

DTI Pre-processing Pipeline

Step – 1 : Converting DICOM Data to NIfTI format

DICOM, the raw image data acquired from MRI scanners (diffusion-weighted image (DWI) volumes) is converted to NIfTI format that can be processed.

Command: **dcm2nii <options> <sourcenames>**

dcm2nii datafolder_path/DICOM/

Output: Returns compressed data in NIfTI format with nii.gz suffix in the filename.

E.g. : dwi.nii.gz

Step – 2: CONVERTING TO MRTRIX FORMAT (.mif - MRTrrix Image Format)

The data thus converted to NIfTI format is collated with b-values and b-vectors to give the Diffusion Weighted Image in .mif format

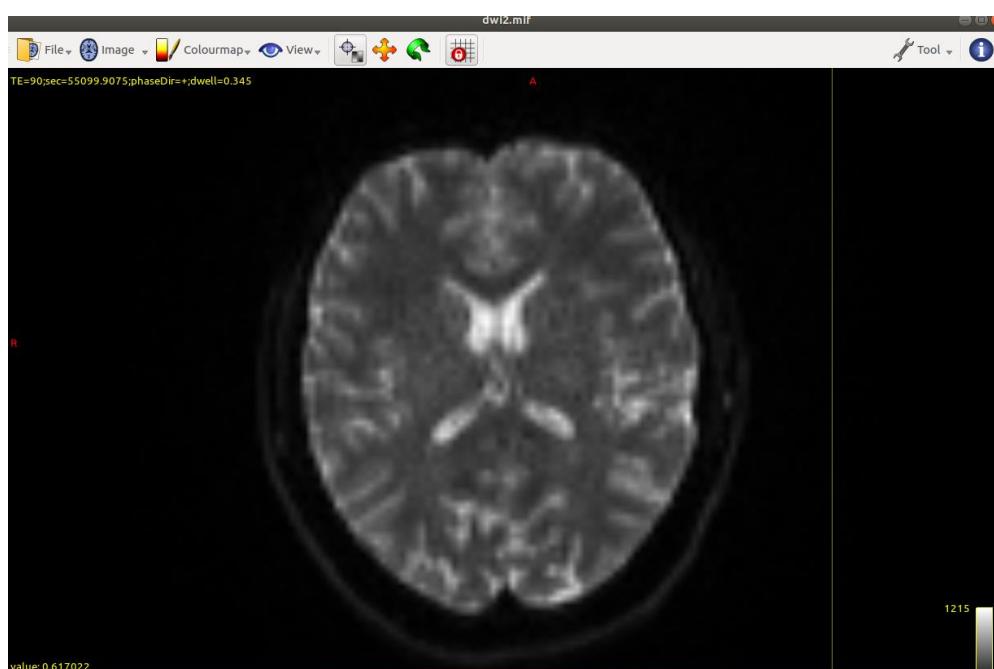
Command : mrconvert [options] input output

mrconvert RAW/dwi2.nii.gz -fslgrad RAW/bvecs RAW/bvals dwi.mif

RAW/dwi2.nii.gz : input file

-fslgrad RAW/bvecs RAW/bvals : Provides the diffusion-weighted gradient scheme used in the acquisition in FSL bvecs/bvals format files

Dwi.mif : output file

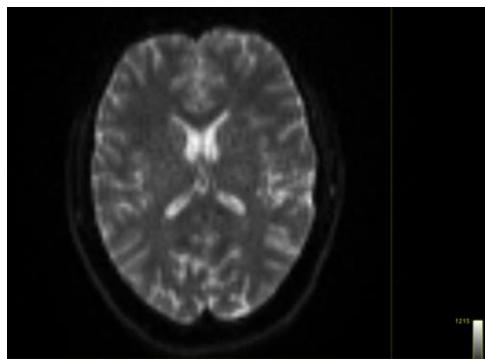


Step – 3 : De-noising

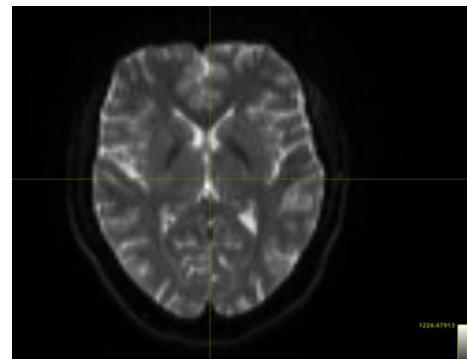
To remove the noise from DTI data. The 'dwidenoise' estimates the noise map based on optimal threshold for Principal component analysis(PCA) and exploiting the data redundancy in the PCA domain.

