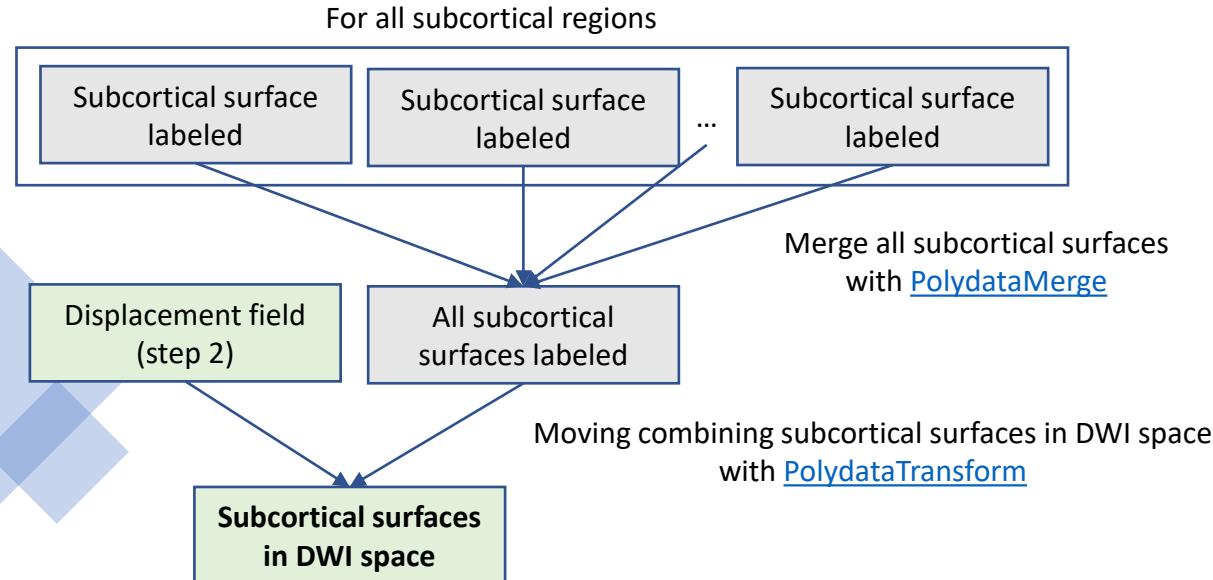
## Step 3 : Preprocessing for subcortical regions integration

### 1- Labelling of subcortical surfaces



Amygdala  
Caudate  
Pallidum  
Thalamus  
Hippocampus  
Putamen

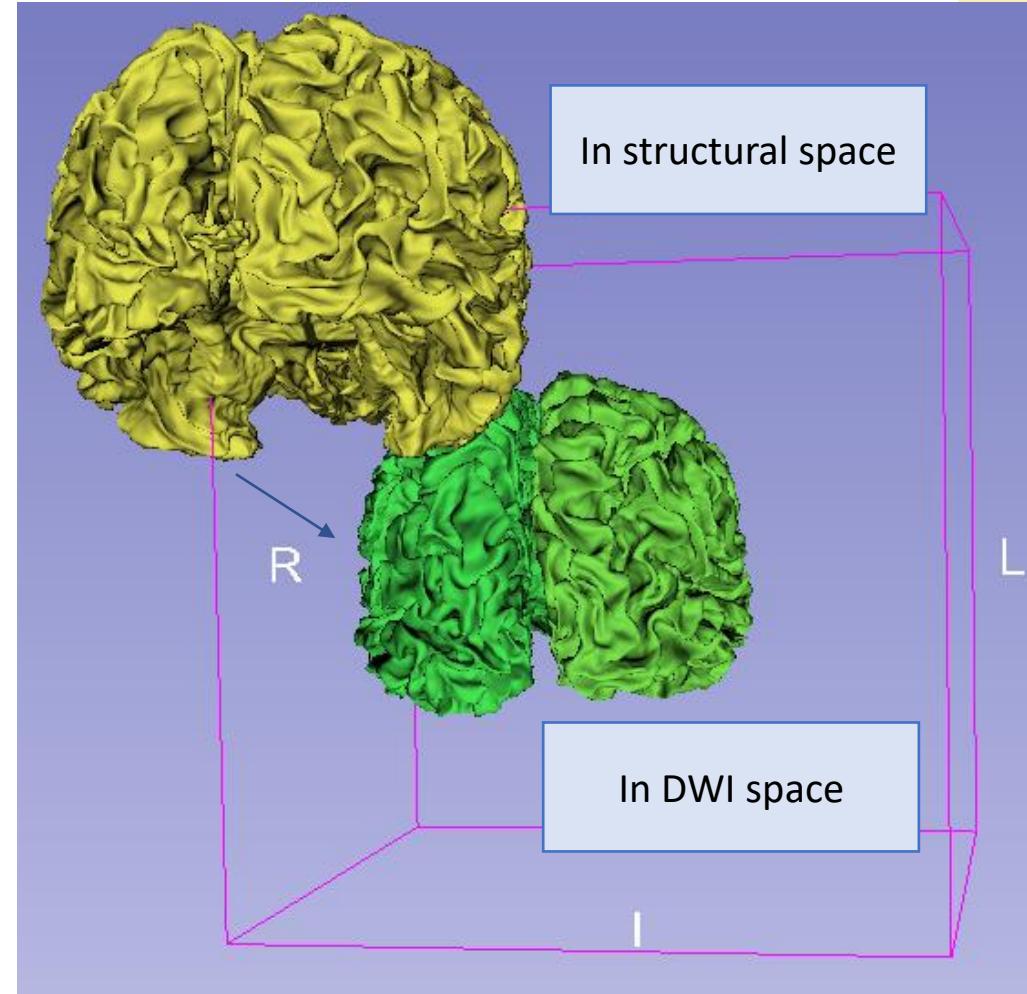
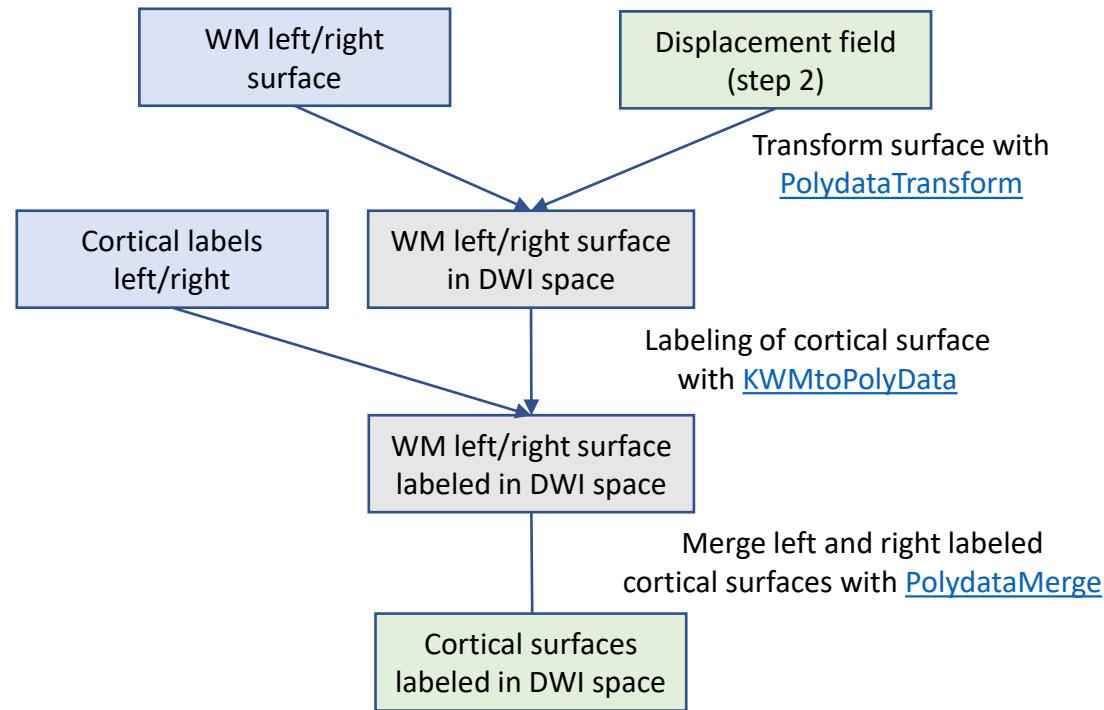
### 2- Registration of subcortical surfaces in DWI space



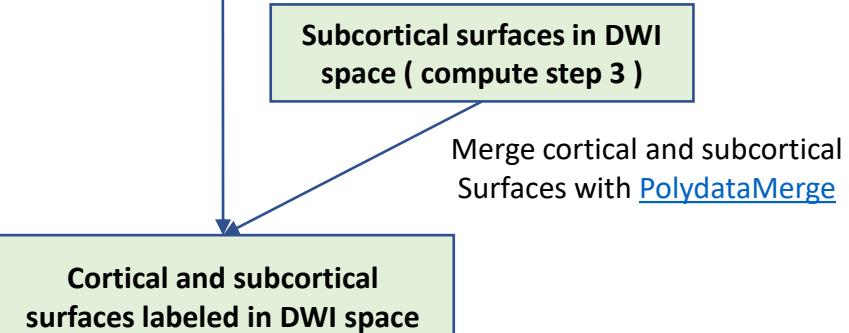
# 1. Registration pipeline: cortical and subcortical regions integration

CONTINUITY

## Step 4: Preprocessing for cortical regions integration



## Step 5: Cortical and subcortical regions integration



# Registration pipeline: cortical and subcortical integration

CONTINUITY

## Step 3 : Preprocessing for subcortical regions

### 1- Get subcortical data

→ Option 1: SPHARM surfaces and KWM folder provided by the user

→ Option 2: creation of SPHARM surfaces and KWM files:

For each subcortical region

#### 1-1- Computation of SALT files

Labeled image      Label of the considered region

[SegPostProcessCLP](#)

Image filtered

[GenParaMeshCLP](#)

Parametrization

Surface Mesh

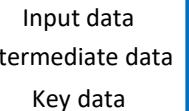
[ParaToSPHARMMeshCLP](#)

SPHARM surfaces

#### 1-2- Write KWM files

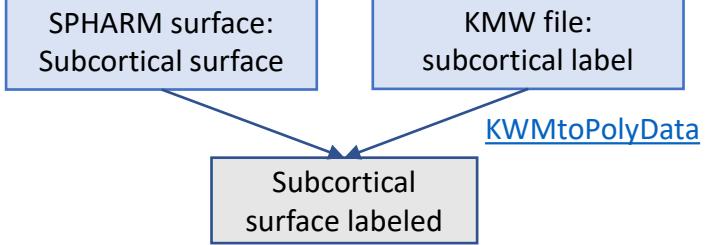
Write a text file with lines containing the label of this region (number of line = number of points)

## Summary step 3, 4 and 5



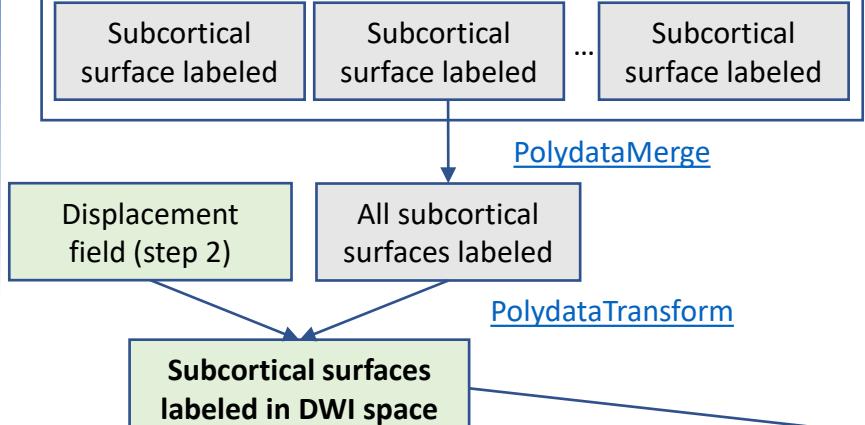
### 2- Labelling of subcortical surfaces

For each subcortical region



### 3- Registration of subcortical surfaces

All subcortical regions



## Step 4: Preprocessing for cortical regions

Displacement field  
(step 2)

WM left/right  
surface

[PolydataTransform](#)

WM left/right surface  
in DWI space

Cortical labels  
left/right

[KWMTopolData](#)

WM left/right surface  
labeled in DWI space

[PolydataMerge](#)

Cortical surfaces  
labeled in DWI space

## Step 5: Cortical and subcortical regions integration

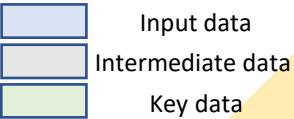
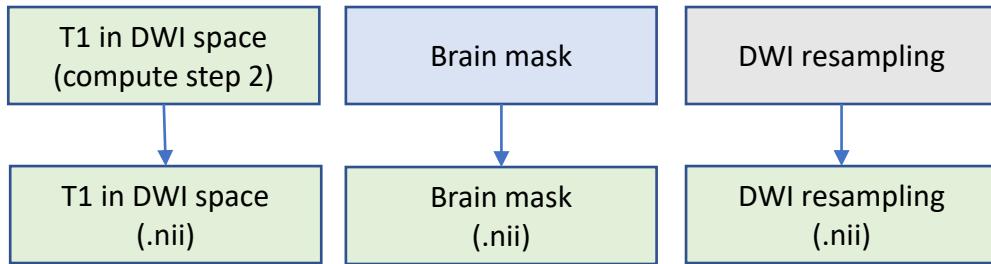
[PolydataMerge](#)

Cortical and subcortical surfaces  
labeled in DWI space

## 2. Tractography pipeline: create the seed list

CONTINUITY

### Step 6: FSL to NIFTI format with [DWIConvert](#)



```
1 #!ascii - generated by CreateLabelFiles project
2 1345 2529
3 -75.6246 101.444 -16.4707 0
4 -43.9875 105.254 -24.089 0
5 -55.178 106.206 -18.1101 0
6 -62.9873 105.324 -18.0872 0
7 -63.7495 98.7527 -12.8731 0
8 -54.4642 104.49 -21.3192 0
9 -50.18 106.297 -23.0675 0
10 -54.5996 101.066 -12.1431 0
11 -57.6687 101.907 -18.0114 0
12 -70.02 106.391 -14.2041 0
```

Extract of one of the .asc file

### Step 7: Extract label surfaces

Cortical and subcortical surfaces labeled in DWI space (compute step 5)

Creation of label surfaces for each regions from a vtk surface containing labels information with [ExtractLabelSurfaces](#)

Label name information for each point (.txt)

Label number information for each point (.txt)

Surfaces for each label

```
1 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11101.asc
2 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11102.asc
3 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11103.asc
4 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11104.asc
5 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11105.asc
6 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11106.asc
7 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11107.asc
8 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11108.asc
9 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11109.asc
10 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11110.asc
11 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11111.asc
12 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11112.asc
13 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11113.asc
14 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11114.asc
15 ./output_new_civility/T0054-1-1-6yr/Tractography/OutputSurfaces_overlapping/labelSurfaces/11115.asc
```

Extract of the seed list

### Step 8: Write a seed list

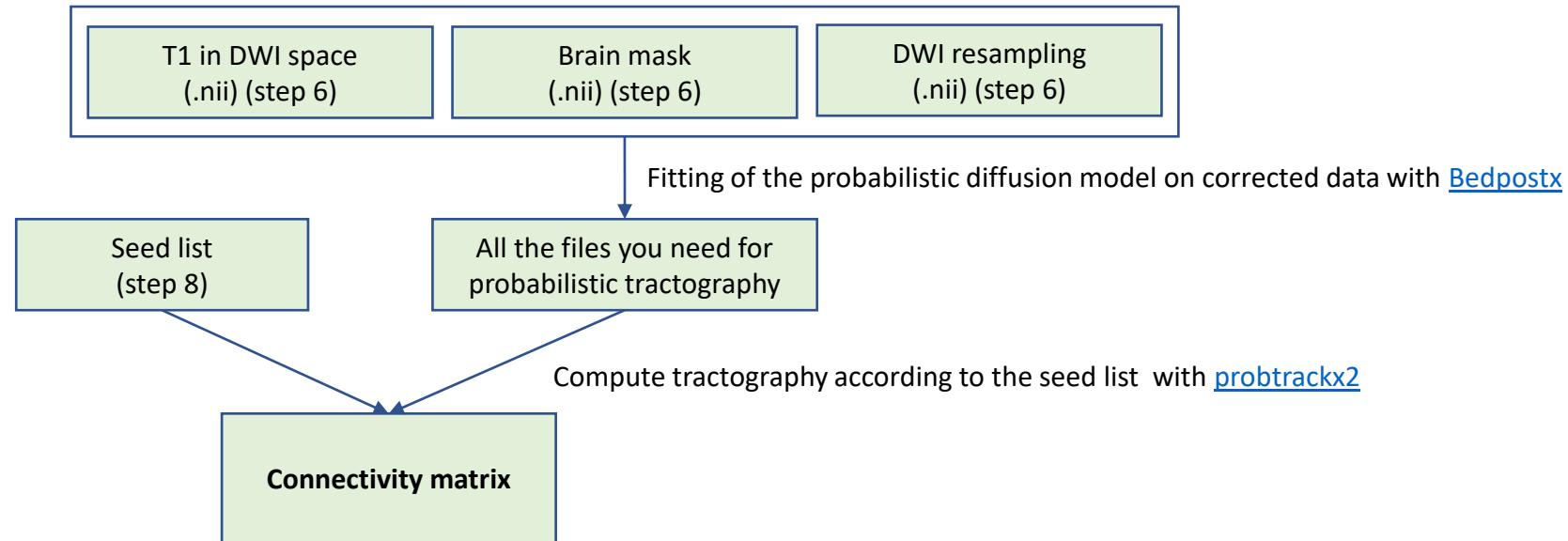
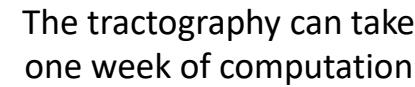
Write a text file listing all path of label surfaces created by ExtractLabelSurfaces

Seed list

## 2. Tractography pipeline with FSL

## CONTINUITY

## Step 9 - option 1: Tractography with FSL

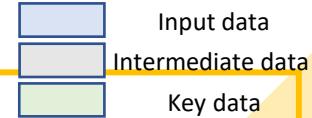


### Extract of the connectivity matrix

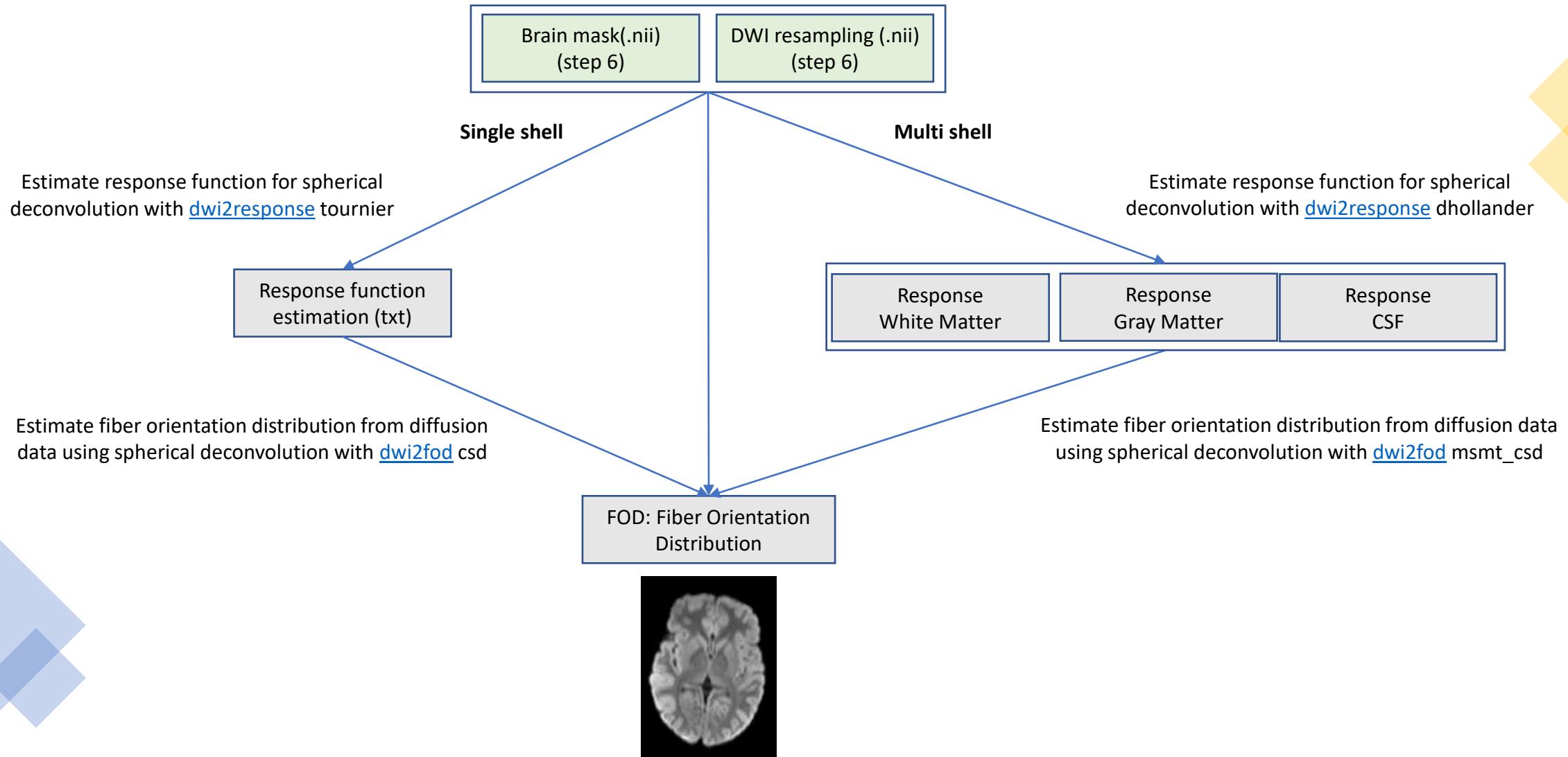
## 2. Tractography pipeline with MRtrix

CONTINUITY

### Step 9 - option 2: Tractography with MRtrix : Preprocessing



#### Step 9 - Preprocessing 1: computation of FOD (for iFDO1, iFOD2 or tcksif option)



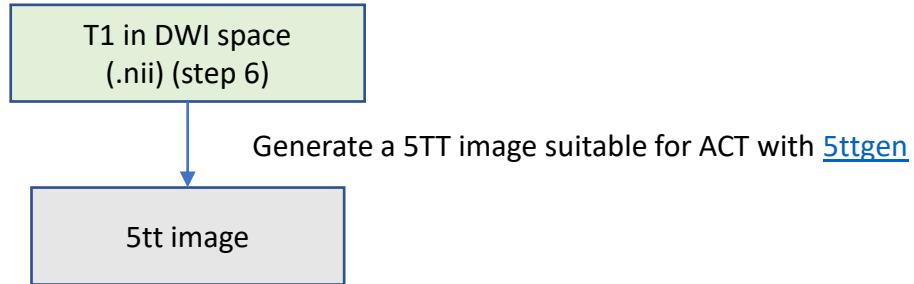
## 2. Tractography pipeline with MRtrix

CONTINUITY

### Step 9 - option 2: Tractography with MRtrix : Preprocessing

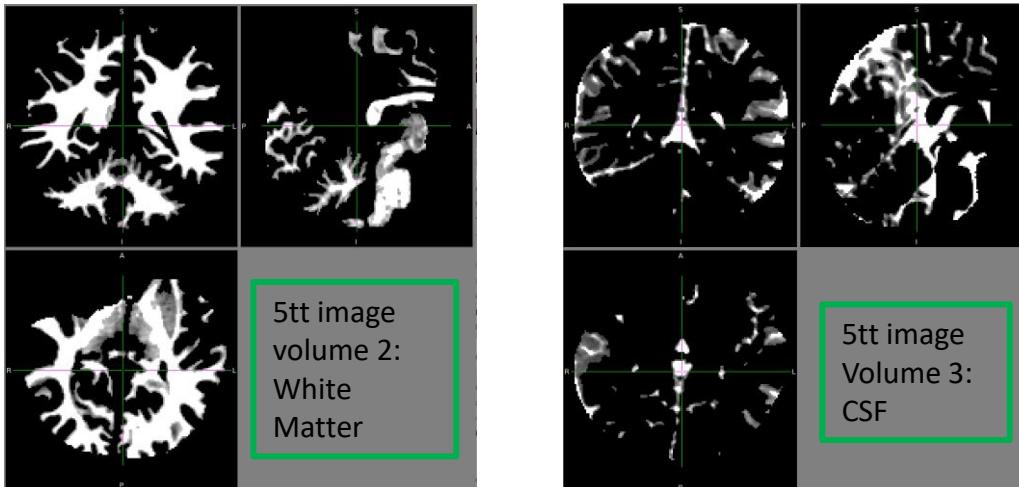
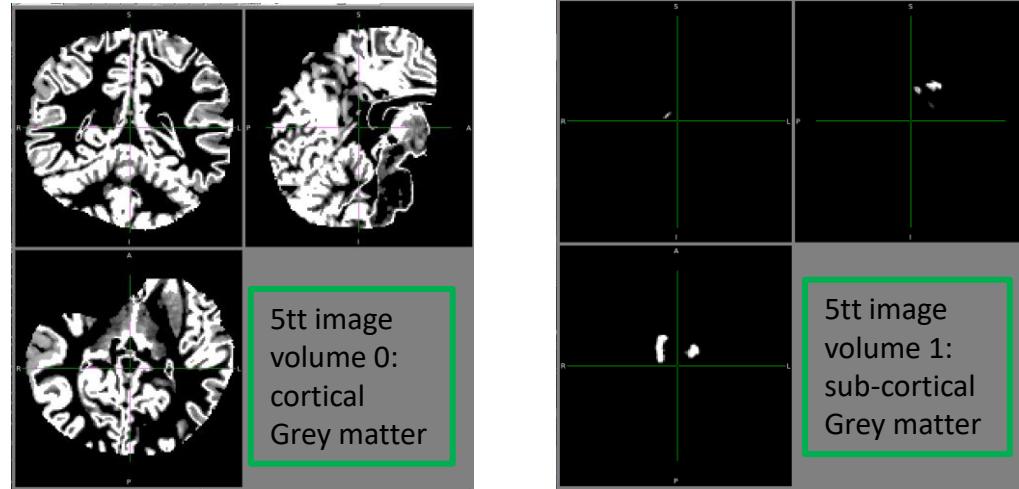
#### Step 9 - Preprocessing 2:

