

École polytechnique de Louvain

Development of an image processing pipeline for large-scale in vivo population studies of the brain microstructure

Authors: **Quentin DESSAIN, Mathieu SIMON**

Supervisor: **Benoit MACQ**

Readers: **Laurence DRICOT, Gaëtan RENSONNET, Michel VERLEY-SEN**

Academic year 2020–2021

Master [120] in Electrical Engineering

Abstract

Diffusion weighted magnetic resonance imaging (DW-MRI) is a rapidly evolving, non radiating and non invasive technique that allows to capture information on the brain microstructure through the restricted diffusion of water molecules. DW-MRI has seen a growing interest in the recent years motivating the acquisition of large multi-scanner multi-site data sets. The substantial acquisition time of this type of MRI sequence has also encouraged the extensive use of Echo Planar Imaging which suffers from additional artifacts and noise. Several tools have been developed in order to correct those individual problems but they come with the disadvantages of processing only one subject at a time and requiring different softwares making them cumbersome to use. This work presents and evaluates the performances of the ElikoPy library, a complete diffusion MRI processing pipeline that reduces common sources of artifact and captures information on the brain microstructure through multiple microstructural diffusion models. ElikoPy has been designed to deal with large databases and to be robust to different types of acquisitions. A particular attention has been paid to the correction of the motion artifacts and the handling of the challenges introduced by the new gradient cycling acquisition scheme.

Acknowledgements

We would like to express our deepest appreciation to our supervisors Pr.Benoît Macq and Pr.Laurence Dricot for the opportunity that was presented to us to discover this fascinating topic and the world of diffusion weighted imaging.

We owe a deep sense of gratitude to PhD.Gaëtan Renzonnet for the time spent answering our numerous questions and for guiding us throughout this Master's thesis.

We wish also to extend our sincere thanks to Nicolas Delinte for his reactivity, his advices and his valuable comments on this work.

Finally, we would like to thank profusely all the people who have supported and encouraged us during our studies.

Contents

Abstract	i
Acknowledgements	ii
List of Figures	v
List of Tables	ix
List of Acronyms	x
Introduction	1
1 Theoretical background	3
1.1 Magnetic Resonance Imaging	3
1.1.1 Magnetic properties	3
1.1.2 Magnetic Resonance	5
1.1.3 Relaxation	6
1.1.4 Spin echo sequence	8
1.1.5 Spatial localization	9
1.2 Diffusion Weighted Magnetic Resonance Imaging	12
1.2.1 Pulse Gradient Spin Echo	12
1.2.2 Diffusion tensor imaging	13
1.2.3 Tractography	15
1.2.4 Echo Planar Imaging	18
1.3 Beyond DTI, models of the brain microstructure	19
1.3.1 NODDI	20
1.3.2 DIAMOND	22
1.3.3 Microstructure Fingerprinting	24
1.4 Artifacts in Magnetic Resonance Imaging	25
2 ElikoPy pipeline	30
2.1 State of the art	30
2.2 The ElikoPy pipeline	33
2.2.1 Introduction	33
2.2.2 Preprocessing	36
2.2.3 Extraction of white matter mask	50
2.2.4 Computed microstructural metrics	54
2.2.5 Common space registration and Statistics	55
2.2.6 Quality control report	58
2.2.7 Research oriented design choices	59
2.2.8 Relevance for clinical data and population studies	59
2.3 Methods and materials	60
2.3.1 Subjects and data	60