Command : dwidenoise [options] dwi_input denoised_output

```
dwidenoise -force -noise noise.mif dwi.mif dwi_denoised.mif
```



RAW data : dwi2.mif



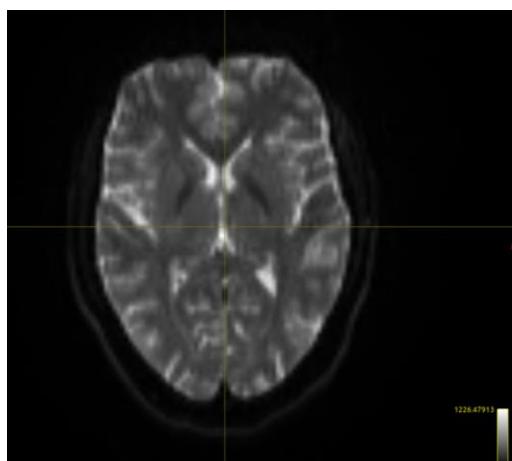
Denoised data: dwi_denoised2.mif

Step – 4: Removing Gibbs Artefact

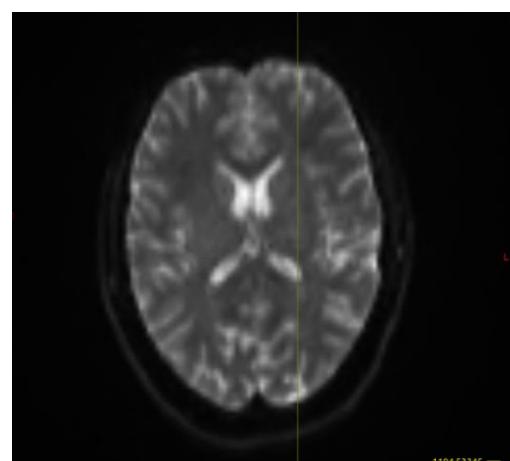
Gibbs artefact, also known as ringing artefact, typically appear as multiple fine parallel lines immediately adjacent to high-contrast interfaces, as a consequence of using Fourier transforms to reconstruct MR signals into images.

Command: mrdegibbs [options] input_file output_file

```
mrdegibbs -force dwi_denoised.mif dwi_degibbs.mif
```



Denoised data: denoised.mif



After removing Gibbs artefact: degibbs.mif

Step – 5: DWI distortion correction using FSL preprocessing

Performs diffusion image pre-processing using FSL's eddy tool that accounts for eddy correction as well as susceptibility induced distortion correction

Command: dwifslpreproc [options] input output

```
dwifslpreproc dwi_degibbs.mif dwi_preproc.mif -rpe_none -pe_dir ap -  
eddy_options "--slm=linear "
```

dwi_degibbs.mif : input file

dwi_preproc.mif : output file

-rpe_none : Specify that no reversed phase-encoding image data is being provided; eddy will perform eddy current and motion correction only

-pe_dir : phase encode direction, code = ap (Anterior to Posterior)

-eddy_options "--slm=linear " : To model as to how diffusion gradients generate eddy currents, which is Linear in this case(could also be none or quadratic)

Step – 6: Extracting all b₀ volumes

Extracting all volumes corresponding to b=0 (no gradient) from the scan obtained after FSL preprocessing.

Command: dwiextract [options] input_file output_file

```
dwiextract dwi_preproc.mif -bzero b0.mif
```

-bzero : Output b=0 volumes (instead of the diffusion weighted volumes, if -singleshell is not specified).

Step – 7 : Average b0 images into one

Command used: mrmath

Function : This command is used for computing the summarised statistics of image intensities either across multiple images or along a particular direction(axis) for a single image.