If ACT option (Anatomically-Constrained Tractography)



CONTINUITY and MRtrix can perform streamlines tractography with 3 algorithms:

- [iFOD1](#): First-order Integration over Fiber Orientation Distributions
- [iFOD2](#): Second-order Integration over Fiber Orientation Distributions
- [tensor\\_prob](#): Probabilistic Fiber Tracking Using the Wild Bootstrap With Diffusion Tensor MRI



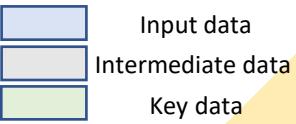
5tt image Volume 4: Pathological Tissue (empty for this subject)

## 2. Tractography pipeline with MRtrix

CONTINUITY

### Step 9 - option 2: Tractography with MRtrix :

For each region



#### Step 9 – 1: Tractography computation:

Seed list  
(step 8)      Brain mask  
.nii) (step 6)

If iFOD1 or iFOD2  
algorithm:  
FOD (step 9 –  
preprocessing 1)

If tensor\_prob  
algorithm:  
DWI resampling  
.nii) (step 6)

Perform streamlines tractography  
with [tckgen](#)

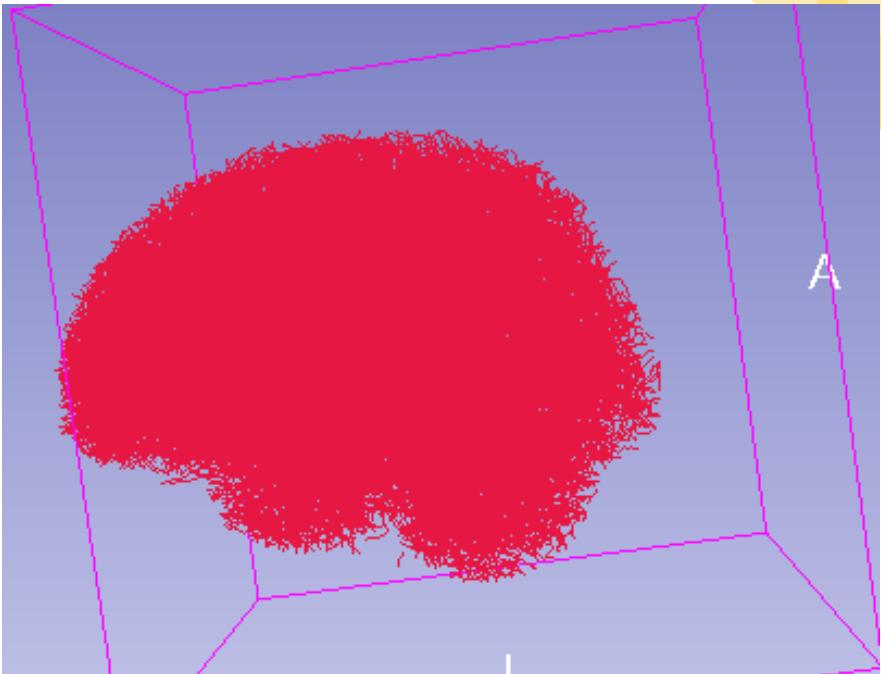
If act option :  
5tt image (step 9 –  
preprocessing 2)

Tracks ( all streamlines connected  
to the considered region)

#### Step 9 – 1': Filtering option:

Filtering with [tcksift](#)

Filtered tracks



Tracks: output tckgen

## 2. Tractography pipeline with MRtrix

CONTINUITY

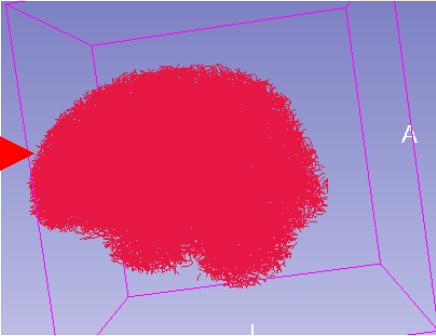
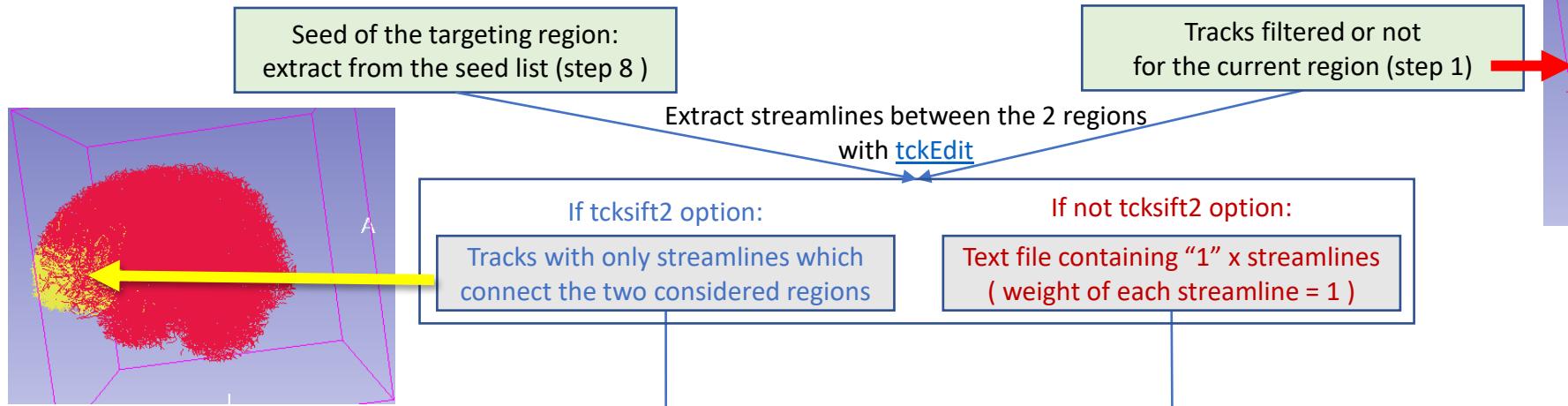
### Step 9 - option 2: Tractography with MRtrix :



For each region (current region: line of the connectome )

For each region (target region: row of the connectome)

#### Step 9 – 2: Extraction of streamlines between the two considered regions



#### Step 9 – 3: Connectivity value of the two considered regions computation

Compute weighting factor with [tcksift2](#)

FOD  
(compute step 9 preprocessing 1)

Text file containing the weighting  
factor for each streamline

Sum of all weights

Sum of all weights (= number of streamlines)

Connectivity values for the two considered regions

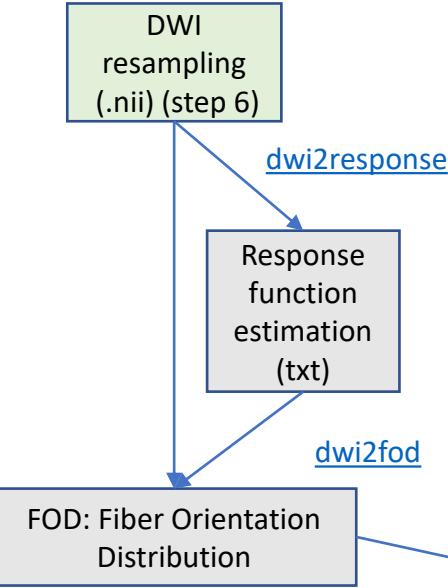
## 2. Tractography pipeline with MRtrix

### Step 9 - option 2: Tractography with MRtrix :

### Summary

### MRtrix

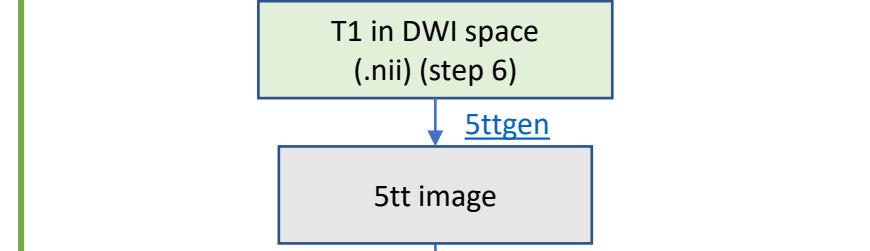
1- FOD (for iFOD1 and iFOD2 algorithm or tcksif option)



CONTINUITY and MRtrix can perform streamlines tractography with 3 algorithms:

- **iFOD1**: First-order Integration over Fiber Orientation Distributions
- **iFOD2**: Second-order Integration over Fiber Orientation Distributions
- **tensor\_prob**: Probabilistic Fiber Tracking Using the Wild Bootstrap With Diffusion Tensor MRI

2- If ACT option  
(Anatomically-Constrained Tractography)



For each subcortical region:

### 3- Tractography:

Seed list (step 8)

Brain mask (.nii) (step 6)

DWI resampling (.nii) (step 6)

If act option

tckgen

If tensor\_prob algorithm:

Tracks ( all streamlines connected to the considered region)

If iFOD1 or 2 algorithm

### 3': Filtering option:

tcksift

FOD (step MRtrix 1)

Filtered tracks

For each subcortical region (current region: line of the connectome)

For each subcortical region (target region: row of the connectome)

### 4- Extraction of streamlines between the 2 regions

Seed of the targeting region:  
extract from the seed list (step 8 )

Tracks filtered or not for the current region

Extract streamlines with tckEdit

If tcksift2 option:  
Streamlines which connect the two considered regions

If not tcksift2 option:  
Text file containing "1" x streamlines (weight of each streamline = 1)

### 5- Connectivity value of the 2 regions

Compute weighting factor with tcksift2

FOD (step MRtrix 1)

Text file containing the weighting factor for each streamline

Sum of all weights  
(= number of streamlines)

Sum of all weights

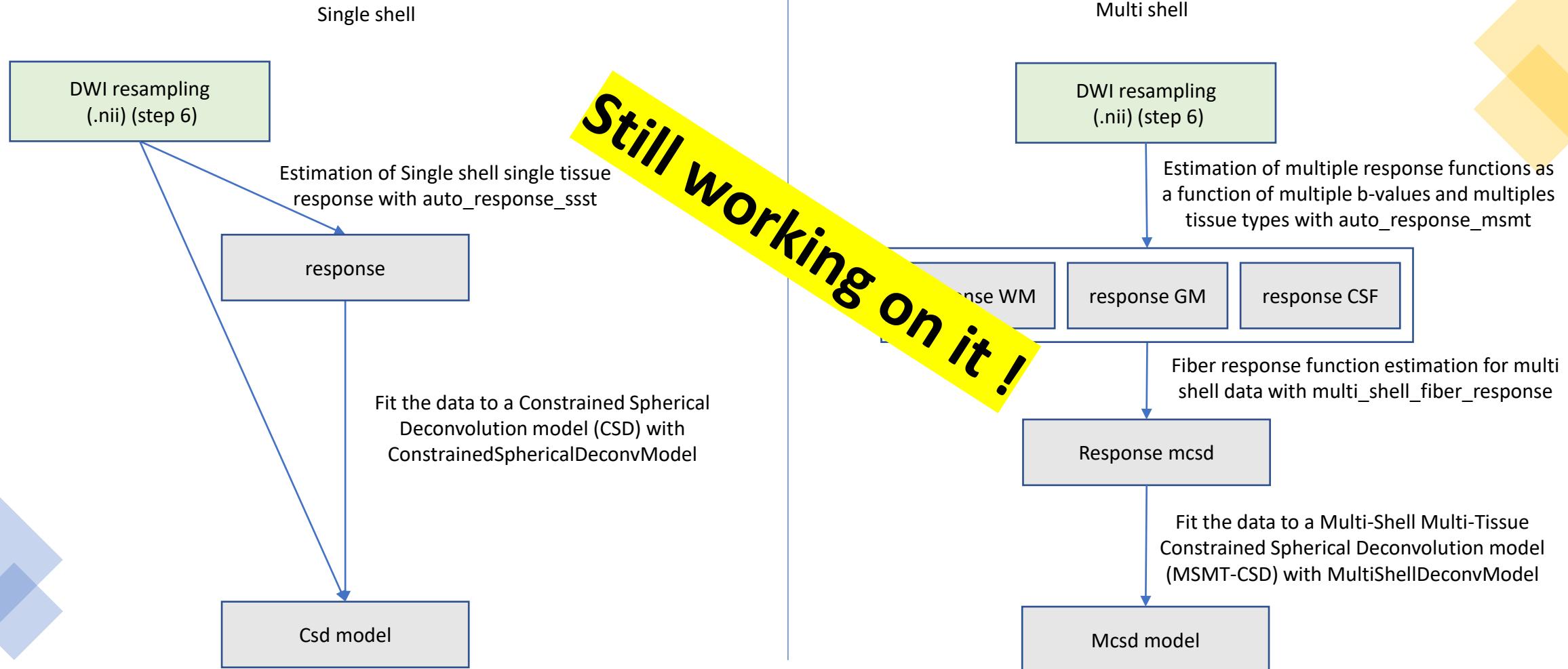
Connectivity values for the two considered regions

## 2. Tractography pipeline with DIPY

CONTINUITY

### Step 9 - option 3: Tractography with DIPY

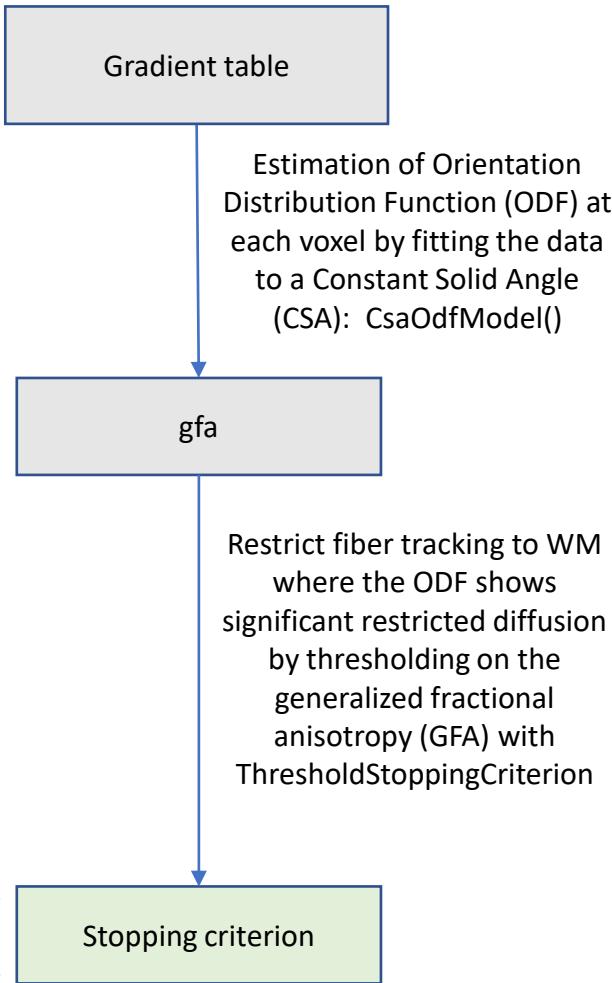
#### 1- Getting directions from a diffusion data set



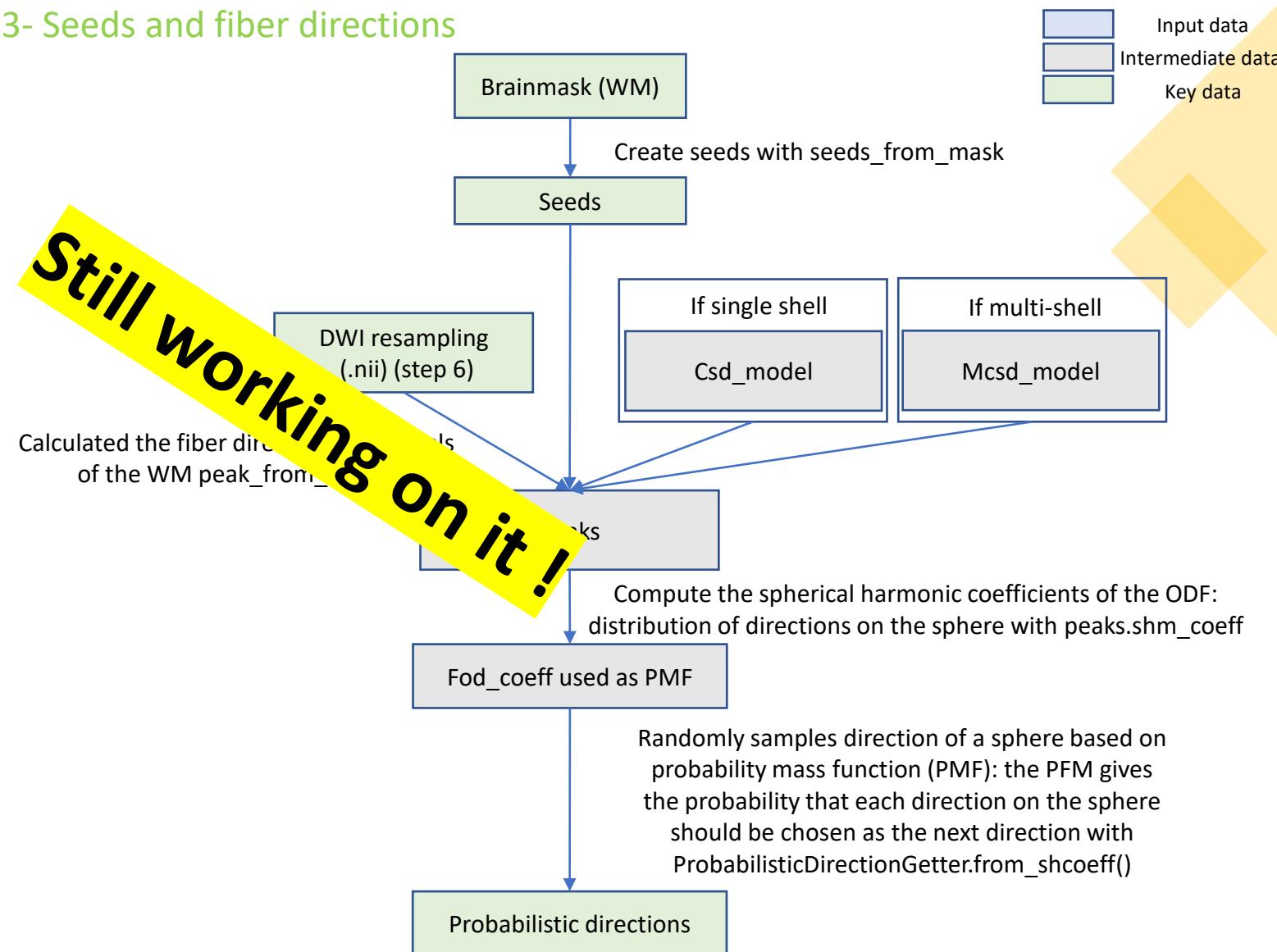
## 2. Tractography pipeline with DIPY

### Step 9 - option 3: Tractography with DIPY

#### 2- Stopping criterion



#### 3- Seeds and fiber directions

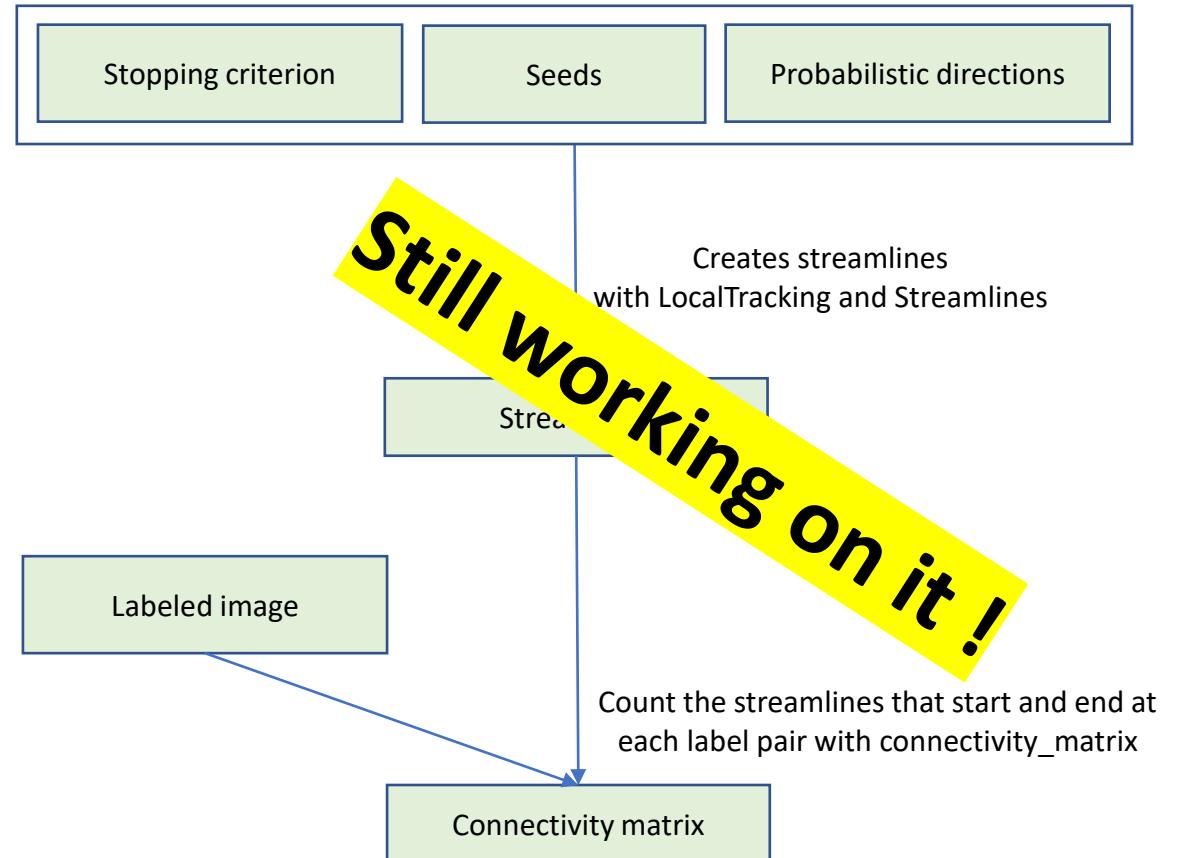


## 2. Tractography pipeline with DIPY

CONTINUITY

### Step 9 - option 3: Tractography with DIPY

#### 4- Generate streamlines and the connectivity matrix



Input data  
Intermediate data  
Key data

## 2. Tractography pipeline with DIPY

### Step 9 - option 3: Tractography with DIPY

#### 1- Getting directions from a diffusion data set

Single shell

DWI resampling (.nii) (step 6)