2.3.2	Validation methods	65
2.4	Results	68
2.4.1	Visual results	68
2.4.2	Mean squared error and Coefficient of determination	72
2.4.3	Diffusion metric group statistics	76
2.4.4	Ground truth diffusion metric comparisons	84
2.4.5	Temporal Signal-to-Noise Ratio	86
2.4.6	Signal-to-Noise-Artifact Ratio	89
2.4.7	Comparison with ground truth parameters	91
2.4.8	Susceptibility off resonance field estimation	97
2.4.9	Evaluation of the White matter mask	98
2.5	Final discussion and perspectives	101
3	Motion correction with gradient cycling	104
3.1	Gradient cycling	104
3.2	Eddy slice-to-volume correction	107
3.3	Methods	108
3.4	Results	109
3.4.1	Region wise mean FA	109
3.4.2	Visual comparison of b0 volumes	110
3.4.3	Visual comparison of bundles	113
3.5	Final discussion and perspectives	116
	Conclusion	120
	Bibliography	121
	Appendices	129
A	Outputed metrics for each microstructural model	130
A.1	DTI Metrics	130
A.2	DIAMOND Metrics	130
A.3	NODDI Metrics	131
A.4	Microstructure Fingerprinting Metrics	131
B	Parameters used during data processing	132
B.1	PreQual and DESIGNER parameters during simulated data processing . .	132
B.2	Default parameters of ElikoPy's microstructural metrics	133
C	S2V gradient cycling movement correction : Extra information	136
C.1	b0 comparison for subject with almost no S2V motion artifact	137
C.2	Tractography comparison	139
C.3	Tracts definition	144
D	Quality control report of the subject 21TD.	145
E	API of the package ElikoPy	178

List of Figures

1.1	Magnetic moment $\boldsymbol{\mu}$ of a rotating charged particle and influence of the external field	4
1.2	Double precession movement in the fixed frame and in the rotating frame .	6
1.3	T_1 and T_2 relaxation after a 90° -pulse	6
1.4	Relaxation curves for the transverse and longitudinal magnetization	7
1.5	Free induction decay signal	8
1.6	Effect of the external field inhomogeneities on the MR images	8
1.7	Spin echo sequence	9
1.8	T_1 and T_2 contrasts	10
1.9	Spin echo timing sequence with encoding gradients	11
1.10	Pulse gradient spin echo sequence	12
1.11	Effect of diffusion gradients on spins without and with water molecules diffusion	13
1.12	Ellipsoid representation of the diffusion tensor in the isotropic and anisotropic cases	14
1.13	Representation of a white matter fiber trajectory as a space curve, $\mathbf{r}(s)$. .	16
1.14	Deterministic and probabilistic tractography : example	16
1.15	White matter tracts using a DTI-based tractography	17
1.16	Single shot blipped Echo Planar Imaging sequence (Spin echo) and its k-space trajectory	18
1.17	Slice acquisition scheme for 54 slices acquired in an interleaved manner with a multi-band factor of 3.	19
1.18	Models of diffusion for NODDI	20
1.19	DIAMOND spin packets modelization	23
1.20	Representation of the motion artifacts, Eddy current distortions and susceptibility induced distortions in the distorted and undistorted space . . .	28
1.21	Aliasing artifact, Gibbs ringing, chemical shift and Bias field	29
2.1	Schematic description of the state of the art UK-Biobank, DTIPrep, PreQual and DESIGNER pipelines compared to the proposed ElikoPy pipeline . . .	32
2.2	Overview of the main processing steps in the ElikoPy pipeline	34
2.3	Overview of the folder structure used by the ElikoPy toolbox	35
2.4	Original b0 images and binary mask obtained with using median_otsu . .	37
2.5	Original and denoised b0 images using MPPCA	38
2.6	Histogram of eigenvalues after Principal Component Analysis	39
2.7	Gibbs ringing correction	41
2.8	Synthetic b0 susceptibility induced distortion correction	43
2.9	SynB0-DisCo image synthesis training and application of the network . . .	44