Syntax: mrmath [options] input [input ...] operation output

```
mrmath b0.mif -axis 3 mean b0_mean.nii.gz
```

-quiet : to not display information messages or progress status. This option can be omitted as per convenience

Input file: b0_ap.mif (volumes corresponding to b_val = 0 for anterior to posterior phase encoding)

-axis : to specify the direction along which the statistic is to be computed (Z axis in this case, the same axis along which bo_ap.mif and b0_pa.mif were concatenated)

Operation : mean

Output file : b0.nii

Step – 8: Compute transform to align T1 image to DWI

Command: flirt

Function: Performs affine registration, i.e., technique is typically used to correct for geometric distortions and deformations that occur with non-ideal imaging angles.

Syntax: flirt [options] -in <inputvol> -ref <refvol> -omat <outputmatrix>

```
flirt -dof 6 -cost normmi -in RAW/t1mprnssagpat2iso.nii -ref b0 mean -omat T fsl.txt
```

Options:

- -in : input volume (avg T1 image: t1mprnssagpat2iso.nii)
 - -ref : reference volume (b0_mean)
 - -omat: 4x4 matrix which stores the calculated affine transformation that registers the input to the reference (T_fsl.txt)
 - -dof : Degrees of freedom (dof = 6 in this case)
 - -cost : to specify the cost function (can be inter- and intra-modal functions)
 - Normmi : Inter modal cost function, computes Normalized Mutual Information measuring the mutual dependence between two clustering labels

Step – 9: convert the transform matrix to MRtrix compatible format

Command: transformconvert

Function: To convert the transformation matrix provided by FSL's flirt command in the previous step to a format usable in MRtrix.

Syntax: transformconvert [options] input [input ...] operation output

```
transformconvert T_fsl.txt RAW/t1mprnssagpat2iso.nii  
b0_mean.nii.gz flirt_import T_T1toDWI.txt
```

Options :

Step- 10: Apply the transform to the T1 image

Command: mrtransform

Function : If a linear transform is applied, it modifies the image transformation matrix. For a DW scheme having the number of directions matching the number of volumes in the images, any transformation applied using the *-linear* option will be also be applied to the directions.

Syntax: mrtransform [options] input output

```
mrtransform -force -linear T_T1toDWI.txt RAW/t1mprnssagpat2iso.nii  
T1_al.nii.gz
```

Options

- *-linear* : To specify a linear transform to apply, 4x4 ascii file format
- Transformation matrix : T_T1toDWI.txt (obtained from previous step)
- Input file: t1mprnssagpat2iso.nii
- Output file: T1_al.nii.gz (-force causes force overwrite of output files)

Step – 11: Convert the aligned T1 into .mif format

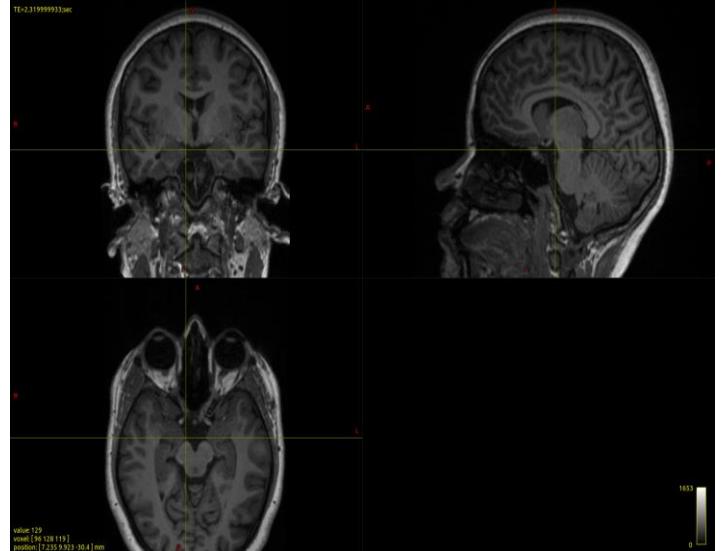
Command:mrconvert

Function: Perform conversion between different file types and optionally extract a subset of the input image

Syntax: mrconvert [options] input output

```
mrconvert -force T1_al.nii.gz T1_al.mif
```