Estimation of response with auto\_response\_ssst

response

Fit the data to a Constrained Spherical Deconvolution model (CSD) with ConstrainedSphericalDeconvModel

Csd model

Multi shell

DWI resampling (.nii) (step 6)

Estimation of multiple response with auto\_response\_msmt

WM, GM and CSF responses

Fiber response function multi\_shell\_fiber\_respons

Response mcsd

Fit the data with MultiShellDeconvModel

Mcsd model

### Summary DIPY

#### 3- Seeds and fiber directions

Brainmask (WM)

seeds\_from\_mask

DWI resampling (.nii) (step 6)

Seeds

csd\_model

peaks

Calculated the fiber direction with peak\_from\_model

Fod\_coeff

Probabilistic directions

#### 4- Streamlines and connectivity matrix

Seeds

LocalTracking and Streamlines

Streamlines

Labeled image

Count the streamlines at each label pair

Connectivity matrix

#### 2- Stopping criterion

Gradient table

gfa

Stopping criterion

Estimation of Orientation Distribution Function (ODF) with CsaOdfModel and computation of GFA: generalized fractional anisotropy

Restrict fiber tracking to WM with ThresholdStoppingCriterion

Intermediate data

Key data

### 3. Run CONTINUITY: install libraries and activate the environment

CONTINUITY

#### To install all libraries required for CONTINUITY:

- In your terminal, run:

```
cd /tools/CONTINUITY/CONTINUITY_v1.1 ( On Longleaf: cd /proj/NIRAL/tools/CONTINUITY/CONTINUITY_v1.1 )
```

- Install miniconda by running :

```
bash Miniconda3-latest-Linux-x86_64.sh
```

- Now, you have to restart your terminal at the end

- On the new terminal, to install libraries, write :

```
cd /tools/CONTINUITY/CONTINUITY_v1.1 ( On Longleaf: cd /proj/NIRAL/tools/CONTINUITY/CONTINUITY_v1.1 )
conda env create -f CONTINUITY_env.yml
```

- Then (3min later):

```
conda activate CONTINUITY_env
```

#### For the next time, to run CONTINUITY, you just need to activate the environment:

```
conda activate CONTINUITY_env
```

### 3. Run CONTINUITY for one subject: two options

#### First option: with interface

On Longleaf:

```
python3 /proj/NIRAL/tools/CONTINUITY/CONTINUITY_v1.1/ main_CONTINUITY.py
```

#### Second option: with command line

On Longleaf:

```
python3 /proj/NIRAL/tools/CONTINUITY/CONTINUITY_v1.1/main_CONTINUITY.py -noGUI
```

→ Arguments required for new CIVILITY script  
(at least):

- ‘-DWI\_DATA’
- ‘-T1\_DATA’,
- ‘-BRAINMASK’
- ‘-PARCELLATION\_TABLE’,
- ‘-PARCELLATION\_TABLE\_NAME’
- ‘-SURFACE\_USER’

→ Other parameters are optional  
→ You have default values in an intern json file  
→ You can specify all parameters in the command line:

```
elodie@ubuntu:~/Desktop/CONTINUITY$ python3 main_CONTINUITY.py --help
usage: main_CONTINUITY.py [-h] [-default_config_filename [DEFAULT_CONFIG_FILENAME]] [-csv_file [CSV_FILE]] [-noGUI] [-cluster] [-ID] [-DWI_DATA] [-DWI_DATA_bvecs]
                           [-DWI_DATA_bvals] [-T1_DATA] [-T2_DATA] [-BRAINMASK] [-PARCELLATION_TABLE] [-PARCELLATION_TABLE_NAME] [-labelSetName] [-WM_L_Surf] [-WM_R_Surf]
                           [-SURFACE_USER] [-WM_L_Surf_NON_REGISTRATION] [-WM_R_Surf_NON_REGISTRATION] [-SALTDir] [-labeled_image] [-KWDdir] [-wm_mask] [-gm_mask]

Main CONTINUITY

optional arguments:
  -h, --help            show this help message and exit
  -default_config_filename [DEFAULT_CONFIG_FILENAME]
                        json with default configuration
  -csv_file [CSV_FILE]  csv file with data information for one or several subject
  -noGUI               Use CONTINUITY script without interface and json user file
  -cluster              Run script in a cluster
  -ID                  Job name
  -DWI_DATA             DWI Image (.nrrd)
  -DWI_DATA_bvecs       DWI Image bvectors (in case of DWI in FSL format (.nii.gz))
  -DWI_DATA_bvals       DWI Image bvalues (in case of DWI in FSL format (.nii.gz))
  -T1_DATA              T1 (.nrrd)
  -T2_DATA              T2 (.nrrd)
  -BRAINMASK            Brain mask in DWI space (.nrrd)
  -PARCELLATION_TABLE   Parcellation table (.json)
  -PARCELLATION_TABLE_NAME
                        Parcellation table name
  -labelSetName         Labelset name in vtk surface file
  -WM_L_Surf            Path to WM_L_Surf data
  -WM_R_Surf            Path to WM_R_Surf data
  -SURFACE_USER          Surface data (left and right white matter already combined)
  -WM_L_Surf_NON_REGISTRATION
                        Path to WM_L_Surf data in diffusion space
  -WM_R_Surf_NON_REGISTRATION
                        Path to WM_R_Surf data in diffusion space
  -SALTDir              Path to SALT directory
  -labeled_image         labeled_image
  -KWDdir               Set where the labels are stored with specific parcellation
  -wm_mask              white matter mask for DIPY
  -gm_mask              gray matter mask for DIPY
```

### 3. Run CONTINUITY for several subjects thanks to a CSV file

CONTINUITY

With just a CSV file:

On Longleaf:

```
/proj/NIRAL/Tools/CONTINUITY/CONTINUITY_v1.1/python3 main_CONTINUITY.py  
    -noGUI  
    -csv-file path/to/your/csv/file
```

Headers

Subject 1

```
1 ID.DWI_DATA.T1_DATA.BRAINMASK.PARCELLATION_TABLE.WM_L_Surf.WM_R_Surf.OUT_PATH  
2 T0054-1-1-6yr,./input_CONTINUITY/T0054-1-1-6yr_42_DWI-Trio_QCed_VC.nrrd,./input_CONTINUITY/T0054-1-1-6yr-T1_SkullStripped_scaled.nrrd,./input_CONTINUITY/  
T0054-1-1-6yr-brainmask.nrrd,./input_CONTINUITY/TABLE Destrieux.json,./input_CONTINUITY/  
stx_T0054-1-1-6yr-T1_SkullStripped_scaled_BiasCorr_corrected_multi_atlas_white_surface_rsl_left_327680_native_ITKspace.vtk,./input_CONTINUITY/  
stx_T0054-1-1-6yr-T1_SkullStripped_scaled_BiasCorr_corrected_multi_atlas_white_surface_rsl_right_327680_native_ITKspace.vtk,./output_CONTINUITY  
3 T0055-1-1-6yr,./input_CONTINUITY/T0055-1-1-6yr_42_DWI-Trio_QCed_VC.nrrd,./input_CONTINUITY/T0055-1-1-6yr-T1_SkullStripped_scaled.nrrd,./input_CONTINUITY/  
T0055-1-1-6yr-brainmask.nrrd,./input_CONTINUITY/TABLE Destrieux.json,./input_CONTINUITY/  
stx_T0055-1-1-6yr-T1_SkullStripped_scaled_BiasCorr_corrected_multi_atlas_white_surface_rsl_left_327680_native_ITKspace.vtk,./input_CONTINUITY/  
stx_T0055-1-1-6yr-T1_SkullStripped_scaled_BiasCorr_corrected_multi_atlas_white_surface_rsl_right_327680_native_ITKspace.vtk,./output_CONTINUITY
```

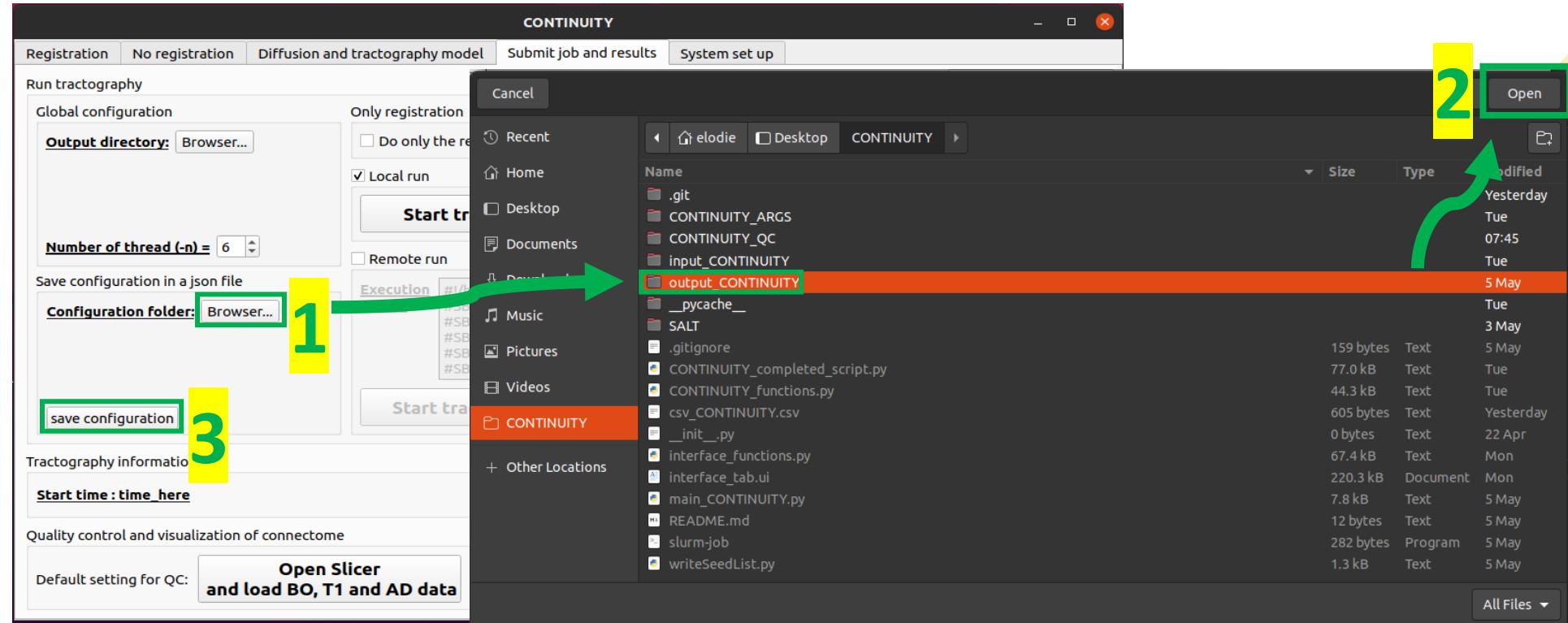
→ You can add other headers

Subject 2

### 3. Run CONTINUITY with your own saved parameters (for one or several subjects )

CONTINUITY

Step 1- Save your configuration in a json file: click on “Browser” → select a folder and click on “Open” → download your json file



Step 2- Call the json file on the command line to use it at the default configuration file:

For one subject:

On Longleaf:

```
python3 /proj/NIRAL/tools/CONTINUITY/CONTINUITY_v1.1/  
main_CONTINUITY.py  
-noGUI -default_config_filename path/to/your/saved/json
```

For several subject thanks to a CSV file:

On Longleaf:

```
python3 /proj/NIRAL/tools/CONTINUITY/CONTINUITY_v1.1/  
main_CONTINUITY.py  
-noGUI -default_config_filename path/to/your/saved/json  
-csv-file path/to/your/csv/file
```

### 3. Run CONTINUITY locally and remotely

Run CONTINUITY locally (by default)

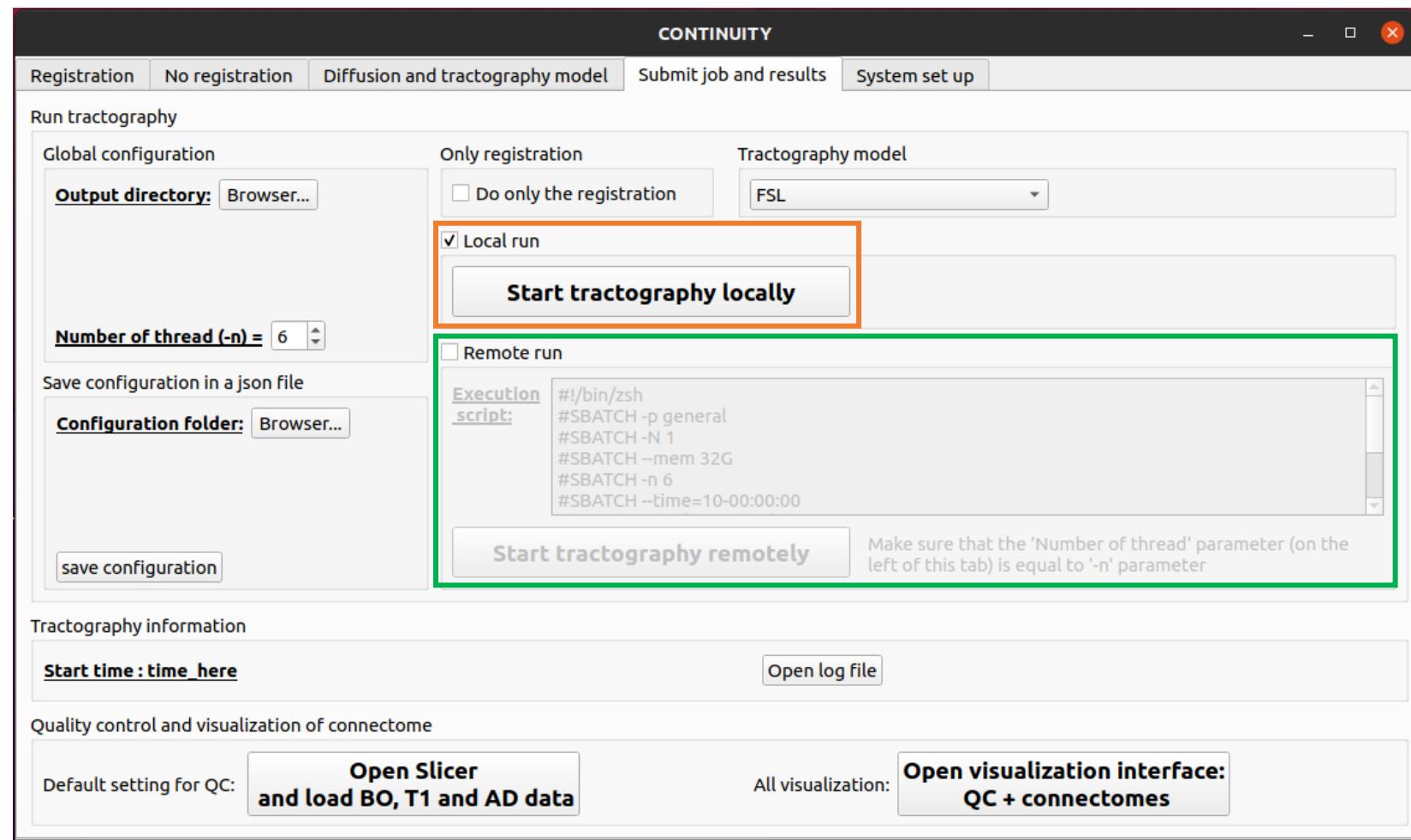
Run CONTINUITY remotely:

-First option: with the interface

-Second option: with the command line:

On Longleaf:

```
python3 /proj/NIRAL/tools/CONTINUITY/
CONTINUITY_v1.1/main_CONTINUITY.py
    - noGUI
    - cluster
```



## First tab: ‘Registration’ → ‘Data specification’ → ‘Job name and image data’

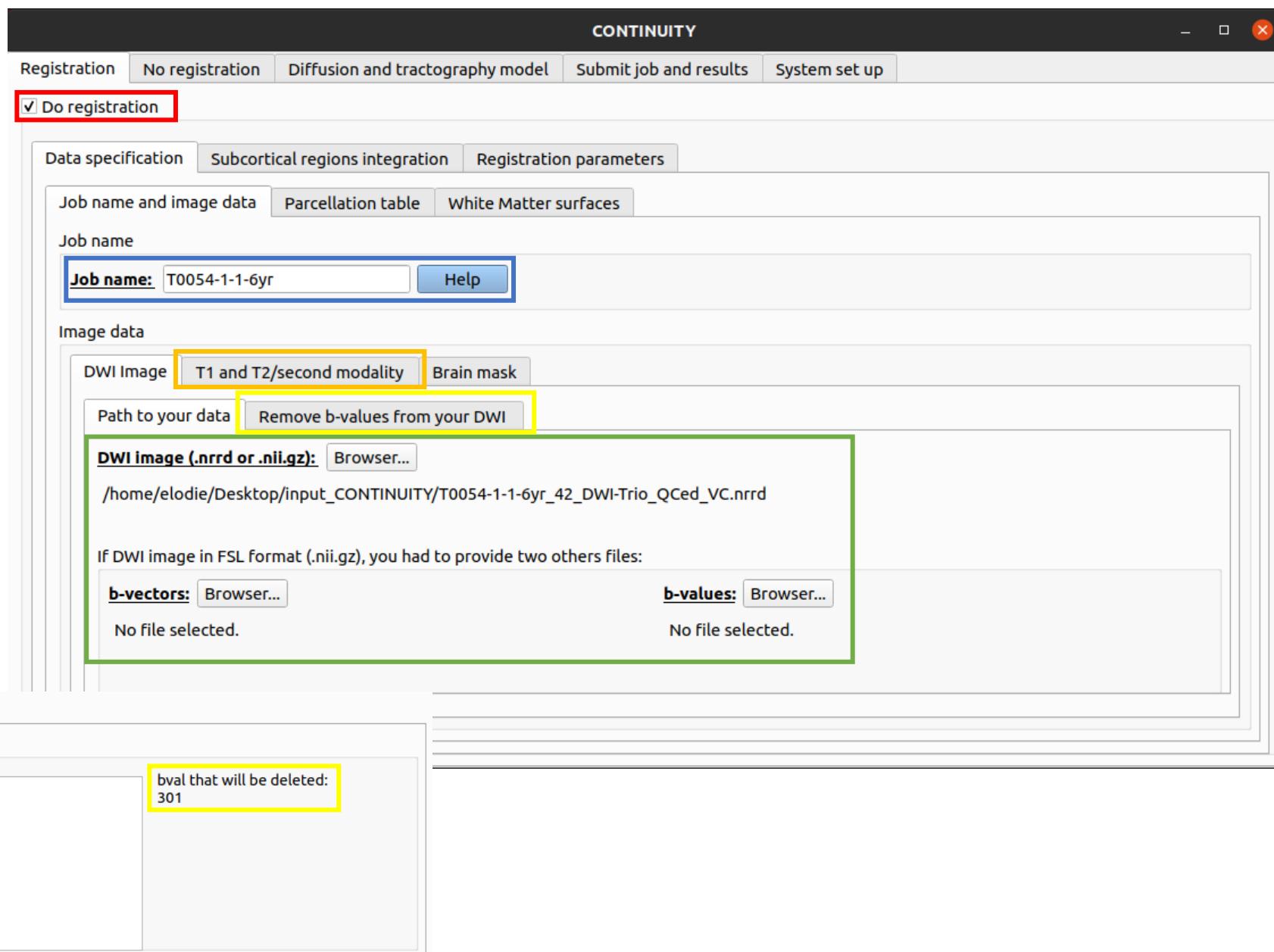
- Click on “Do registration” to be able to select input data
- If you don’t want to do it with CONTINUITY, you can skip this tab

- The job name is optional, and you can click on the “help” button for more information.
- Click again to close it.

- T2/second modality image is an option. If this modality is specified, you can use it in ANTs command

- DWI image can be in .nrrd or in .nii format

- In case of a FSL format (.nii.gz) and multi-shell DWI data, you can select b-values to remove them from your DWI data :



# First tab: ‘Registration’ → ‘Data specification’ → ‘Parcellation table’

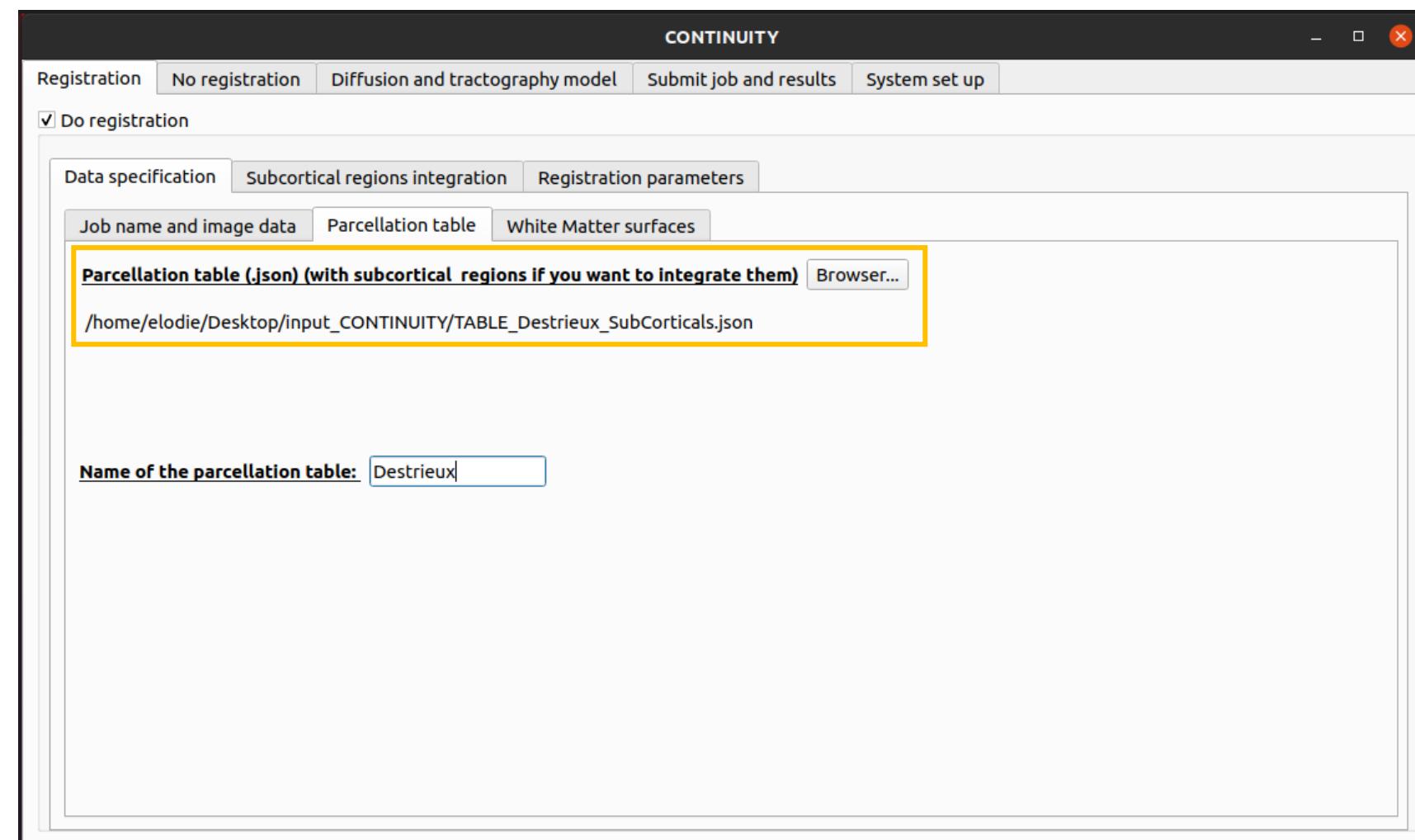
CONTINUITY

- To integrate the subcortical regions, you must provide a parcellation table with subcortical data information
- Example: extract from the parcellation table (.json file) for one region:

```
"VisuOrder": 149,  
"MatrixRow": 148,  
"name": "sub_lh_amy",  
"VisuHierarchy": "seed.left.",  
"coord": [  
    0,  
    0,  
    0  
,  
    "labelValue": "11176",  
    "AAL_ID": 11176,  
    "subcortical": true  
]
```

CONTINUITY needs at least (for each region):