2.10	Correction algorithm followed by Eddy of FSL	46
2.11	Illustration of the problems encountered when trying to interpolate the brains with slice-to-volume motion	47
2.12	Reconstruction of the brains with slice-to-volume motion	48
2.13	Eddy FSL interpolation process along the z-direction	49
2.14	N4 iterative algorithm pseudocode	50
2.15	T1 preprocessing pipeline	52
2.16	Intermediate pipeline step for the T1 image processing	52
2.17	Anisotropic Power map compared to T1 image	54
2.18	Main functionalities available with the common space registration and statistics toolset of ElikoPy	56
2.19	Details of the JHU White Matter (WM) Tractography Atlas	57
2.20	Gradient shells represented on a 3D sphere	61
2.21	DW-POSSUM pipeline for simulating diffusion weighted images	63
2.22	Simulation pipeline for the <i>Custom data</i>	64
2.23	Brain extraction of the <i>Custom data</i>	69
2.24	Visual inspection of the volume-to-volume and slice-to-volume motion arti- facts using the <i>Possum</i> and <i>Custom data</i>	69
2.25	Visual results of the ElikoPy library processing for the correction of the slice-to-volume motion	70
2.26	Visual results of the ElikoPy library processing for the correction of the susceptibility off resonance field and Eddy current induced distortions . . .	70
2.27	Comparison of the Gibbs ringing artifacts affecting the raw data and the data processed using the ElikoPy pipeline	71
2.28	Marchenko-Pastur PCA denoising effect on the shells of the <i>Possum data</i> .	71
2.29	Boxplots of the coefficient of determination (R2) metric averaged over the gray matter for each in vivo data before and after preprocessing	73
2.30	Boxplots of the coefficient of determination (R2) metric averaged over the white matter for each in vivo data before and after preprocessing	73
2.31	Boxplots of the mean square error (MSE) metric averaged over the gray matter for each in vivo data before and after preprocessing	74
2.32	Boxplots of the mean square error (MSE) metric averaged over the white matter for each in vivo data before and after preprocessing	74
2.33	Coefficient of determination maps (R2) of the subject 43DYS	75
2.34	Mean squared error maps of the subject 43DYS	75
2.35	Boxplots of the white matter Fractional Anisotropy for the processed and unprocessed data	77
2.36	Boxplots of the white matter Orientation Dispersion Index for the processed and unprocessed data	78
2.37	Boxplots of the white matter kappa value for the processed and unprocessed data	79
2.38	Boxplots of the white matter total fiber volume fraction for the processed and unprocessed data	80
2.39	Voxel wise FWE-corrected 1-(p value) maps of the statistical differences be- tween the processed and unprocessed datasets for the Fractional Anisotropy, Orientation Dispersion Index , kappa and total fiber volume fraction	82

2.40	Fractional Anisotropy, Mean Diffusivity and Axial Diffusivity (DTI) computed for the <i>GT data</i> , the unprocessed image and the data processed using DESIGNER, PreQual and ElikoPy.	84
2.41	Errors on the diffusion metrics : Fractional Anisotropy, Mean Diffusivity and Radial Diffusivity (DTI)	85
2.42	Gray matter changes in tSNR between the processed data and the unprocessed one	86
2.43	White matter changes in tSNR between the processed data and the unprocessed one	87
2.44	Evolution of the temporal SNR with processing for the $b = 1000[s/mm^2]$ shell.	88
2.45	Evolution of the temporal SNR with processing for the $b = 3000[s/mm^2]$ shell.	88
2.46	SNAR at each processing step of the ElikoPy library	89
2.47	SNAR for each volume of the <i>Possum data</i>	90
2.48	SNAR for each volume of the <i>Custom data</i>	90
2.49	Rigid body motion parameters for the volumes $10 \rightarrow 20$ of the <i>Custom data</i>	92
2.50	Rigid body motion parameters for the volumes $71 \rightarrow 81$ of the <i>Custom data</i>	93
2.51	Rigid body motion for the <i>Possum data</i> before and after preprocessing for each volume considering only volume-to-volume motion	94
2.52	Comparison between the ground truth susceptibility off resonance field applied during the simulation and the estimated field computed by the ElikoPy pipeline. The values are given in Hz	95
2.53	Comparison between the ground truth bias field applied during the simulation and the estimated field computed by the ElikoPy pipeline	95
2.54	Eddy current induced distortion parameters in function of the volume index	96
2.55	Comparison between the correction of the susceptibility resonance field applied using two opposed gradient encoding direction and a structural T1 image	98
2.56	Illustration of the different steps performed on the T1 structural image to compute the white matter mask	99
2.57	Comparison of the first b_0 of the preprocessed dmri data, the segmented T1 structural image and the segmented Anisotropic Power map	100
2.58	Comparison of the preprocessed T1 structural image and the Anisotropic Power map	100
3.1	Thermal load evolution over slices for volume-sorted, volume-interleaved and slice-interleaved cases on a clinical Philips 3T-Achieva	105
3.2	Conventional sequential acquisition and acquisition realized with gradient cycling	106
3.3	Boxplot comparison of FA for the slice-to-volume motion correction	110
3.4	V2V motion correction, subject with mild slice-to-volume	111
3.5	S2V motion correction, subject with mild slice-to-volume	111
3.6	V2V motion correction, subject with severe slice-to-volume	112
3.7	S2V motion correction, subject with severe slice-to-volume	112
3.8	Comparison of the effect of movement correction on probabilistic tracking of the posterior corpus callosum on the subject with mild slice-to-volume movement artifact.	113