- Input : T1_al.nii.gz (output of previous step)
- Output: T1_al.mif



Step – 12: 5-tissue type segmentation

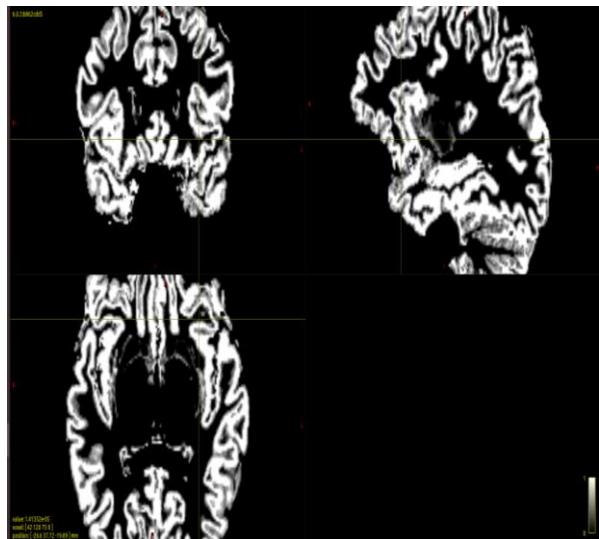
Command: 5ttgen

Function: Generates a five-tissue-type (5TT) segmented tissue image suitable for use in Anatomically-Constrained Tractography (ACT). There are various algorithms available for performing this task.

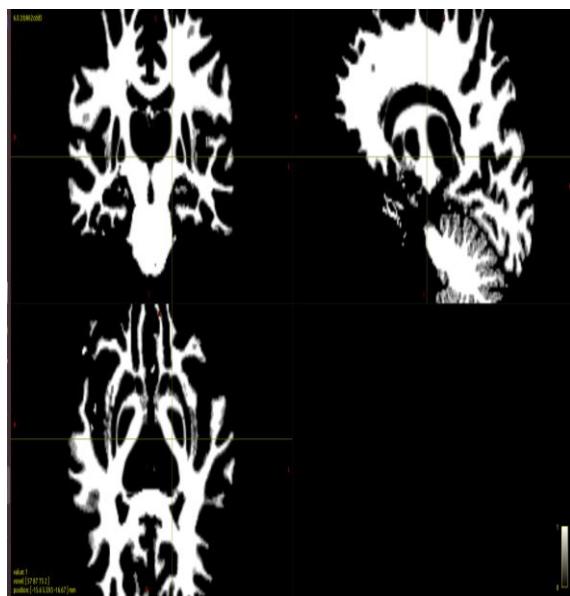
Syntax: 5ttgen [options] algorithm input output

```
5ttgen -force -quiet fsl T1_al.mif 5TT.mif
```

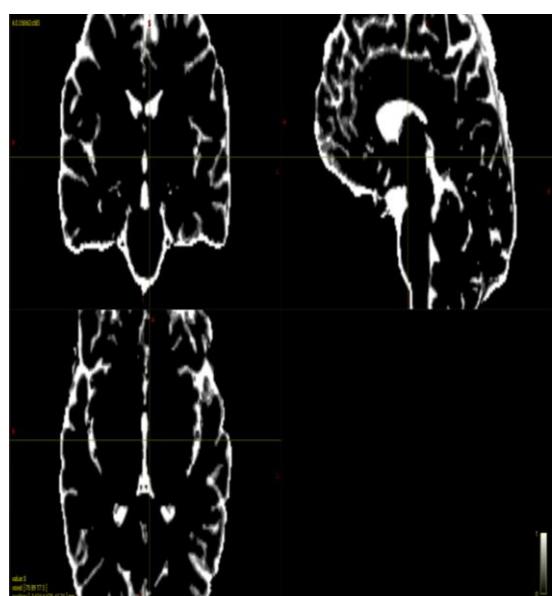
- Algorithm: fsl
- Input: average_t1.mif
- Output: 5TT.mif