- A name and a matching ‘labelValue’
- A “VisuHierarchy” specification to display group of regions in the circle connectome
- Three coordinates to visualize this region in the brain connectome (2D and 3D)  
(except for ‘Destrieux’ parcellation: CONTINUITY can compute the coordinate in case of registration with this tool)
- An AAL\_ID to write the seed list (same number as ‘labelValue’)
- A boolean ‘subcortical’ to indicate if the corresponding region is a cortical or a subcortical region



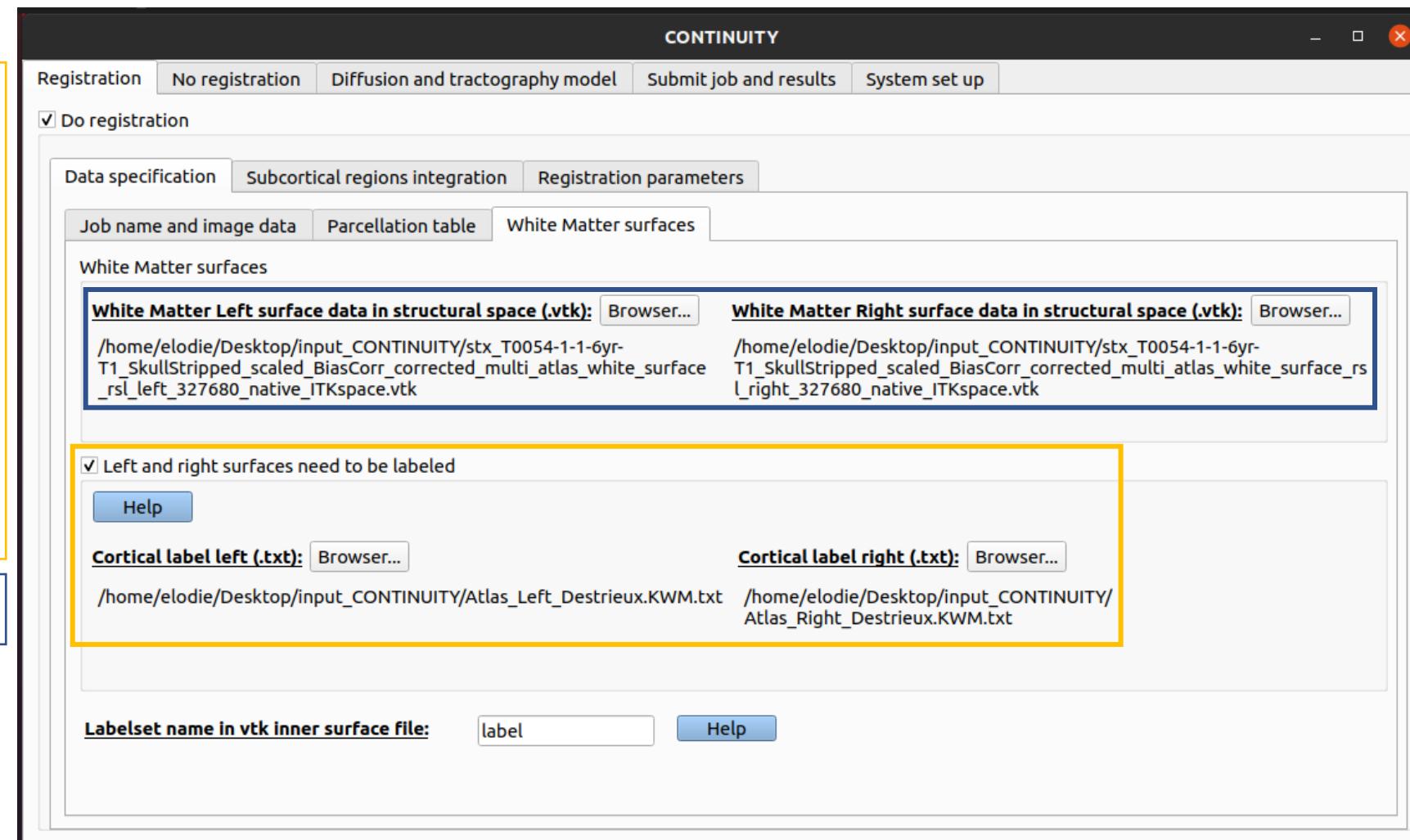
# First tab: ‘Registration’ → ‘Data specification’ → ‘Surface data’

CONTINUITY

- If your left and right surfaces need to be labeled, you should provide two additional files: cortical label left and cortical label right
- Extract of one of this file:

```
NUMBER_OF_POINTS=163842
DIMENSION=1
TYPE=Scalar
1.1106000e+04
1.1166000e+04
1.1112000e+04
1.1174000e+04
1.1115000e+04
1.1165000e+04
1.1157000e+04
1.1102000e+04
```

- You must provide left and right white matter surfaces in structural space (.vtk)



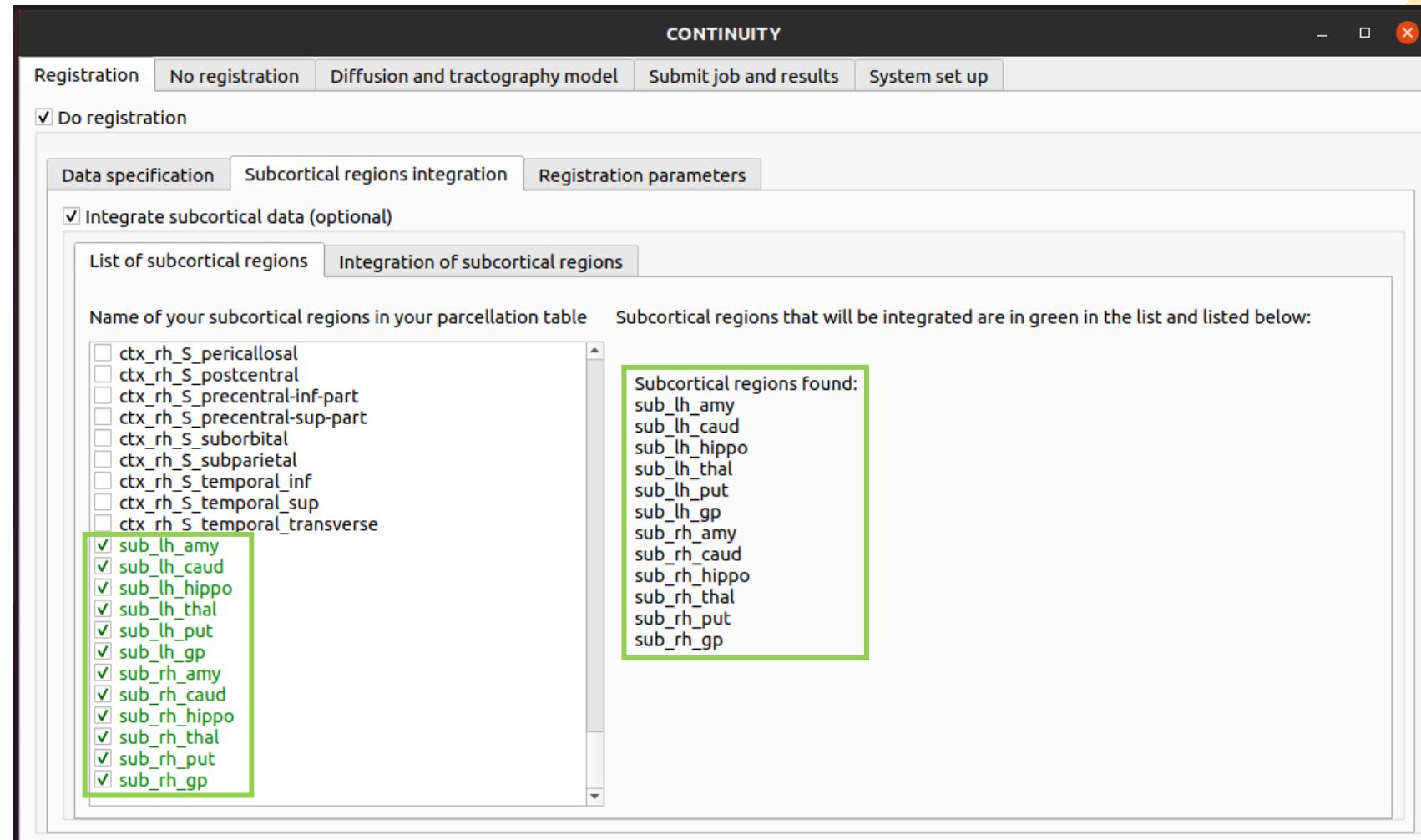
## First tab: ‘Registration’ → ‘Subcortical regions integration’

CONTINUITY

- To integrate subcortical regions, CONTINUITY will open your parcellation table and display in green all regions with ‘subcortical’ = True  
(if this parameter is missing in the parcellation table, you can select manually the subcortical regions thanks to the interface )

```
"VisuOrder": 149,  
"MatrixRow": 148,  
"name": "sub_lh_amy",  
"VisuHierarchy": "seed.left.",  
"coord": [  
    0,  
    0,  
    0  
,  
    "labelValue": "11176",  
    "AAL_ID": 11176,  
    "subcortical": true
```

- You can remove subcortical or add region by checked the corresponding name



## First tab: ‘Registration’ → ‘Subcortical regions integration’

CONTINUITY

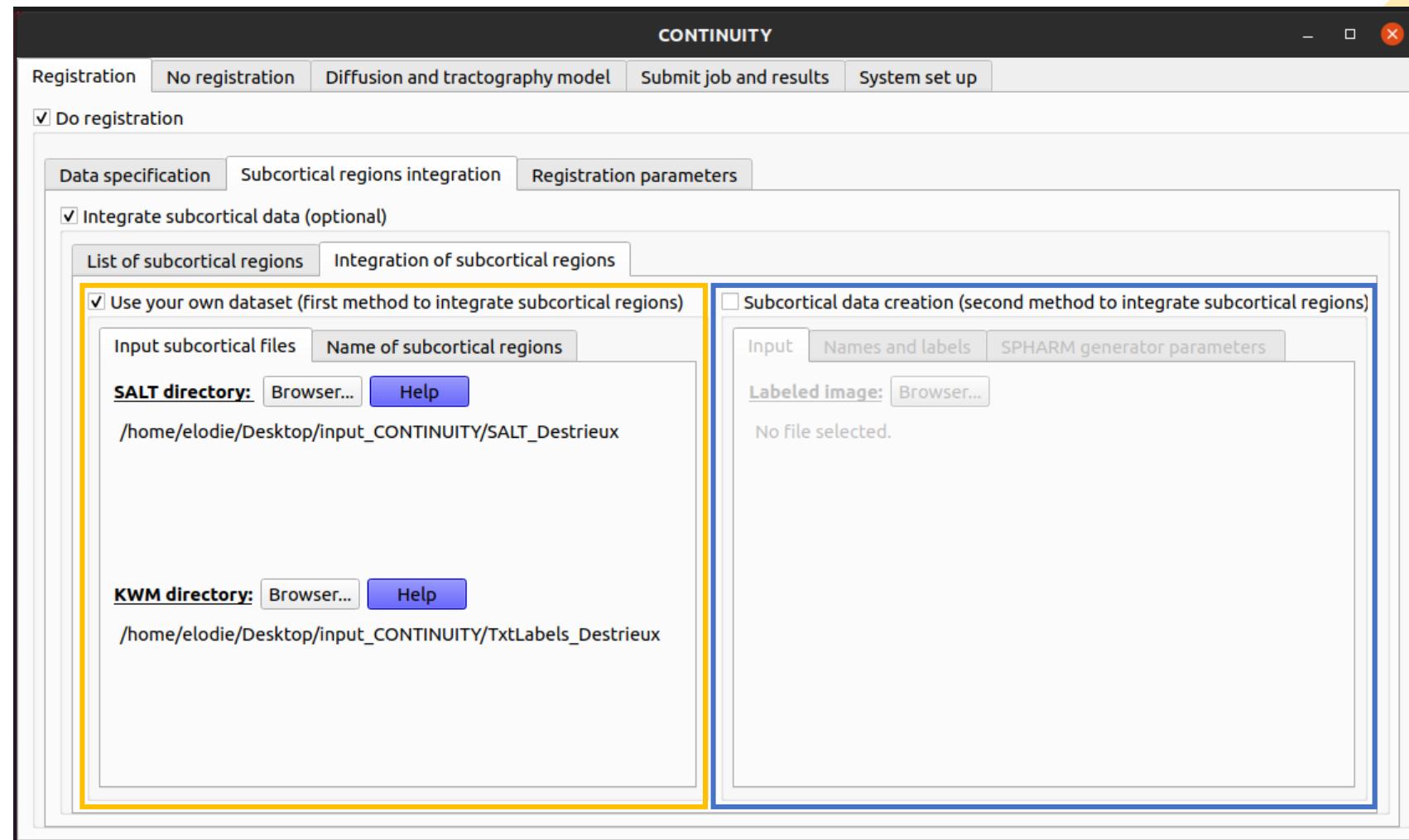
To integrate subcortical regions, CONTINUITY provides two different methods:

First method: ‘Use you own dataset’

Subcortical region integration with SALT and KWM folder provided by the user

Second method: ‘Subcortical data creation’

Subcortical region integration with labeled image and match between label and label-name for each subcortical regions



First method ‘Use you own dataset’:

Subcortical region integration with SALT and KWM folder provided by the user

→ In SALT directory, you must provide one .vtk file (SPHARM surface) per subcortical region.

→ Example of a name:

T0054-1-1-6yr-T1\_SkullStripped\_scaled\_label\_AmyL\_pp\_surfSPHARM.vtk  
job name region

→ KWM directory contain one text file per subcortical regions (see explanation of pipeline to have an example)

Use your own dataset

Input subcortical files

Name of subcortical regions

Checkbox in green: file for this region in the SALT and KWM directory  
 Checkbox in red: file for this region only in the KWM directory  
 Checkbox in purple: file for this region only in the SALT directory

sub\_lh\_amy  
sub\_lh\_caud  
sub\_lh\_gp  
sub\_lh\_hippo  
sub\_lh\_put  
sub\_lh\_thal  
sub\_rh\_amy  
sub\_rh\_caud  
sub\_rh\_gp  
sub\_rh\_hippo  
sub\_rh\_put  
sub\_rh\_thal

Second sub-tab “name of subcortical regions”:

- After setting the path of the SALT and KWM folder, CONTINUITY provide a list of subcortical regions
- You can only integrate ‘green’ regions

Use your own dataset (first method to integrate subcortical regions)

Input subcortical files

Name of subcortical regions

SALT directory:

Browser...

Help

/home/elodie/Desktop/input\_CONTINUITY/SALT\_Destrieux

KWM directory:

Browser...

Help

/home/elodie/Desktop/input\_CONTINUITY/TxtLabels\_Destrieux

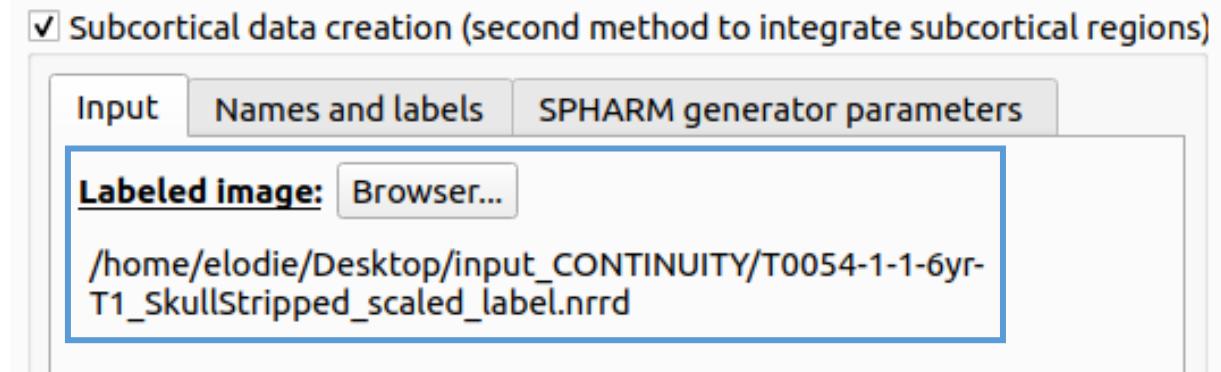
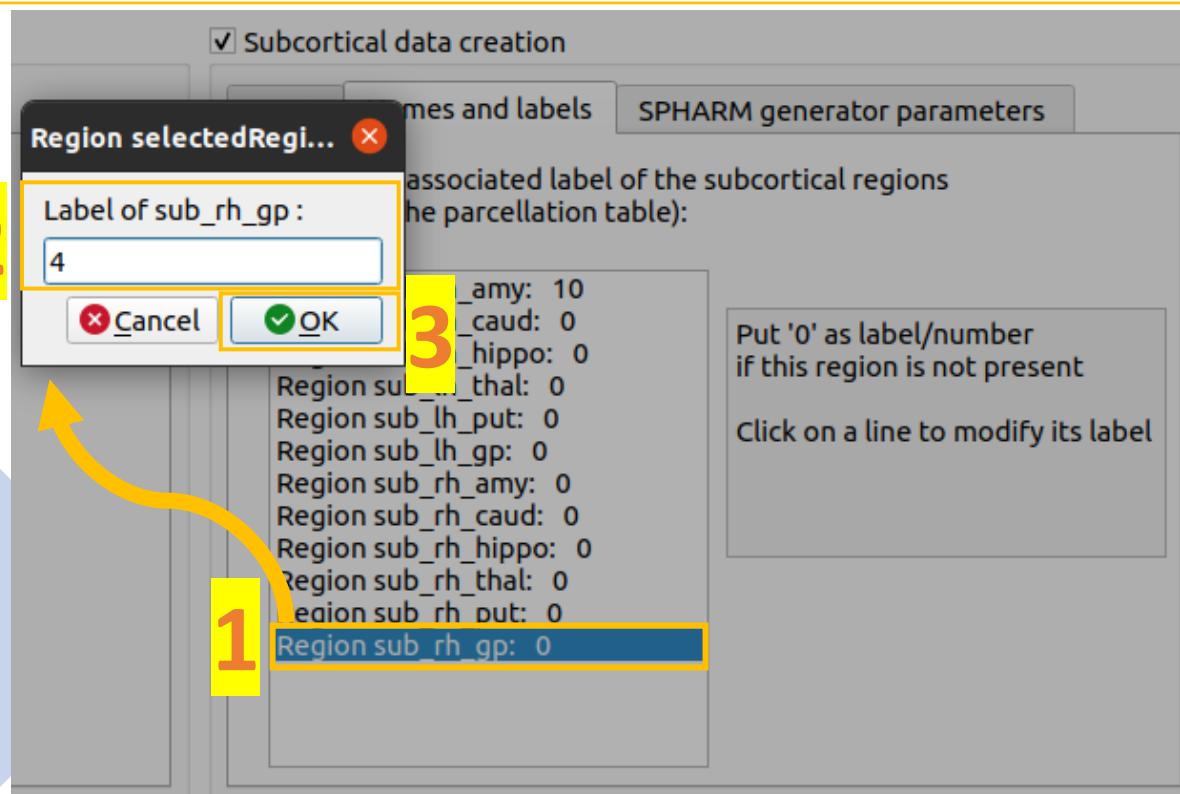
First tab: 'Registration' → 'Subcortical regions integration' → 'Subcortical data creation'

Second method: subcortical data creation

Subcortical region integration with labeled image and match between label and label-name for each subcortical regions

→ First, you need a labeled image with information for each subcortical regions:

→ Then write a label for each subcortical region by clicking on the name, writing the label and clicking on "ok" to save the label:



For example:

```
Region sub_lh_amy: 1
Region sub_lh_caud: 3
Region sub_lh_hippo: 5
Region sub_lh_thal: 40
Region sub_lh_put: 9
Region sub_lh_gp: 7
Region sub_rh_amy: 2
Region sub_rh_caud: 4
Region sub_rh_hippo: 6
Region sub_rh_thal: 41
Region sub_rh_put: 10
Region sub_rh_gp: 8
```

First tab: ‘Registration’ → ‘Subcortical regions integration’ → ‘Subcortical data creation’

Second method: subcortical data creation

Subcortical region integration with labeled image and match between label and label-name for each subcortical regions

→ You can modify some SPHARM generator parameters:

Subcortical data creation (second method to integrate subcortical regions)

Input   Names and labels   SPHARM generator parameters

**Post Processed Segmentation**

Command line parameters to run SegPostProcessCLP function  
**Default :**  
`SegPostProcessCLP labeled_image --label label_of_this_region --rescale --space 0.5,0.5,0.5`

Modify parameters (MODIFY ONLY IF YOU KNOW WHAT TO DO )

if checked: both --rescale and --space were not specified  
 ( and set to the internal default values)

Enforced spacing (x,y,z):  x  x

Subcortical data creation

Input   Names and labels   SPHARM generator parameters

**Post Processed Segmentation**

Command line parameters to run GenParaMeshCLP:  
**Default :**  
`GenParaMeshCLP --EulerFile --outEulerName Euler_txt_file`  
`Output_of_SegPostProcessCLP Output_Para_Mesh`  
`Output_Surface_Mesh --iter 500`

Modify parameters (MODIFY ONLY IF YOU KNOW WHAT TO DO )

Number of iterations=

Subcortical data creation

Input   Names and labels   SPHARM generator parameters

**Generate Mesh Parameters**

Command line parameters to run ParaToSPHARMMeshCLP:  
**Default :**  
`ParaToSPHARMMeshCLP Input_Para_Mesh Input_Surface_Mesh`  
`Surftarget_prefix --spharmDegree 10 --subdivLevel 15`

Modify parameters (MODIFY ONLY IF YOU KNOW WHAT TO DO )

Set the subdivision level for the icosahedron subdivision:  
 SubdivLevel value =

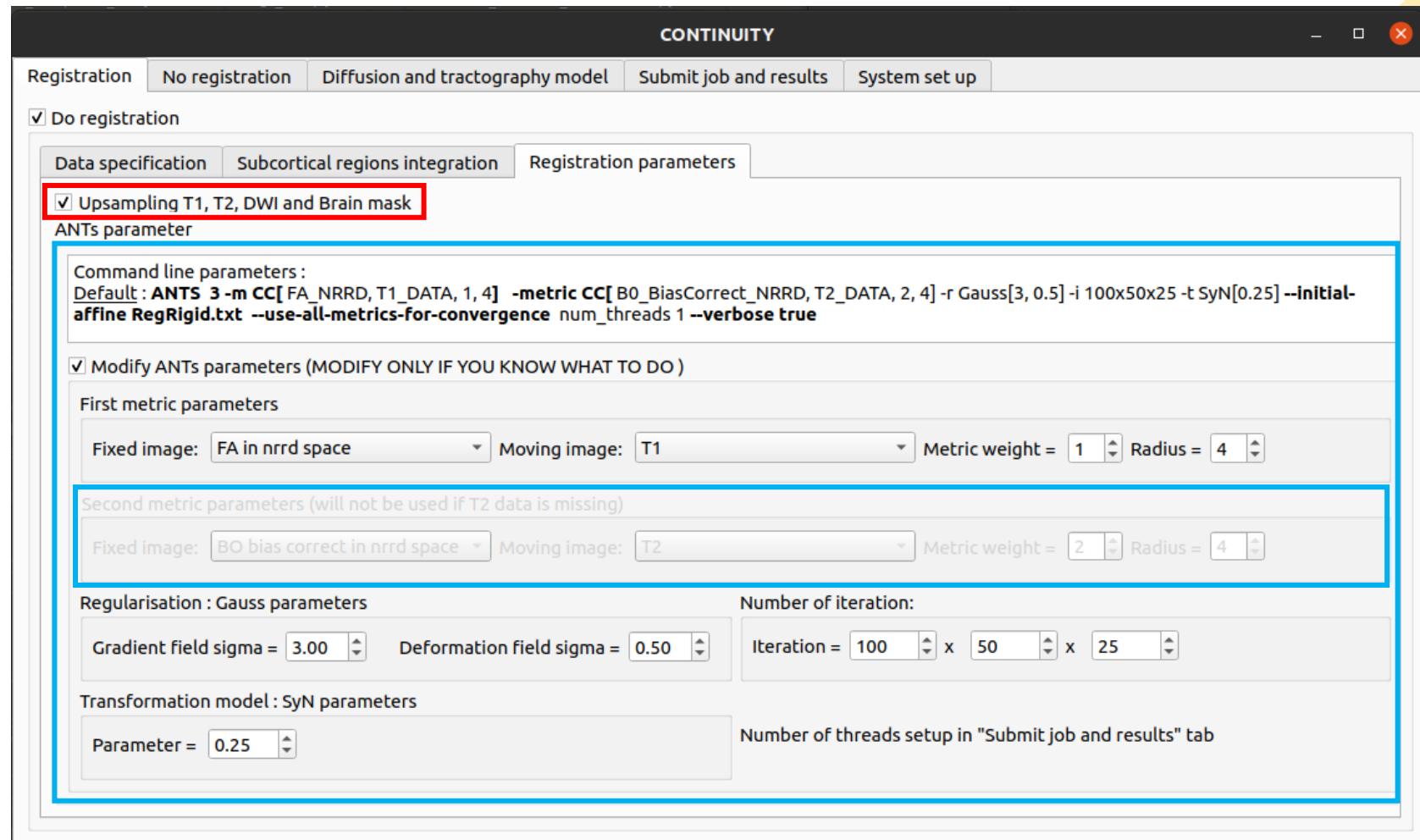
Set the maximal degree for the SPHARM computation:  
 SPHARM Degree value =