3.9	Comparison of the effect of movement correction on probabilistic tracking of the posterior corpus callosum on the subject with severe slice-to-volume movement artifact.	114
3.10	Comparison of the effect of movement correction on probabilistic tracking of the right superior longitudinal fasciculus on the subject with mild slice-to-volume movement artifact.	114
3.11	Comparison of the effect of movement correction on probabilistic tracking of the right superior longitudinal fasciculus on the subject with severe slice-to-volume movement artifact.	115
3.12	Comparison of the effect of movement correction on probabilistic tracking of the right corticospinal tract on the subject with mild slice-to-volume movement artifact.	115
3.13	Comparison of the effect of movement correction on probabilistic tracking of the right corticospinal tract on the subject with severe slice-to-volume movement artifact.	116
3.14	Pseudocode of the algorithm used to apply the motion parameters	118
3.15	Structure of the sequential EPI sequence and gradient cycling with superbloc and interleaving EPI sequence	119
C.1	V2V motion correction, subject with low slice-to-volume	137
C.2	S2V motion correction, subject with low slice-to-volume	138
C.3	Comparison of the effect of movement correction on probabilistic tracking of the posterior corpus callosum on the subject with almost no slice-to-volume movement artifact.	139
C.4	Comparison of the effect of movement correction on deterministic tracking of the posterior corpus callosum on the subject with almost no slice-to-volume movement artifact.	139
C.5	Comparison of the effect of movement correction on deterministic tracking of the posterior corpus callosum on the subject with mild slice-to-volume movement artifact.	140
C.6	Comparison of the effect of movement correction on deterministic tracking of the posterior corpus callosum on the subject with severe slice-to-volume movement artifact.	140
C.7	Comparison of the effect of movement correction on deterministic tracking of the right superior longitudinal fasciculus on the subject with mild slice-to-volume movement artifact.	141
C.8	Comparison of the effect of movement correction on probabilistic tracking of the right corticospinal tract on the subject with almost no slice-to-volume movement artifact.	142
C.9	Comparison of the effect of movement correction on deterministic tracking of the right corticospinal tract on the subject with almost no slice-to-volume movement artifact.	142
C.10	Comparison of the effect of movement correction on deterministic tracking of the right corticospinal tract on the subject with mild slice-to-volume movement artifact.	143
C.11	Comparison of the effect of movement correction on deterministic tracking of the right corticospinal tract on the subject with severe slice-to-volume movement artifact.	143

List of Tables

2.1	Summary of the processing methods applied during the preprocessing and their optional parameters.	40
2.2	Welch's t-test comparison results between the processed and unprocessed data for the R2 and MSE in the gray and white matter.	72
2.3	Welch's t-test comparison results between the processed and unprocessed data for the FA, ODI, kappa and fvf_tot in the regions of the JHU DTI-based probabilist white matter atlas.	81
2.4	Welch's t-test comparison results between the processed and unprocessed data in the gray and white matter for each shell.	87
2.5	Typical runtimes of ElikoPy on the in vivo dyslexia dataset	102
3.1	Diffusion gradient acquisition scheme of the dyslexia dataset	108

List of Acronyms

A Atomic mass

AD Axial Diffusivity

ADC Analog-Digital-Converter

AICU Unbiased Akaike Information Criterion

ANTs Advanced Normalization Tools

AP Anisotropic Power

BET2 Brain Extraction Tool 2

BIDS Brain Imaging Data Structure

BW Bandwidth

CC Corpus Callosum

CMC Conditional Monte Carlo

CSF Cerebrospinal Fluid

CT Computed Tomography

DCT Discrete Cosine Transform

DESIGNER Diffusion parameter EStimation with Gibbs and Noise Removal

DIAMOND Distribution of 3D Anisotropic Microstructural Environments in Diffusion
Compartment Imaging