Grey Matter
Sub-cortical Grey Matter



White Matter
Cerebrospinal fluid

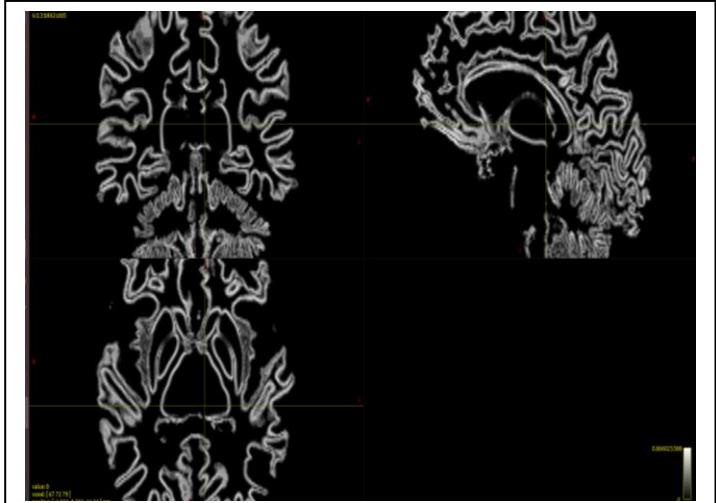


Step – 13 : extract grey-matter-white-matter interface (GMWMI)

- Command: 5tt2gmwmi
- Function: Generate a mask image appropriate for seeding streamlines on the grey matter-white matter interface
- Syntax: 5tt2gmwmi [options] 5tt_in
mask_out

```
5tt2gmwmi -force 5TT.mif
gmwmi.mif
```

- 5tt_in (input - 5TT segmented anatomical image) : 5TT.mif
- Mask_out (output mask image) : gmwmi.mif



Step – 14: compute response function for Constrained Spherical Deconvolution (CSD)

Command: dwi2response

Function: Estimate response function for spherical deconvolution using a specific algorithm

Syntax: dwi2response [options] algorithm input output

```
dwi2response tournier -force dwi_preproc.mif response.txt
```

Algorithm: tournier (This algorithm is a convenient and reliable way to estimate the single-fibre white matter response function to perform (single-tissue) Constrained spherical deconvolution)

Input: dwi_preproc.mif

Output : response.txt

Step – 15: Perform CSD and Estimate Fiber Orientation Distributions (FODs)

Command: dwi2fod

Function: Estimate fibre orientation distributions from diffusion data using response function for spherical deconvolution obtained in the previous step

Syntax: dwi2fod [options] algorithm dwi response odf [response odf ...]

```
dwi2fod -force csd dwi_preproc.mif response.txt WM_FODs.mif
```

Algorithm: csd (This algorithm is designed for single-shell data and only uses a single b-value.)

dwi(input diffusion-weighted image) : dwi_preproc.mif

response(Response function) : response.txt (contains a single row, corresponding to the b-value used for CSD.)

odf : Output image(WM_FODs.mif)

Step – 16: Whole brain tractography using GMWMI as seed

Command: tckgen

Function: Performs streamline tractography using a source(seed) image by attempting to seed from new random locations until the target number of streamlines have been selected or the maximum number of seeds has been exceeded.

Syntax: tckgen [options] source tracks

```
tckgen -force -seed_image gmwmi.mif -act 5TT.mif -backtrack -  
crop_at_gmwmi WM_FODs.mif whole_brain.tck -select 1000000
```

-seed_image: to seed streamlines entirely at random within a mask image (gmwmi.mif)

-act : uses the Anatomically-Constrained Tractography framework during tracking; provided image must be in the 5TT (five-tissue-type) format (5TT.mif)

-backtrack: to allow tracks to be truncated and re-tracked if a poor structural termination is encountered

-crop_at_gmwmi: to crop streamline endpoints more precisely as they cross the GM-WM interface

-select 1000000 : sets desired number of streamlines(1000000) to be selected by tckgen, after all selection criteria have been applied. Seeding of streamlines continues until this number of streamlines have been selected, or the maximum allowed number of seeds has been exceeded

Output: whole_brain.tck

Step-17: Viewing the tractography file

Command: mrview

Syntax: mrview <DWI input file> -tractography.load <.tck file>

Note1: All the commands to be executed in the terminal are highlighted in **RED** color.

Note 2: Instead of running the command sequentially on the terminal, they can be put together in a matlab script and run together. However, there is a slight change in syntax when it comes to running commands in matlab. It is depicted using an example below:

```
% Gibbs artifact removal  
cmd = sprintf('mrdegibbs -force dwi_denoised.mif dwi_degibbs.mif');  
system(cmd);
```