If the creation of one of this surfaces failed:

For each surface file not created, in your output folder/mySALT you will have a file name “manual\_fixe.txt” containing the name of file failing

## First tab: 'Registration' → 'Registration parameters'

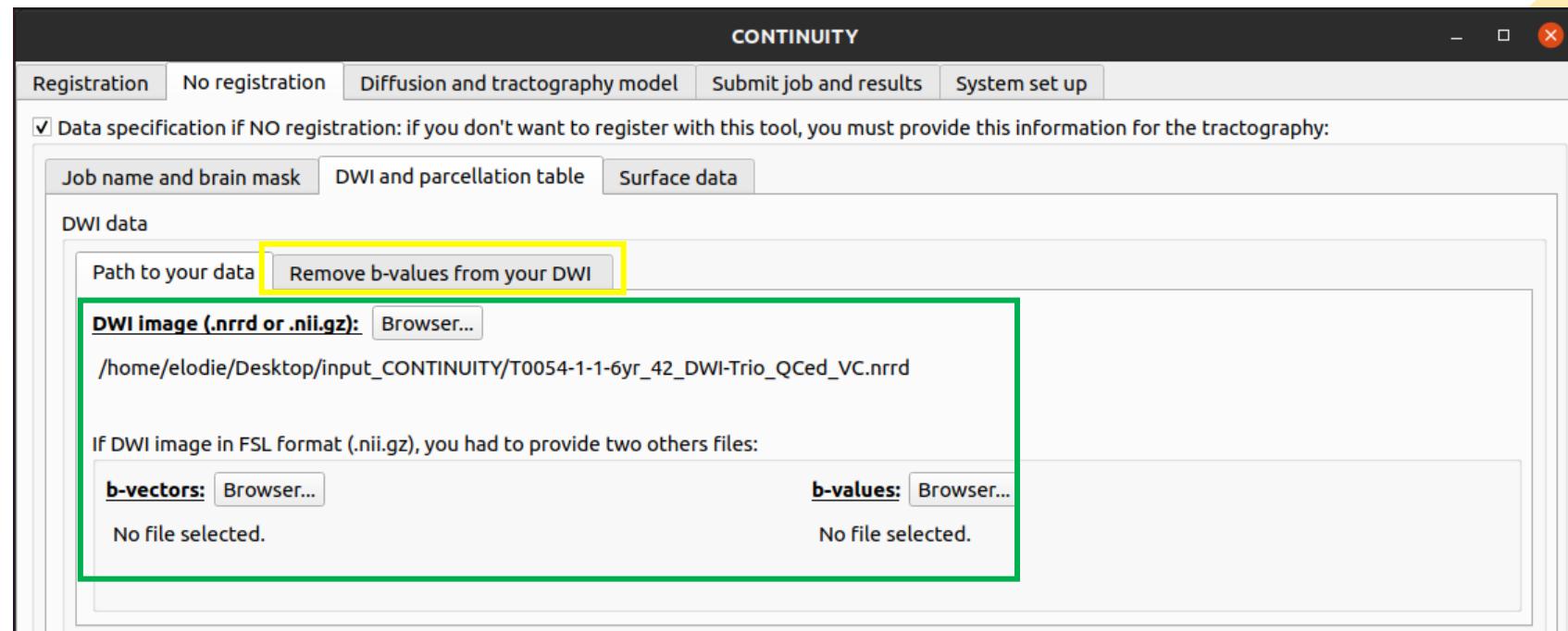
- The up sampling of T1, T2, DWI and brain mask is optional
- ANTS parameters are defined by default as specified
- You can modify it only if you are familiar with this tool
- You can have access to the second metric parameters only if you have provided a 'T2/second modality' path in the 'Data Specification' tab



## Second tab: 'No registration'

→ DWI image can be in .nrrd or in FSL (.nii.gz) format

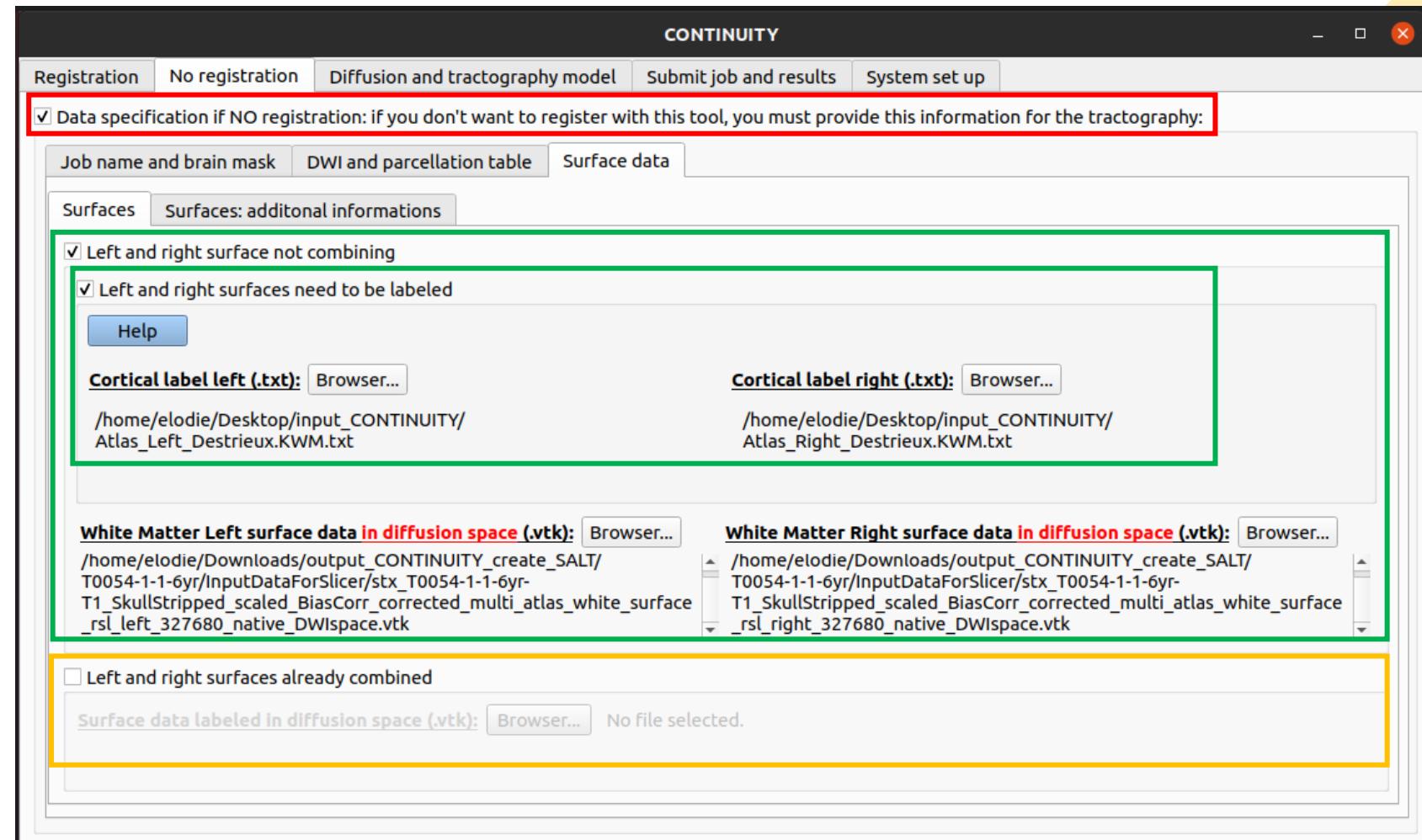
→ In case of a FSL format and multi-shell DWI data, you can select b-values to remove them from your DWI data :



This is a detailed view of the 'Remove bval from your DWI (if multi-shell case)' dialog. It includes a checkbox labeled 'Remove bval from your DWI (if multi-shell case)', a section for selecting b-values from a file (listing 0, 301, 199, 401, 1004), and a field indicating the value '301' is selected for deletion.

## Second tab: 'No registration'

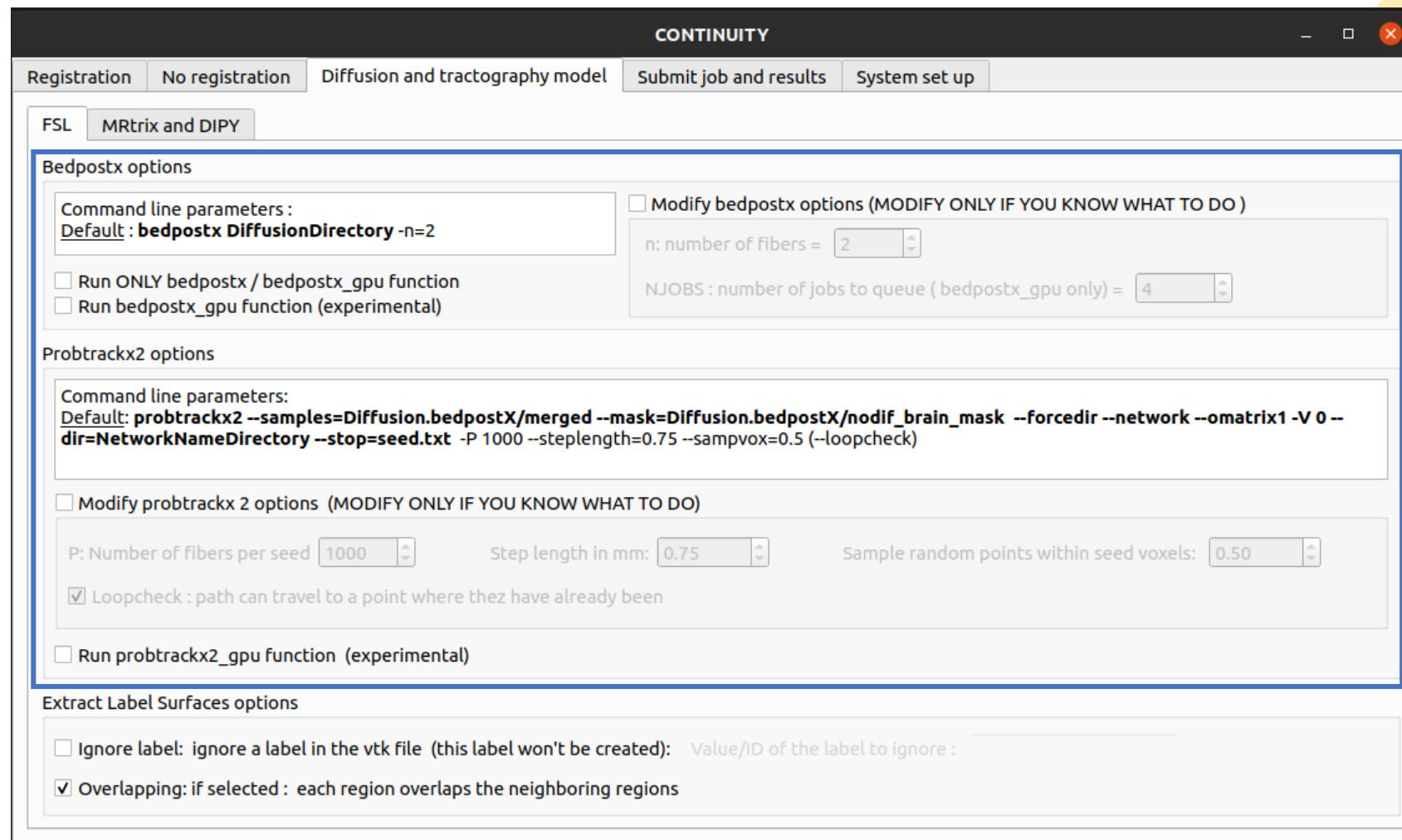
- If you don't want to do the registration with CONTINUITY, you must provide surfaces in DWI space (= diffusion space)
  
- If your surfaces are **not** already combining:
  - If your left and right surfaces need to be labeled, you should provide two additional files: cortical label left and cortical label right
  - In all cases, your left and right surfaces must be in the diffusion space
  
- If your surfaces are already combined, they must be also labeled and in the diffusion space



### Third tab: 'Diffusion and tractography model' → 'FSL'

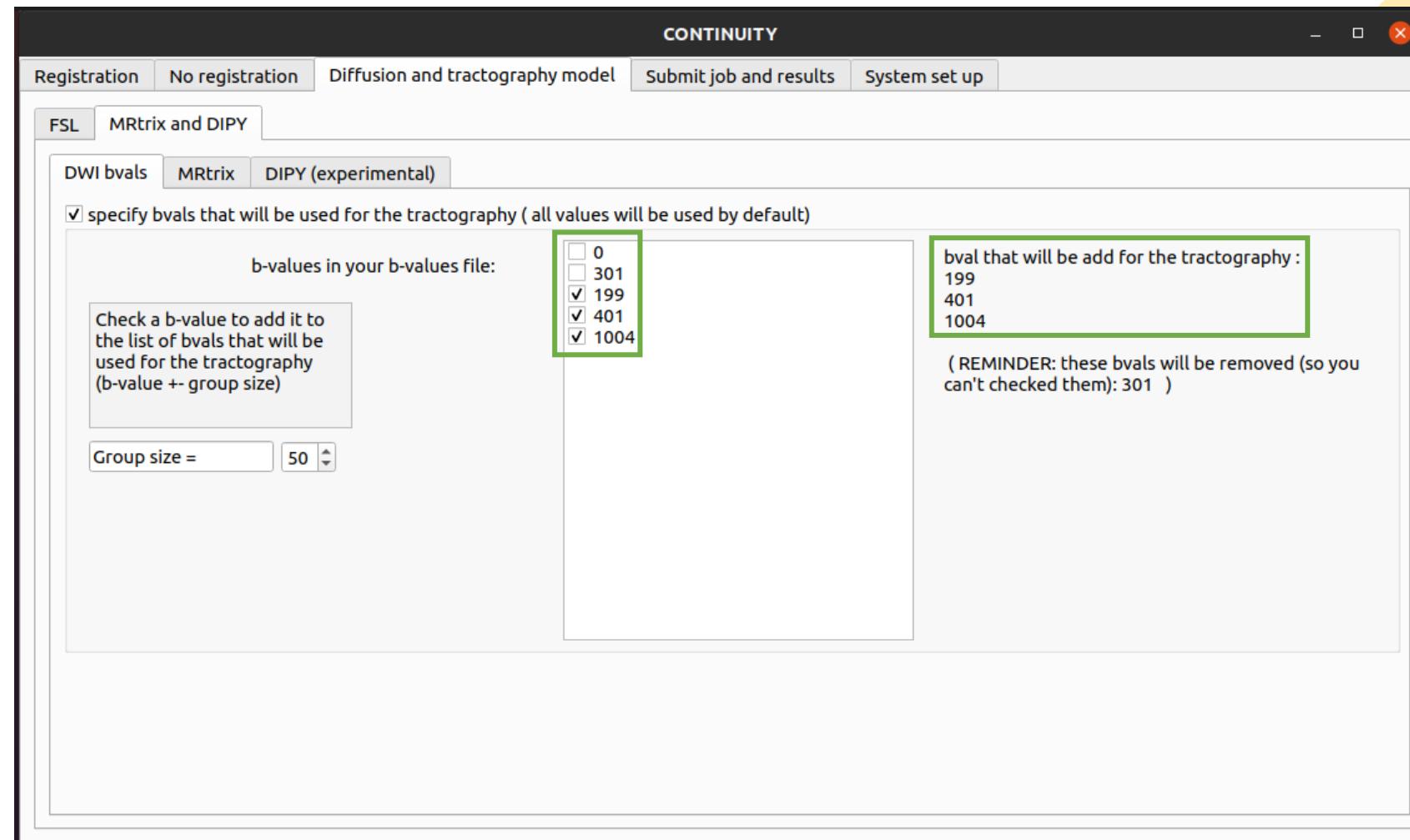
To do tractography with FSL, CONTINUITY will use three different tools

- [Bedpostx](#) and [Probtrackx2](#) parameters are defined by default as specified
- But you can modify it only if you are familiar with this tool



### Third tab: ‘Diffusion and tractography model’ → ‘MRtrix’

CONTINUITY

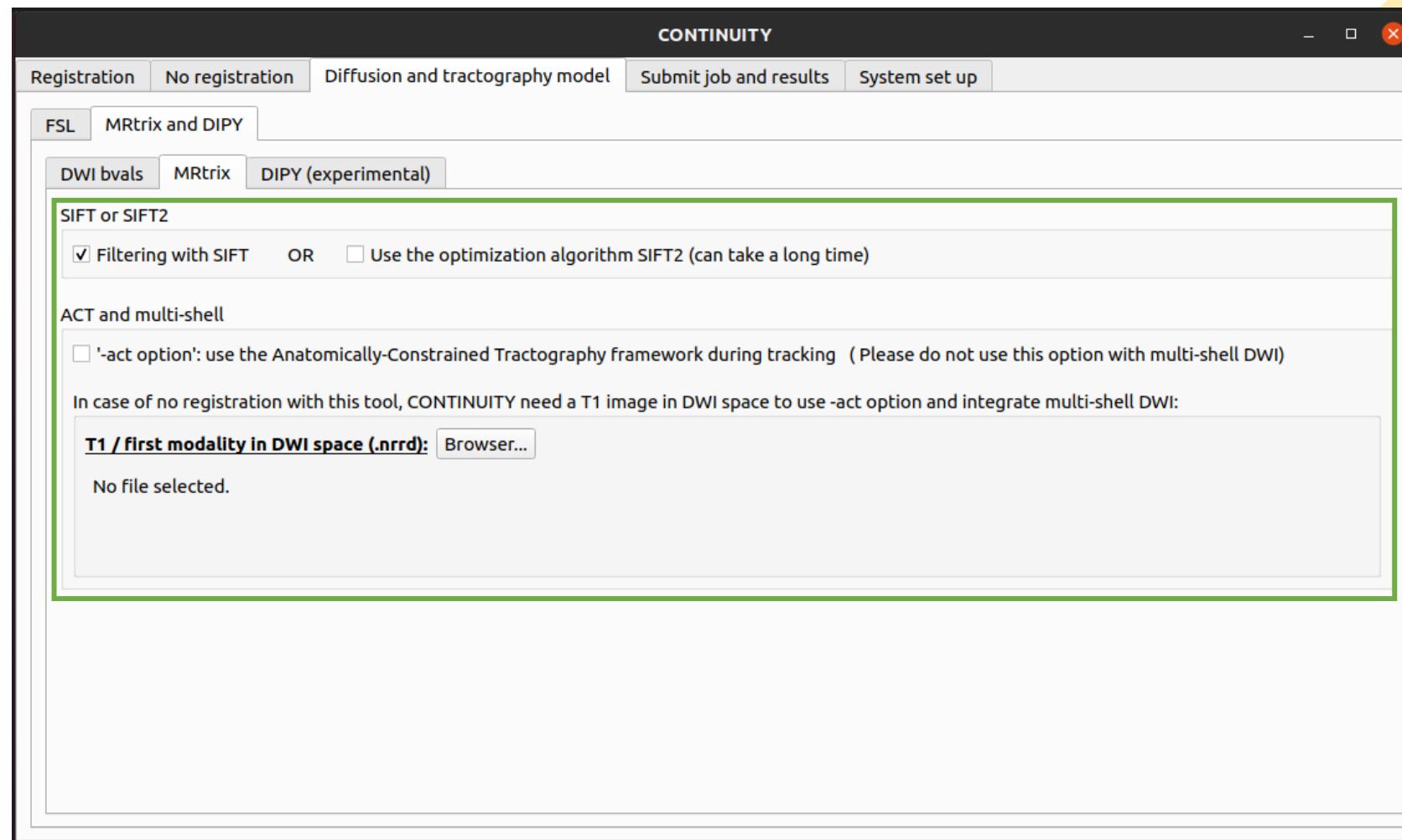


### Third tab: ‘Diffusion and tractography model’ → ‘MRtrix’

CONTINUITY

To do tractography with MRtrix, CONTINUITY will provide different option:

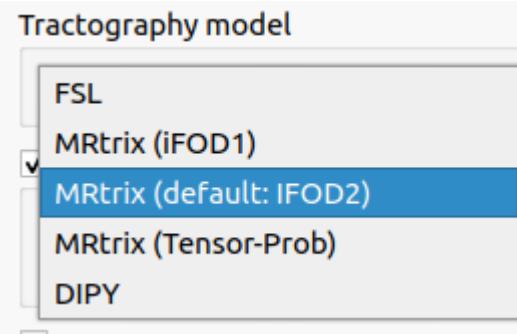
- You can filter (with SIFT) or optimize the tractography (with SIFT2) and add the act option
  
- **SIFT**: filter a whole-brain fiber-tracking dataset such that the streamlines densities match the FOD lobe integrals
  
- **SIFT2**: Optimise per streamline cross-section multipliers to match a whole-brain tractogram to fixel-wise fiber densities



## Fourth tab: ‘Submit job and results’

→ To save your parameters and use it after, you can download a json in a specific folder

→ You can choose the type of algorithm to compute the tractography: (see details below)



→ You can run CONTINUITY locally (by default) or remotely (on Longleaf)

→ You can choose between open Slicer without settings or open the second interface to do the quality control and visualize the connectome

CONTINUITY and MRtrix can perform streamlines tractography with 3 algorithms:

- [iFOD1](#): First-order Integration over Fiber Orientation Distributions
- [iFOD2](#): Second-order Integration over Fiber Orientation Distributions
- [tensor\\_prob](#): Probabilistic Fiber Tracking Using the Wild Bootstrap With Diffusion Tensor MRI

## Last tab: ‘System set up’ → ‘Executables for registration’

unu: up sampling data

N4BiasFieldCorrection: bias correction algorithm

BRAINSFit: register a 3D volume to a reference volume

dtiprocess: handle tensor fields

dtiestim: estimate tensor in a set of DWIs

ANTS: registration of image

ITKTransformTools: basic operation like inverting files

PolydataTransform: transform the landmark

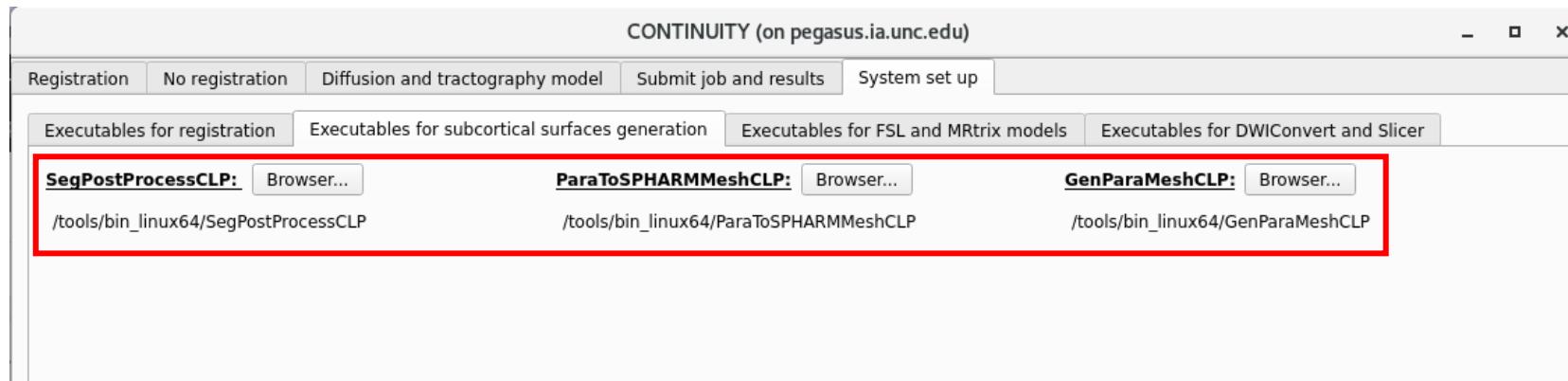
WarpImageMultiTransform: warp an image from a space to another