DICOM Digital Imaging and Communications in Medicine

DiPy Diffusion Imaging in Python

Dmipy Diffusion Microstructure Imaging in Python

dMRI diffusion MRI

DTI Diffusion Tensor Imaging

DW-MRI Diffusion-Weighted Magnetic Resonance Imaging

DWI Diffusion-Weighted Imaging

EC	Eddy Current
EM	Expectation-Maximization
EPI	Echo Planar Imaging
FA	Fractional Anisotropy
FAST	FMRIB's Automated Segmentation Tool
FID	Free Induction Decay
FLIRT	FMRIB's Linear Image Registration Tool
FOV	Field of View
FSL	FMRIB Software Library
FWE	Family-Wise Error
GAN	Generative Adversarial Network
GE	Generalization Error
GM	Gray Matter
GP	Gaussian Process
HARDI	High Angular Resolution Diffusion Imaging
HMMRF	Hidden Markov Random Fields
icvf	intra-cellular volume fraction
MAP	Maximum a Posteriori
mb	multi-band factor
MD	Mean Diffusivity
MF	Microstructure Fingerprinting
MPPCA	Marchenko-Pastur Principal Component Analysis
MR	Magnetic Resonance
MRF	Markov Random Fields
MRI	Magnetic Resonance Imaging
MSE	Mean Squared Error
NIFTI	Neuroimaging Informatics Technology Initiative
NODDI	Neurite Orientation Dispersion and Density Imaging
ODI	Orientation Dispersion Index

PCA	Principal Component Analysis
PE	Phase Encoding
PET	Positron Emission Tomography
PGSE	Pulse Gradient Spin Echo
R²	Coefficient of determination maps
RD	Radial Diffusivity
RF-pulse	Radiofrequency pulse
ROI	Region Of Interest
S2V	Slice to Volume
SE	Spin Echo
SE-EPI	Spin Echo - Echo Planar Imaging
SLF	Superior Longitudinal Fasciculus
SNAR	Signal-to-Noise-Artifact Ratio
SNR	Signal-to-Noise Ratio
Synb0-DisCo	Synthesized b0 for diffusion Distortion Correction
TBSS	Tract-Based Spatial Statistics
TE	Echo Time.
TR	Repetition Time
tSNR	temporal Signal-to-Noise Ratio
V2V	Volume to Volume
VBM	Voxel-based Morphometry
WM	White Matter
Z	Atomic number

Introduction

Diffusion-weighted magnetic resonance imaging (DW-MRI) is a rapidly evolving, non radiating and non invasive technique that allows to capture information on the brain microstructure through the restricted diffusion of water molecules. This technique provides the possibility to improve the characterization of the brain and to make better and earlier diagnoses of brain disorders. This has led to an increasing interest in the structural organization of the human brain and has encouraged the acquisition of large datasets. However, due to its substantial acquisition time, DW-MRI is particularly affected by motion which imposes the use of Echo Planar Imaging forcing to deal with the additional non idealities it introduces. Overall, DW-MRI is often subject to low signal-to-noise ratios and artifacts. Several tools have been developed in order to correct those individual problems but they come with the disadvantages of processing only one subject at a time and require different software making them cumbersome to use.

This Master’s thesis aims to develop and validate a complete processing pipeline in order to deal with large databases by embedding the different processing steps. This provides a standardized framework to ensure reproducibility and consistency, reduce error propagation and improve sensitivity. It also grants the possibility for clinicians to perform fast preprocessing for a large variety of studies with minimal knowledge.

Several pipelines have already been proposed in previous works (Oguz et al., 2014) (Cai et al., 2021) (Maximov et al., 2019) (Ades-Aron et al., 2018). This Master’s thesis expands their framework by :

- Providing complete quality assessment and quality reports for each processed subject.
- Validating the pipeline by a combination of visual inspections, estimated quality metrics using in vivo data (due to the lack of groundtruth) and evaluated quality metrics using “groundtruth” simulated diffusion-weighted images generated based on the framework introduced in (Graham et al., 2016).
- Being compatible with the recent standardized BIDS format.
- Being directly usable on cluster environments with parallelized operations and easy to install through containerization.
- Providing a T1-image registration step as well as an automatic white matter mask extraction.
- Being flexible to different types of acquisition.
- Providing a statistical toolbox enabling inter-group voxel wise and region wise comparison for any microstrutural metric.