The screenshot shows the CONTINUITY interface on a web browser. The title bar says "CONTINUITY (on pegasus.ia.unc.edu)". The top navigation bar has tabs: Registration, No registration, Diffusion and tractography model, Submit job and results, System set up, and Executables for registration (which is the active tab). Below the tabs are four rows of tool entries, each with a tool name, a "Browser..." button, and a file path:

- unu:** Browser... /tools/bin\_linux64/unu
- N4BiasFieldCorrection:** Browser... /tools/bin\_linux64/N4BiasFieldCorrection
- BRAINSFit:** Browser... /tools/bin\_linux64/BRAINSFit
- Dtiprocess:** Browser... /tools/bin\_linux64/dtiprocess
- Dtiestim:** Browser... /tools/bin\_linux64/dtiestim
- ANTS:** Browser... /tools/bin\_linux64/ANTS
- ITKTransformTools v1.2.3:** Browser... /tools/bin\_linux64/ITKTransformTools\_v1.2.3
- Polydatatransform v1.2.1:** Browser... /tools/bin\_linux64/polydatatransform\_v1.2.1
- WarpImageMultiTransform:** Browser... /tools/bin\_linux64/WarpImageMultiTransform

## Last tab: 'System set up' → other executables

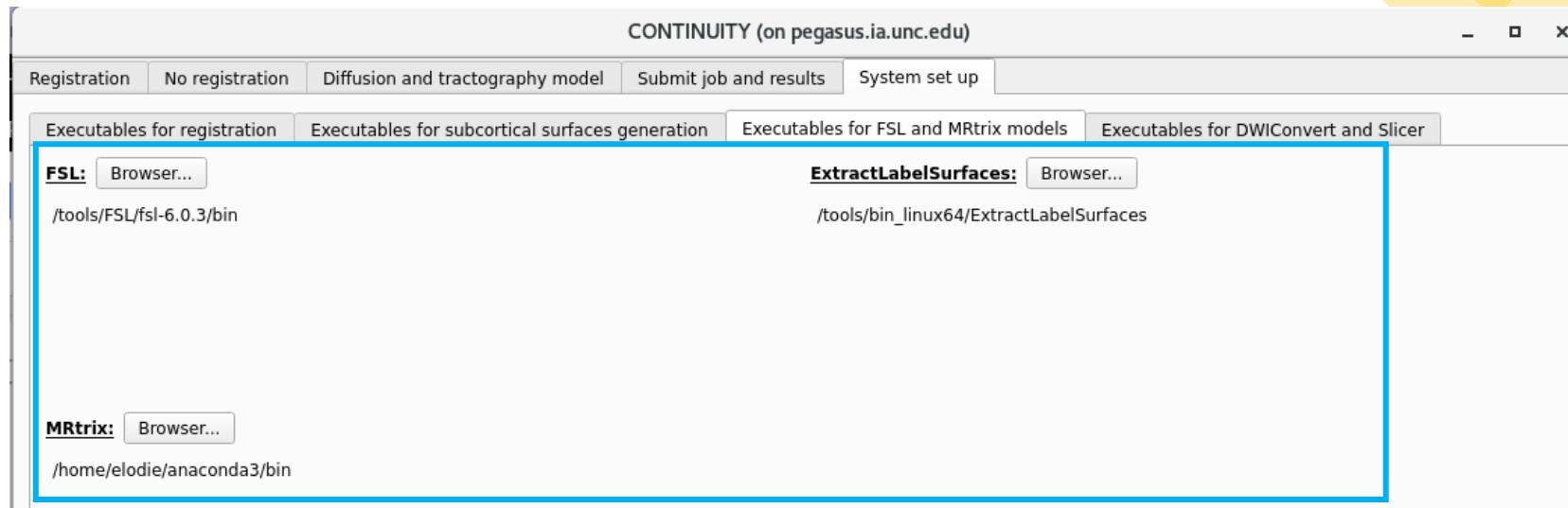
**SegPostProcessCLP:** post processing segmentation  
**GenParaMeshCLP:** generate Mesh Parameters  
**ParaToSPHARMMeshCLP:** parameters to SPHARM Mesh



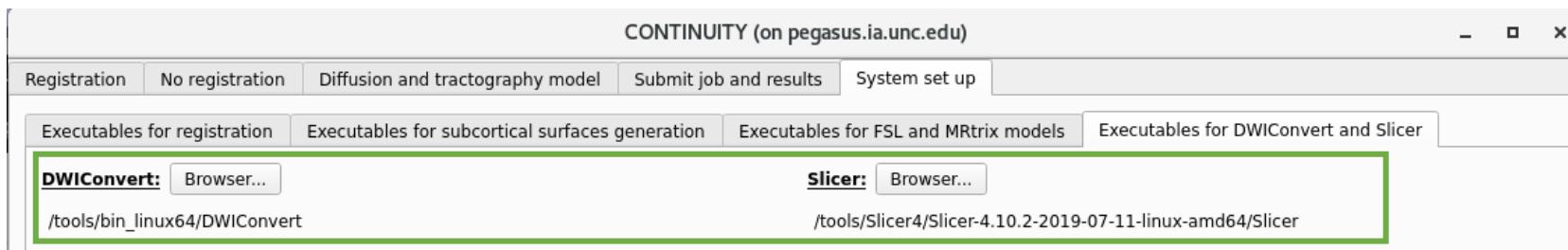
FSL:  
 → [Bedpostx](#): fitting the probabilistic diffusion model  
 → [Probtrackx2](#): compute tractography

[ExtractLabelSurfaces](#): creation label surface from a vtk surface containing label information

[MRtrix](#): advanced tool for the analysis of diffusion MDI data



[DWICreate](#): wrapped executable  
[Slicer](#): image computing platform



# Tutorial **CONTINUITY**

**CONTINUITY visualization**

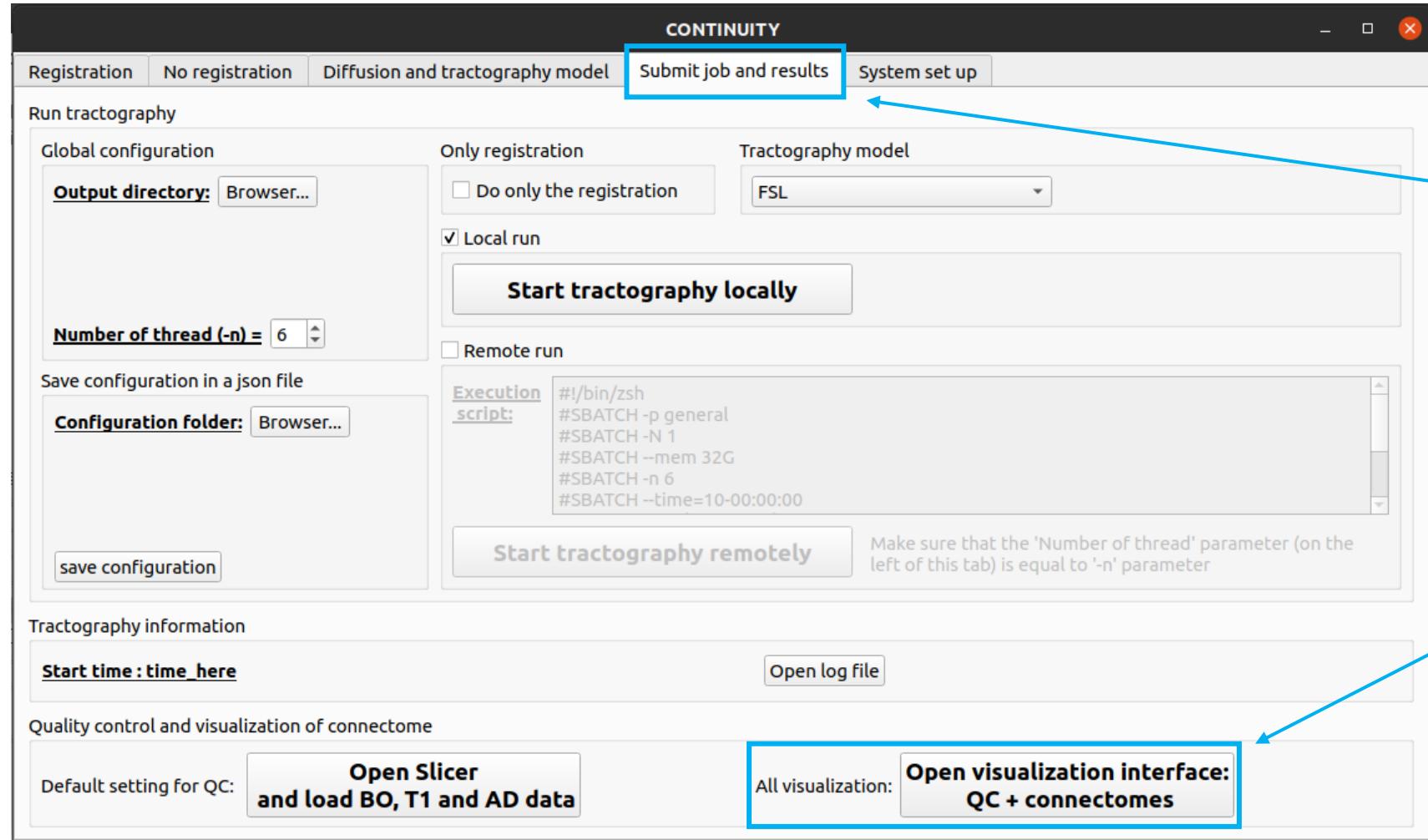
1. Quality control with Slicer

2. Connectivity matrix  
and circle connectome

3. Brain connectome  
in 2D and 3D

## Two options to open the second interface

### First option: using the first interface



This tab

Click here

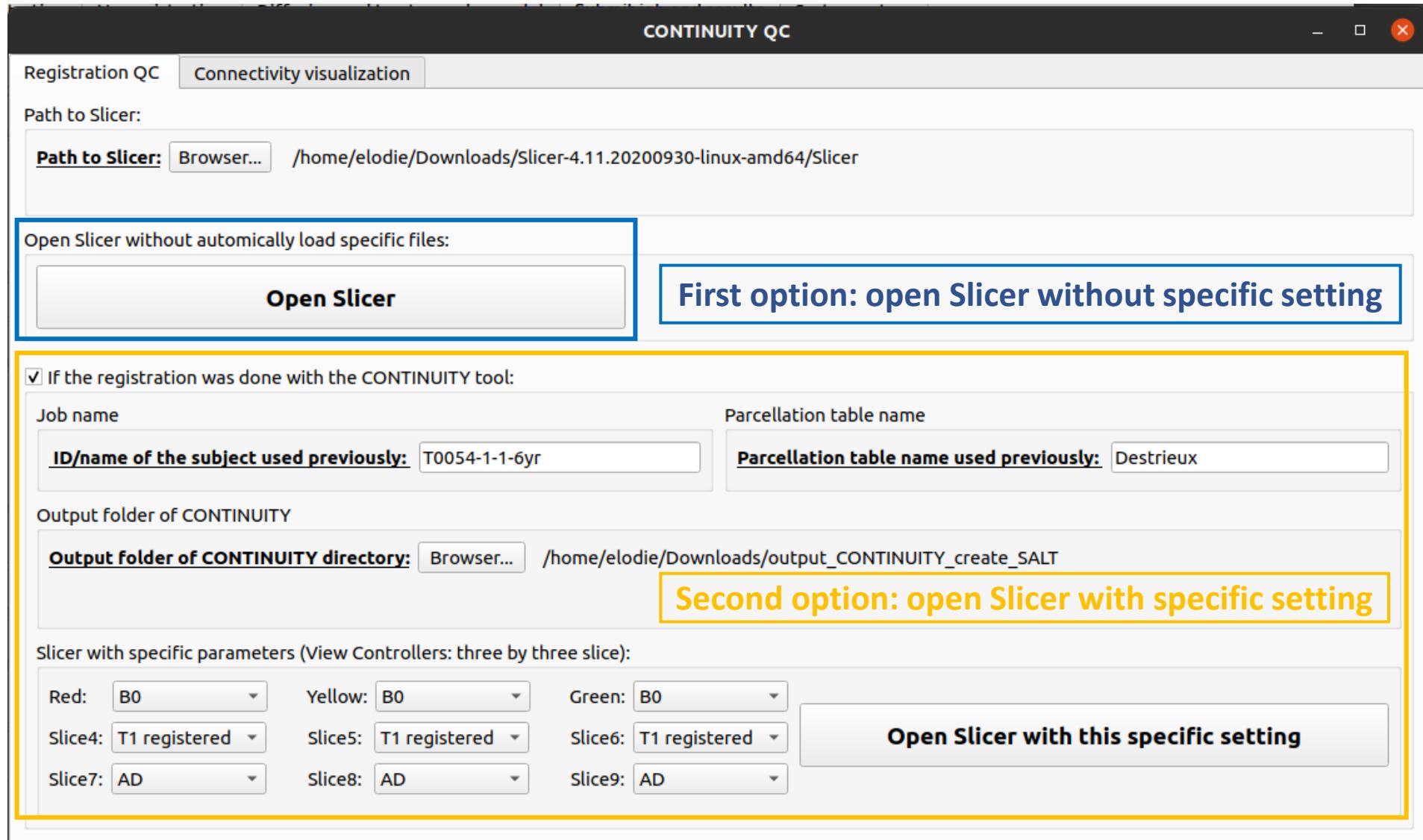
### Second option: using the command line

On Longleaf:

`python3 /proj/NIRAL/tools/CONTINUITY/CONTINUITY_v1.1/CONTINUITY_QC/main_interface_visualization.py`

## First tab: 'Registration QC'

Two options to open Slicer to do the quality control:



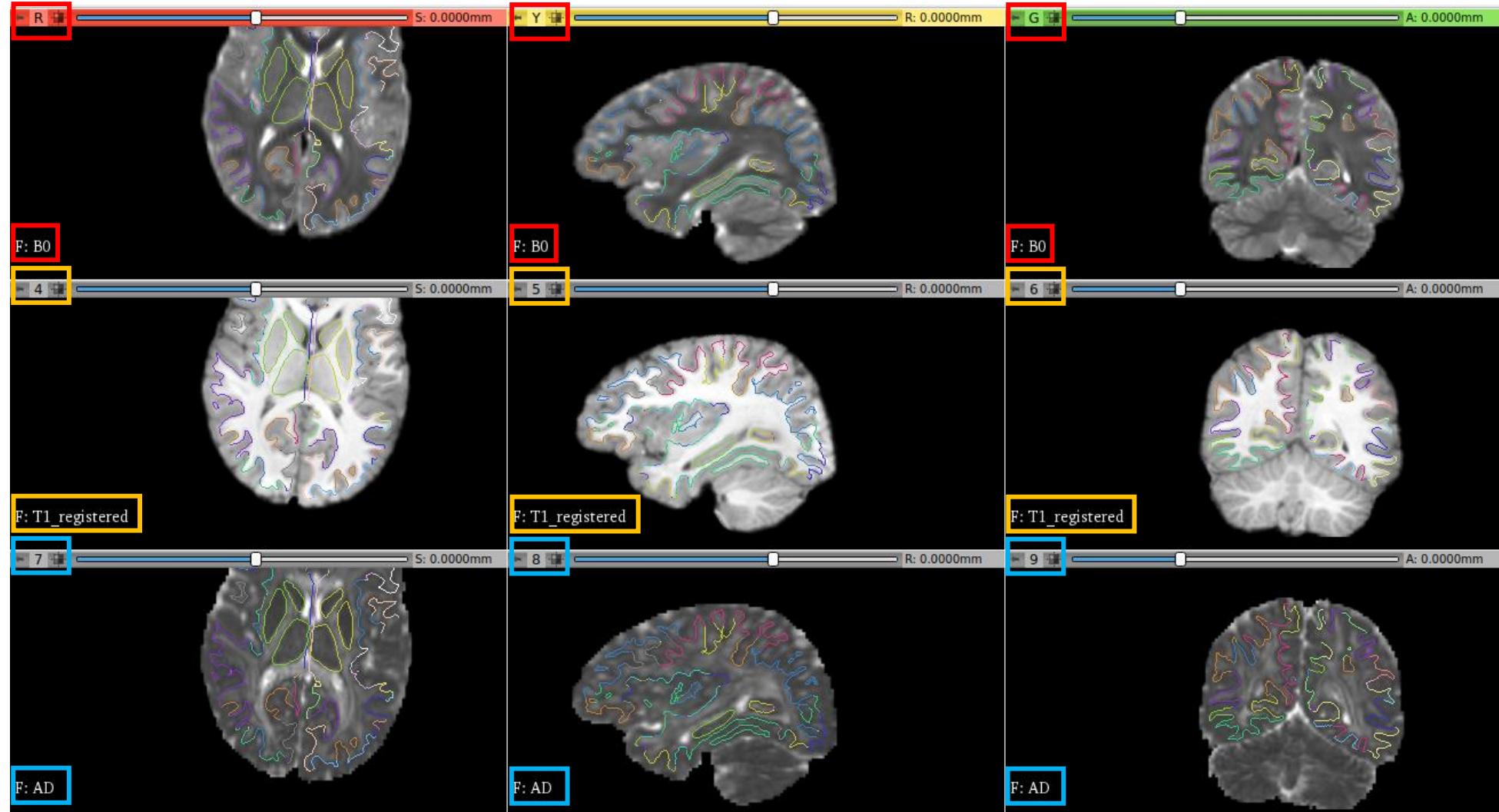
# First tab: 'Registration QC'

Zoom on the option 'Quality control with Slicer and specific parameters':

Parameters on the Interface:

Slicer with specific parameters (View Controllers: three by three slice):

Red: B0	Yellow: B0	Green: B0
Slice4: T1 registered	Slice5: T1 registered	Slice6: T1 registered
Slice7: AD	Slice8: AD	Slice9: AD



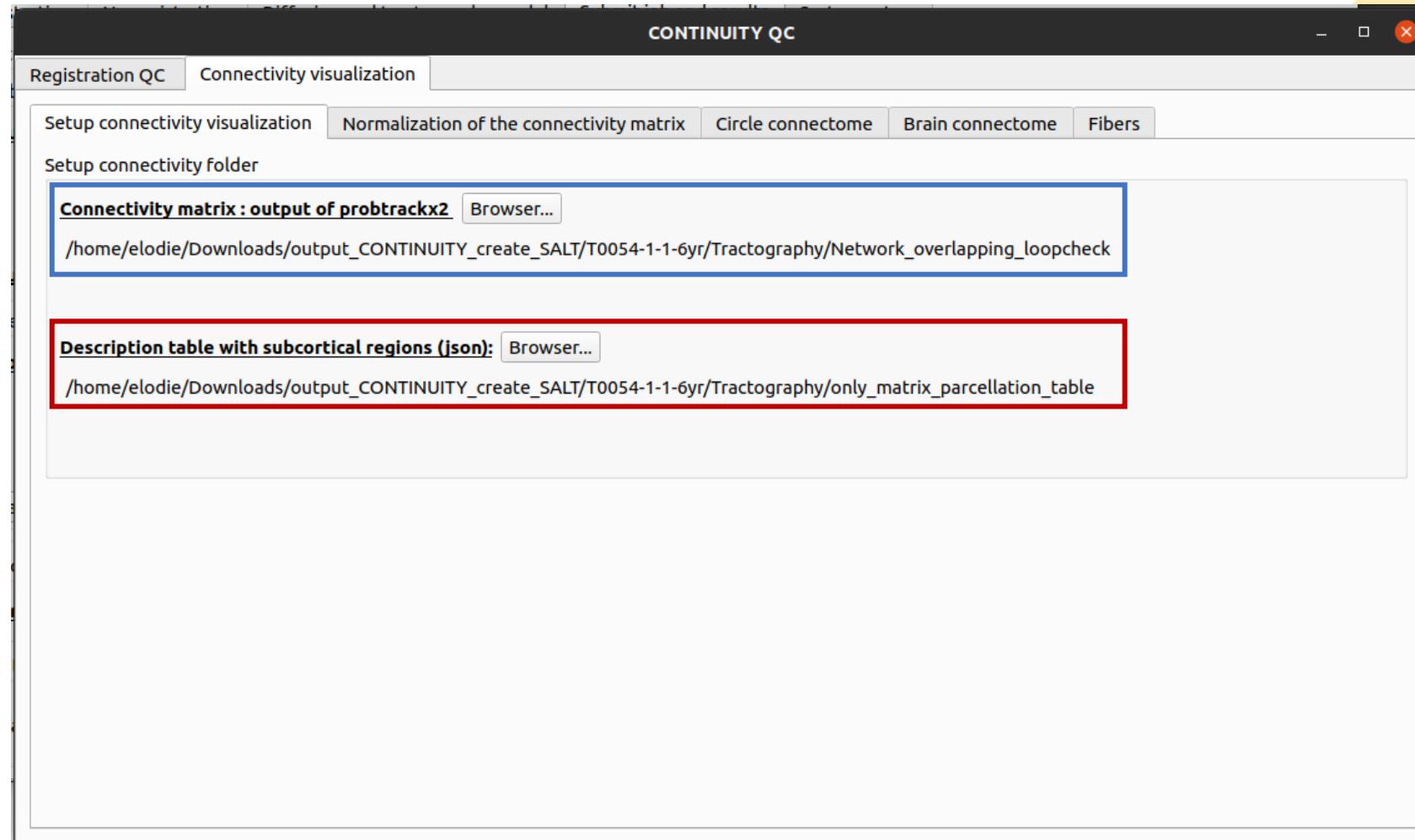
## Second tab: ‘Connectivity visualization’ → ‘Normalization of the connectivity matrix’

CONTINUITY

To be able to display the connectivity matrix, the circle connectome and the brain connectome, CONTINUITY need two files:

- First you can select the directory containing your connectivity matrix
- The name of this file must be: “fdt\_network\_matrix”

- Then you can select the parcellation table with subcortical regions integrated in the tractography pipeline



## Second tab: 'Connectivity visualization' → 'Normalization of the connectivity matrix'

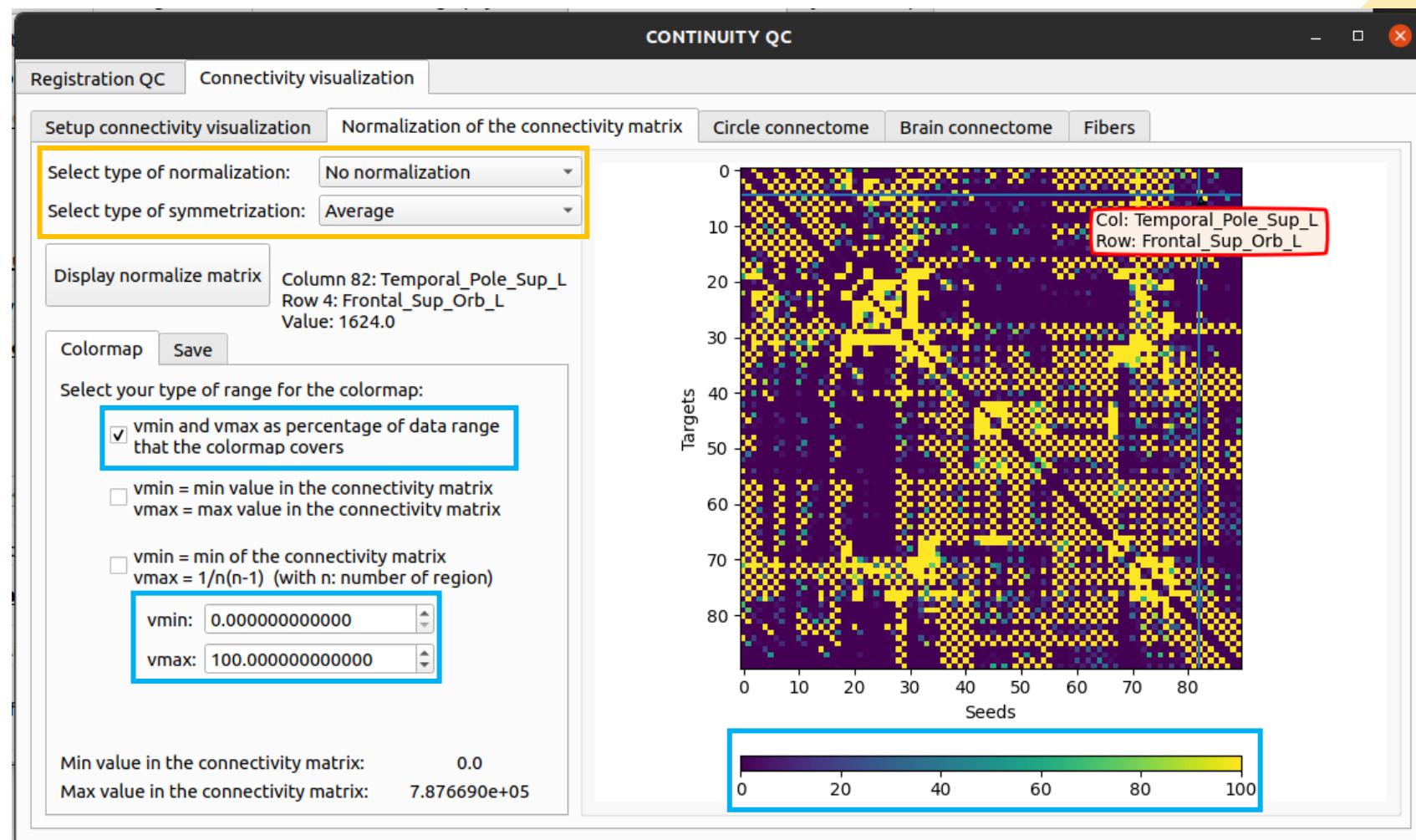
→ You can normalize and symmetrize your connectivity matrix:

Select type of normalization:	No normalization
Select type of symmetrization:	Whole normalization
	Row region normalization
	<b>Row column normalization</b>
Select type of symmetrization:	Average
	Maximum
	Minimum

→ If you change some parameters after clicking on the button, the connectivity matrix will be updated automatically

For the colormap you can select 3 different types of range:

→ The first option means that the color is a percentage of the data range covering by the colormap

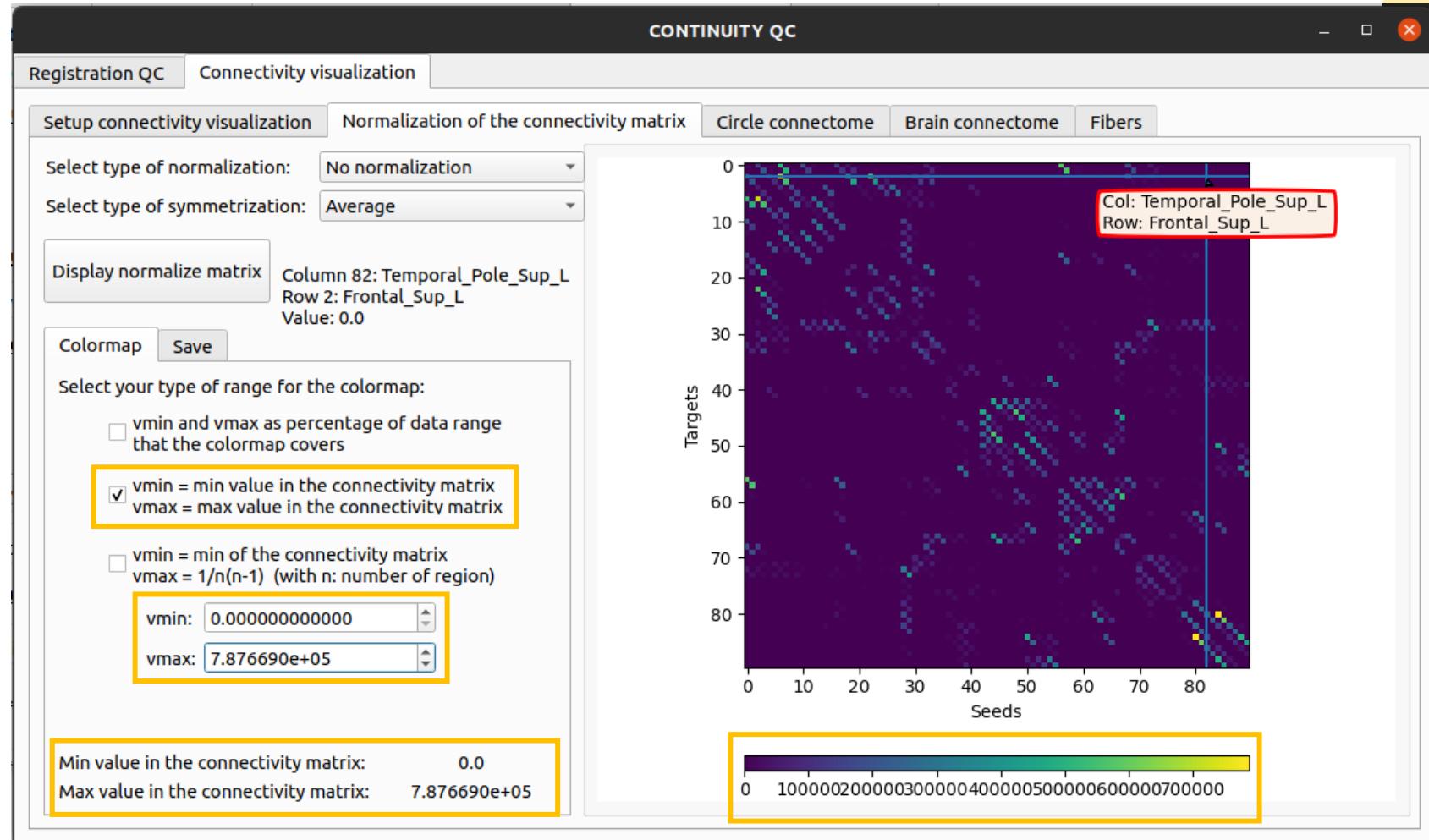


## Second tab: 'Connectivity visualization' → 'Normalization of the connectivity matrix'

CONTINUITY

For the colormap you can select 3 different types of range:

- The second option takes the min and the max of your connectivity matrix and used these two values as the min and max of the range of your colormap



## Second tab: 'Connectivity visualization' → 'Normalization of the connectivity matrix'

For the colormap you can select 3 different types of range:

- The third option displays the colormap using the number of region.

$v_{min} = \min$  of the connectivity matrix  
 $v_{max} = 1/n(n-1)$  (with n: number of region)

$v_{min}: 0.000000000000$

$v_{max}: 787669.000000000000$

Please select 2 values between 0.0 to 0.0001248

- The second sub tab "save" allows you to download the connectivity matrix:

Colormap

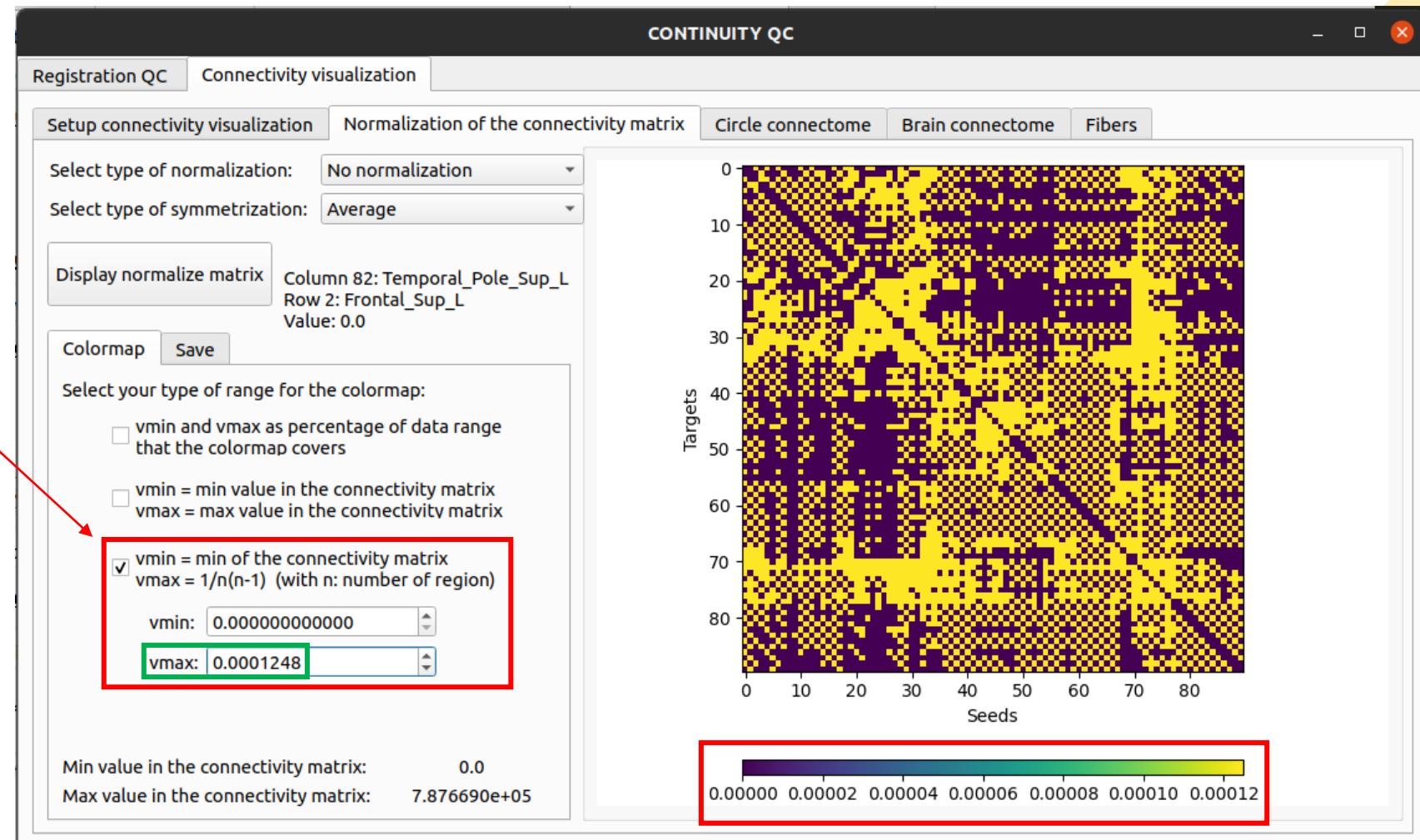
Save

Configuration folder: Browser...

/output\_CONTINUITY

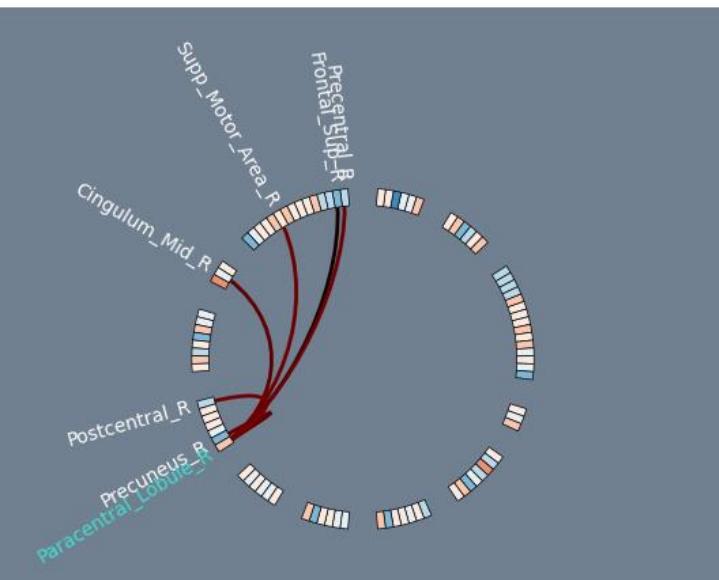
Save

Figure saved here: ./output\_CONTINUITY/  
 Connectivity\_matrix\_normalized\_No  
 normalization\_symmetrization\_by\_Average.pdf

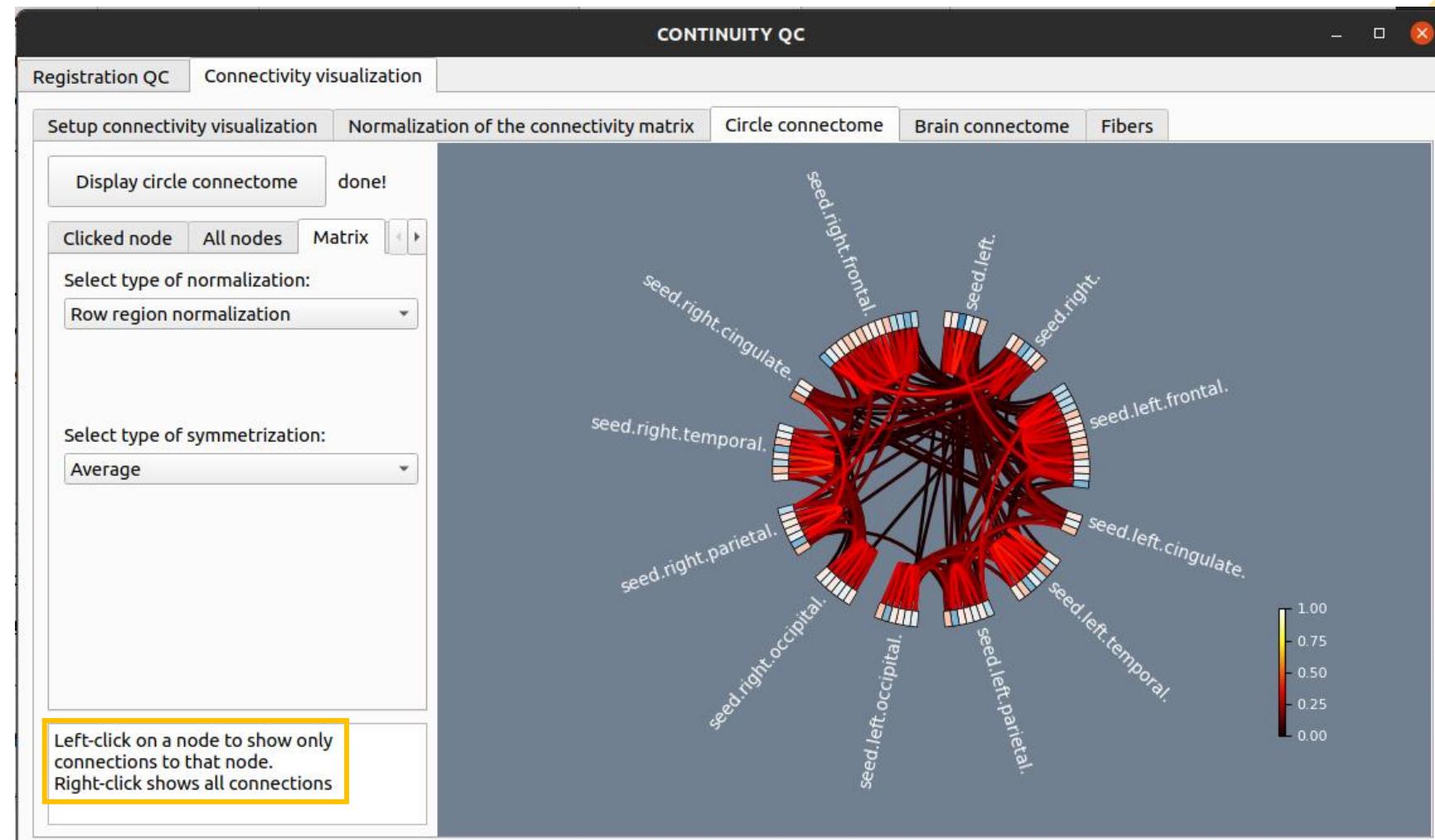
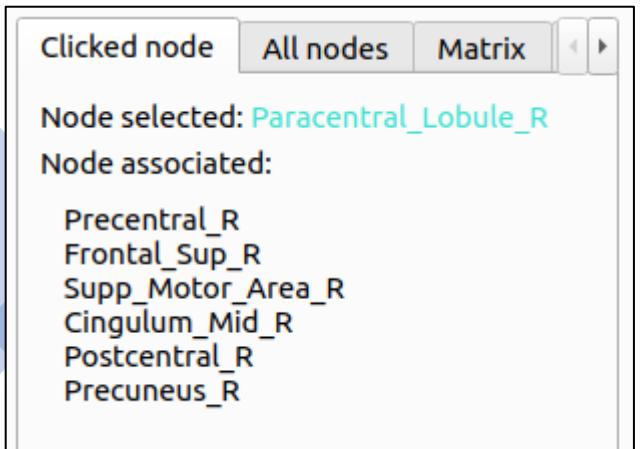


## Second tab: ‘Connectivity visualization’ → ‘Circle connectome’

- If you do a left-click on a node you can visualize only connections to that node:

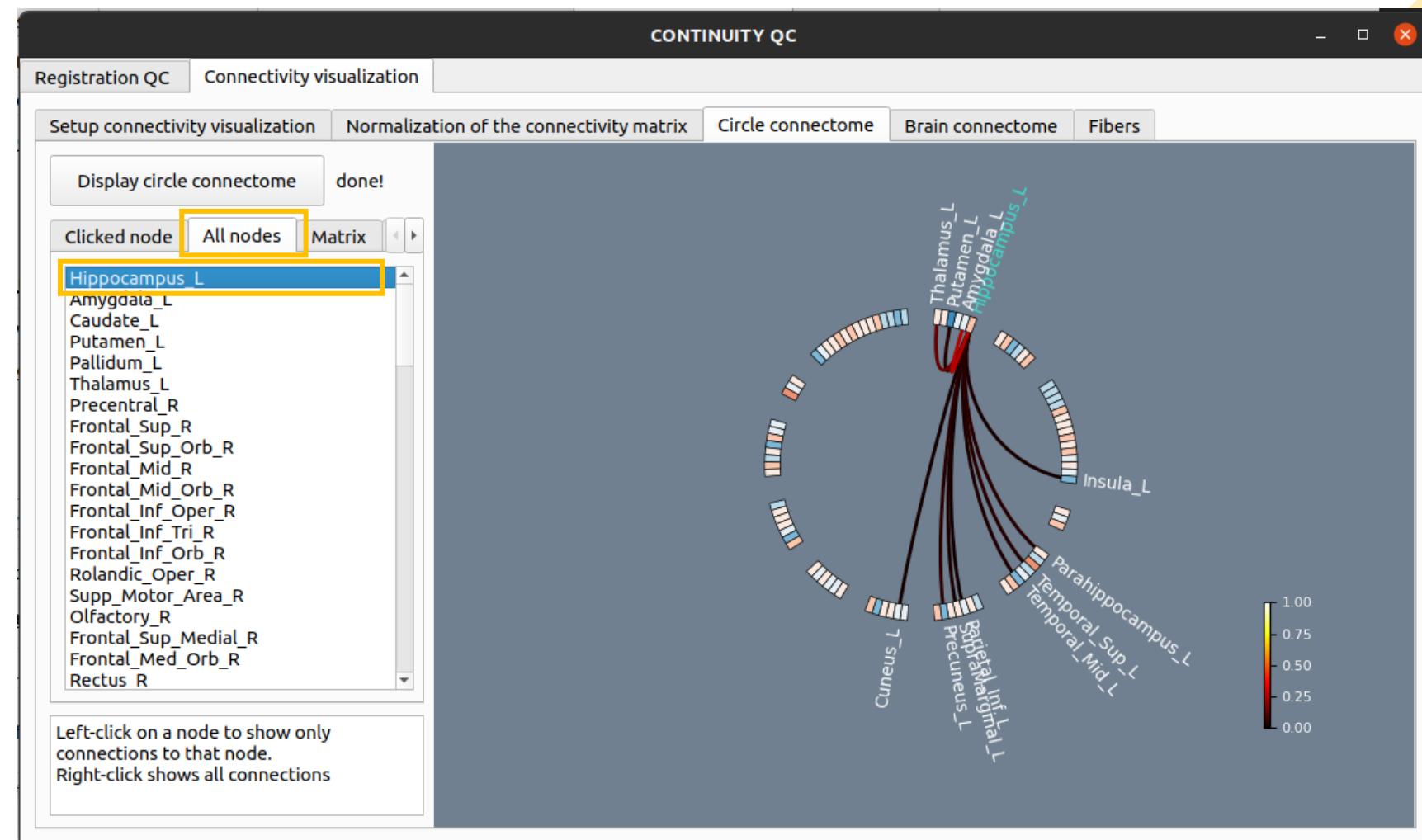


- You can have more information about the connections of this node:



## Second tab: ‘Connectivity visualization’ → ‘Circle connectome’

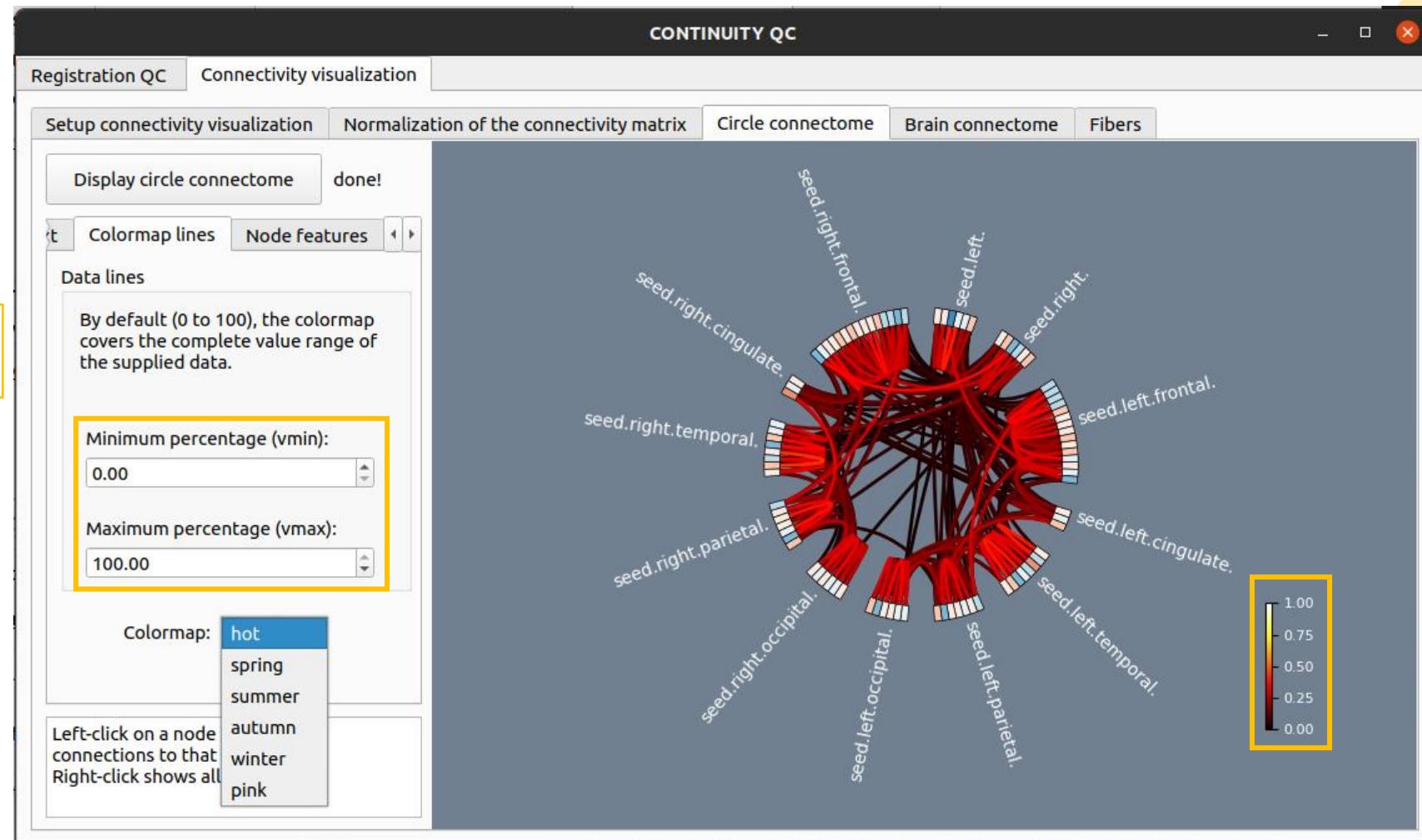
→ You can use the list to select a node:



## Second tab: ‘Connectivity visualization’ → ‘Circle connectome’

CONTINUITY

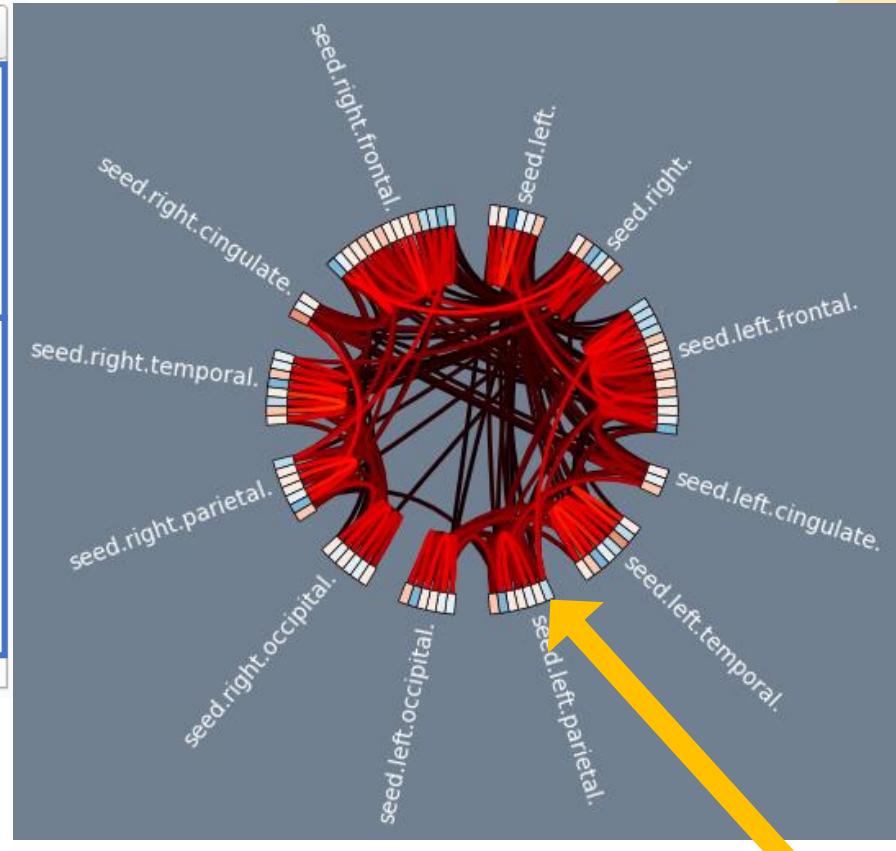
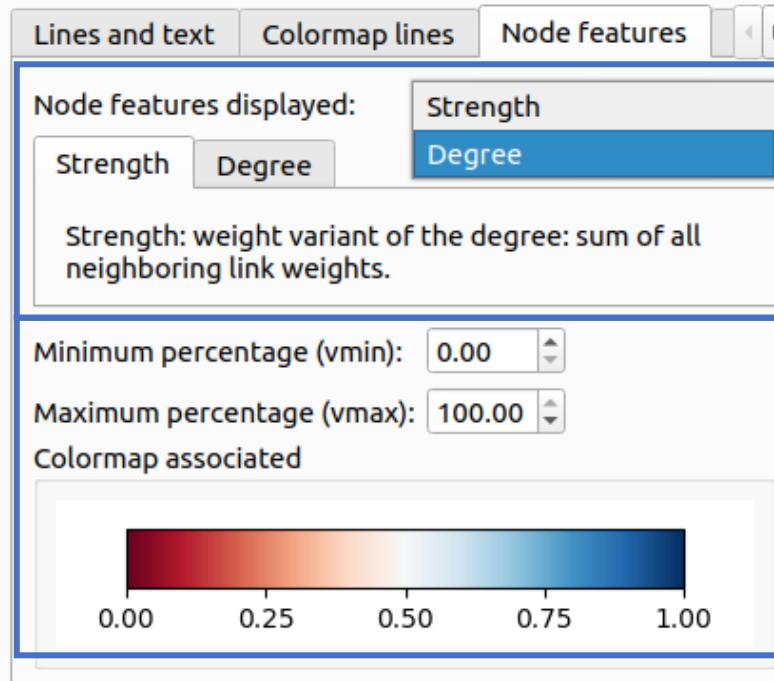
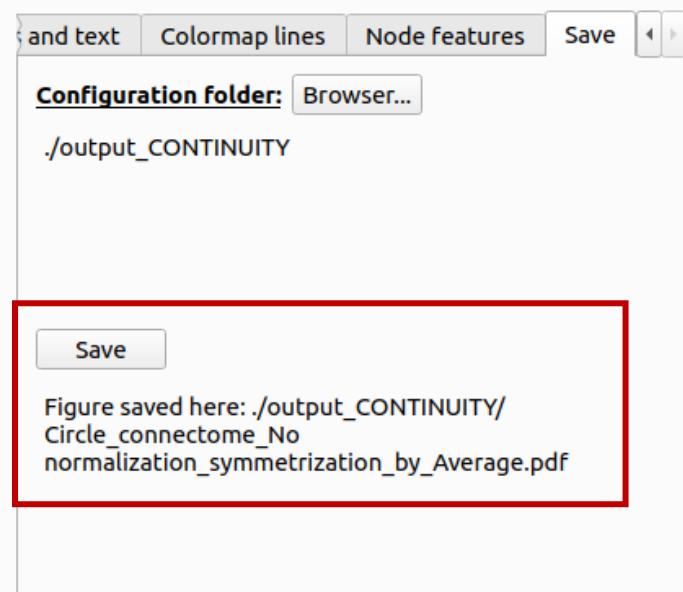
- On the sub-tab “colormap lines” you can change the range (in %) of the colormap.



## Second tab: ‘Connectivity visualization’ → ‘Circle connectome’

- On the third sub-tab “**Node features**”, you can select and display two different features: strength and degree
- The selected feature has its own colormap and you can change its range (in %)

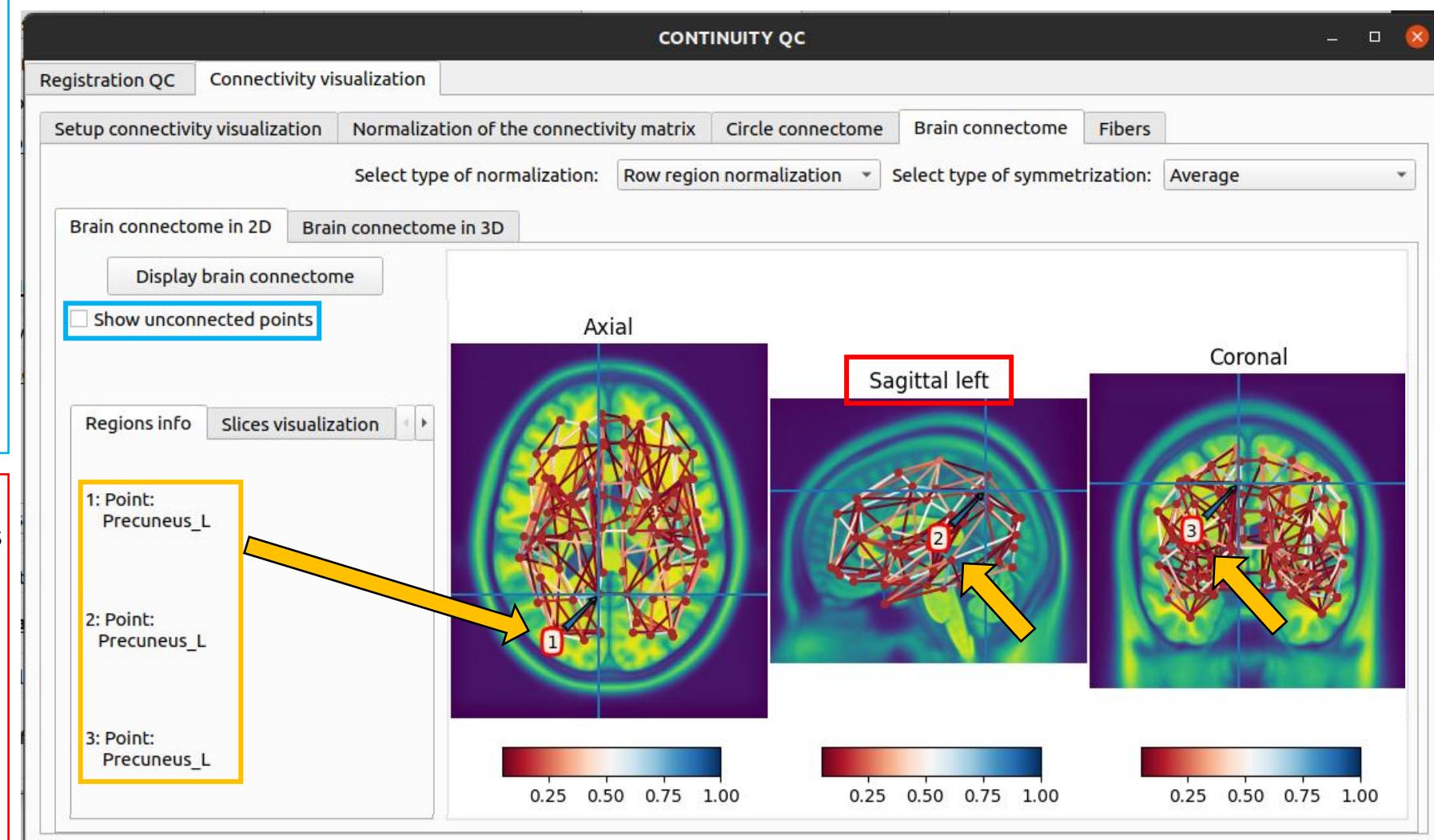
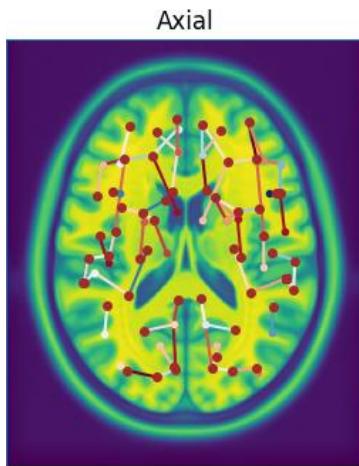
→ You can download the circle connectome:



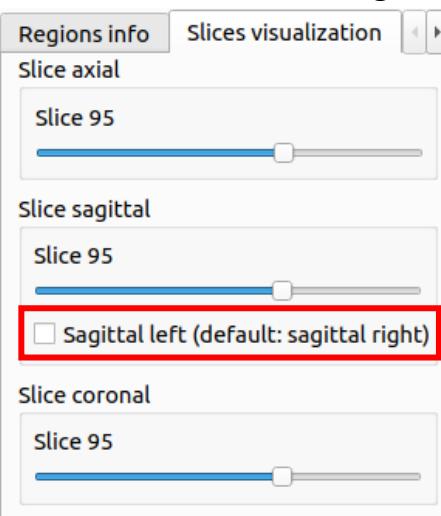
Node

## Second tab: ‘Connectivity visualization’ → ‘Brain connectome’ → ‘Brain connectome in 2D’

- You can display unconnected points (if you have changed the range of the colormap):



- The tab “Slices visualization” modifies the background (brain surfaces) of each planes  
 → You can choose to see the sagittal left

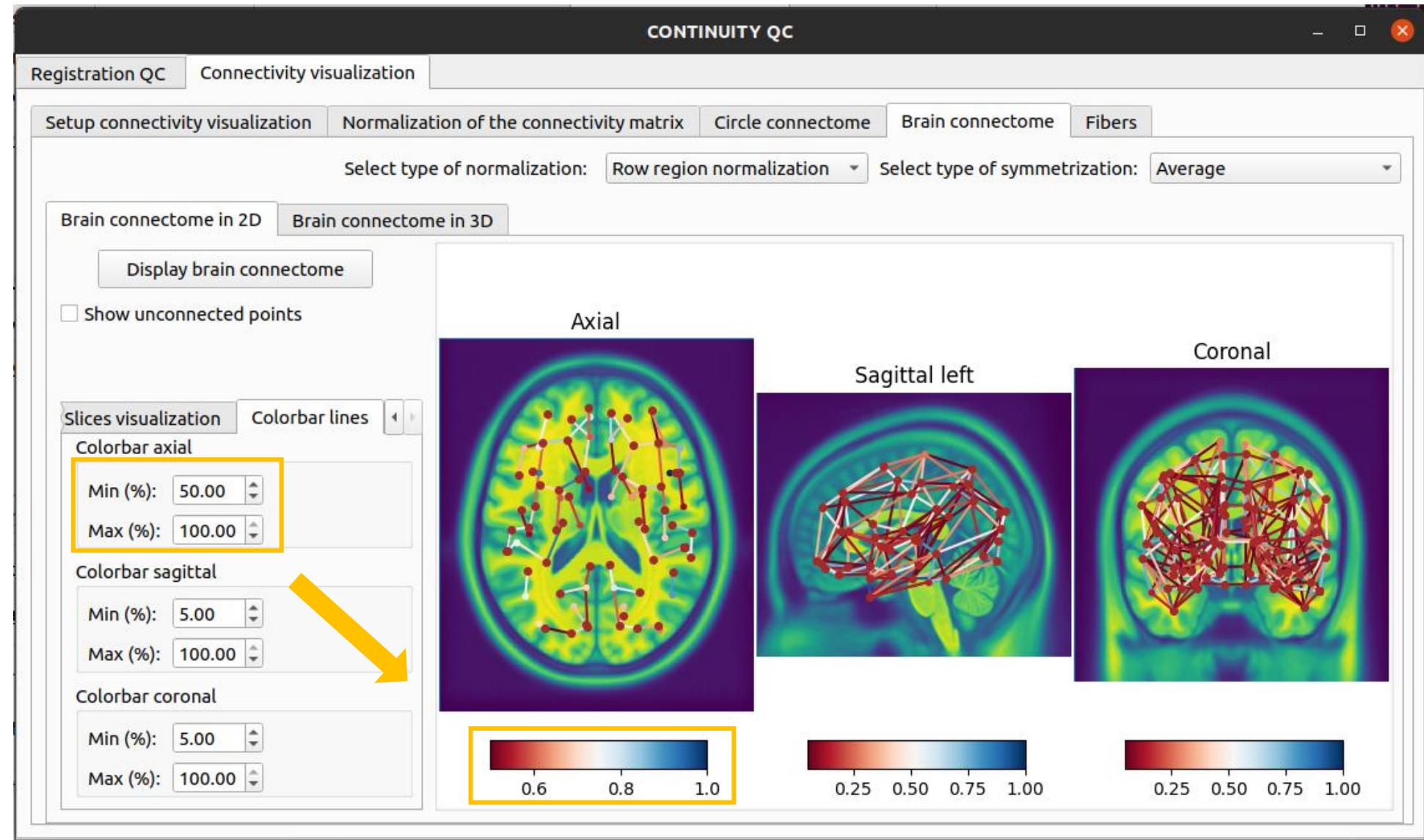


- You can click right on a node or a line to have more information (click left on a view to remove labels of this view)

## Second tab: ‘Connectivity visualization’ → ‘Brain connectome’ → ‘Brain connectome in 2D’

CONTINUITY

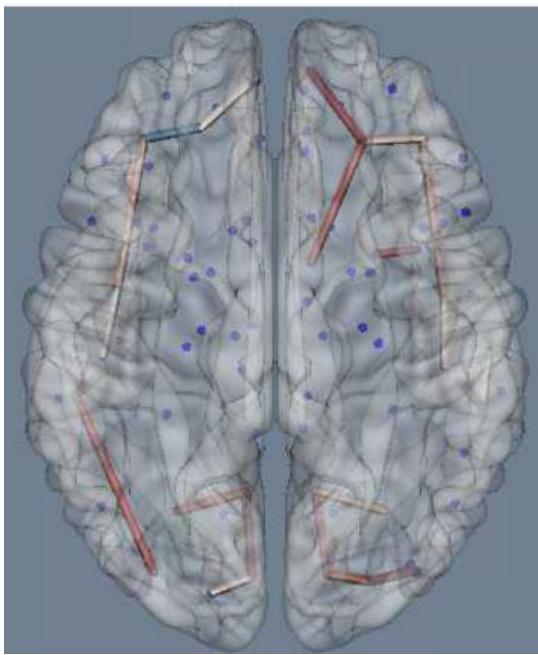
→ The tab “Colorbar lines” changes the range for each planes



## Second tab: ‘Connectivity visualization’ → ‘Brain connectome’ → ‘Brain connectome in 3D’

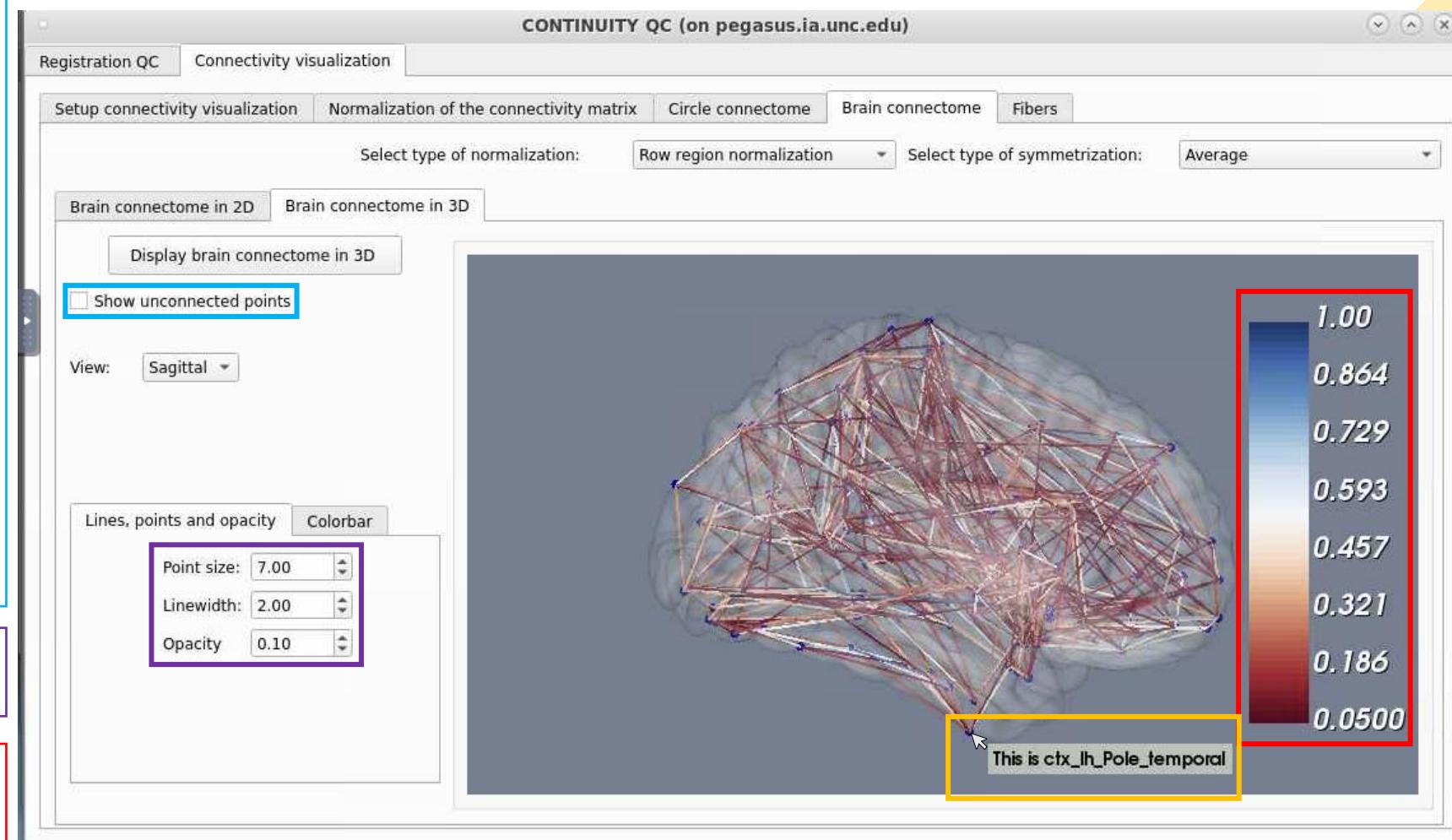
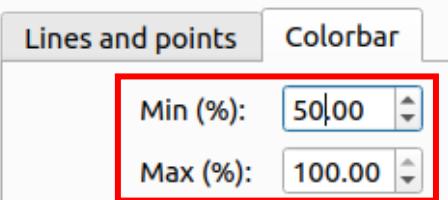
CONTINUITY

- You can display unconnected points (if you have changed the range of the colormap):



- In the first sub-tab “Lines and points”:  
→ Automatically update some parameters

- In the second sub-tab you can change the range of the colormap:



- You can move your mouse on a node or a line to have more information

## Second tab: 'Connectivity visualization' → 'Fibers'

You can select a .vtk file and display it:

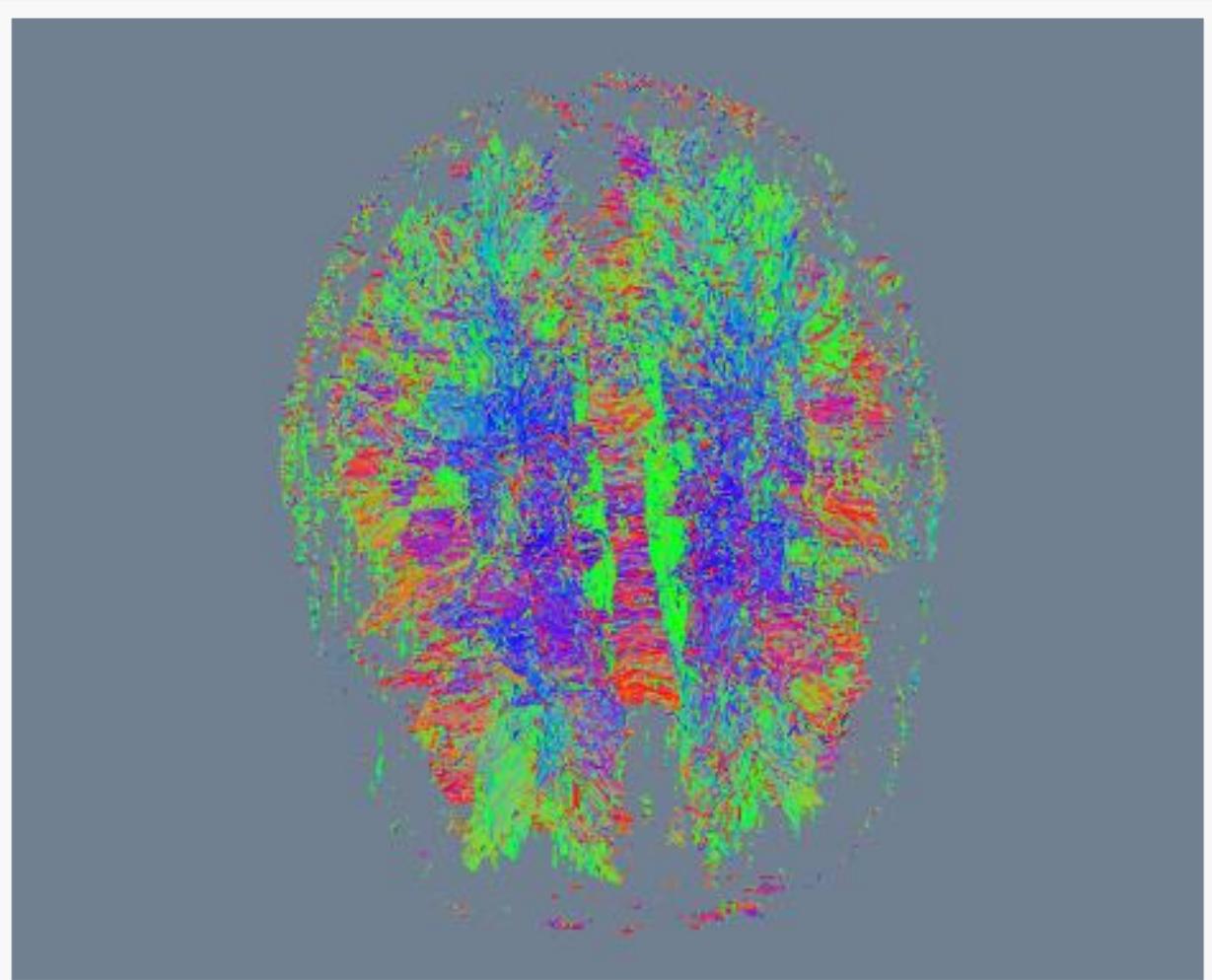
Setup connectivity visualization   Normalization of the connectivity matrix   Circle connectome   Brain connectome   Fibers

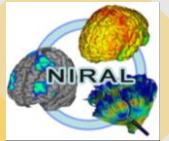
Select fiber:

**Configuration file:** [Browser...](#)

```
/home/elodie/Downloads/output_CONTINUITY_DIPY/  
T0054-1-1-6yr/Tractography/DIPY/tractogram.vtk
```

**Display**





# Tutorial CONTINUITY

Acknowledgement to :  
Martin Styner, Juan Carlos Prieto  
and Maria Bagonis