Classical pipelines also limit themselves to diffusion tensor imaging (DTI) (Le Bihan et al., 2001). However, DTI does not account for crossing fascicles of axons and is

difficult to interpret. Therefore, this work uses other mathematical models in addition to DTI to infer microstructural features of brain tissues based on diffusion data, namely Neurite Orientation Dispersion and Density Imaging (NODDI) (Zhang et al., 2012), DIstribution of 3D Anisotropic MicrOstructural eNvironments in Diffusion-compartment imaging (DIAMOND) (Scherrer et al., 2016) and Microstructure Fingerprinting (MF) (Rensonnet et al., 2019). DIAMOND and MF are recently developed models which have seen little use on large databases, further highlighting the interest of including them in the pipeline.

Motivated by the lack of solutions for motion correction in the case of the new gradient cycling acquisition scheme, a novel motion correction framework is also introduced. This method allows for more flexible slice realignments accommodating any slice acquisition order. The challenges of gradient cycling for the motion artifacts are well known and are still unsolved by the major suppliers of MRI machines and existing pipelines. ElikoPy tries to fill that void by suggesting possible solutions.

Chapter 1

Theoretical background

This chapter covers the main theoretical knowledge required to understand the terminology and the methods used throughout this report. The subjects discussed are the principle of magnetic resonance imaging, diffusion magnetic resonance imaging, the mathematical models used to infer the microstructural properties from Diffusion-Weighted Magnetic Resonance Imaging (DW-MRI) and the artifacts affecting Magnetic Resonance Imaging (MRI) images.

1.1 Magnetic Resonance Imaging

Magnetic resonance imaging is a medical imaging technique that allows to generate images of organs in the body and their physiological activities (structural and functional imaging). Compared to Computed Tomography (CT) or Positron Emission Tomography (PET) scans, MRI comes with the advantage that it does not involve X-rays or the use of ionizing radiations but relies on the use of strong magnetic fields and radio waves which are considered safer. MRI provides a spatial resolution in the order of the millimeter and is shown to be the investigative tool of choice for brain imaging.

1.1.1 Magnetic properties

MRI relies upon the magnetic properties of the nuclei. Every nucleus is characterized by a form of intrinsic angular momentum called the spin. The spin quantum number I can take several values depending on the Atomic mass (A) and the Atomic number (Z) of the nucleus.

- A even and Z even ($^{12}C, ^{16}O$) : $I = 0$
- A odd and Z odd ($^1H, ^{13}C, ^{31}P$) : $I = 1/2$
- A even and Z odd ($^2H, ^{14}N$) : $I = 1$
- A odd and Z even ($^{23}Na, ^{39}K$) : $I = 3/2$

Particles with $I \neq 0$ are called "spins". One of particular interest is the hydrogen (1H). Since it represents 2/3 of the human body nucleus and has a large intrinsic magnetic moment, it will largely dominate the magnetic properties of the organs of interest in MRI (Kastler et al., 2011). The discussion will therefore focus on half spin particles.

In a classical picture, a spin can be visualized as a particle rotating around its axis. A spinning charged particle creates an electric current and therefore induces a magnetic field

characterized by a magnetic moment $\boldsymbol{\mu}$. As a result, a spin can be viewed as a small bar magnet (see Fig.1.1).

Due to random thermal motion, when no external magnetic field is applied, the magnetic moments of the nuclei are randomly oriented resulting in a zero macroscopic magnetization.

$$\mathbf{M} = \frac{1}{V} \sum_i \boldsymbol{\mu}_i = 0, \quad (1.1)$$

where V is the considered volume and $\boldsymbol{\mu}_i$ are the magnetic moment vectors of the individual nuclei in V .

In the presence of an external magnetic field $\mathbf{B}_0 = B_0 \mathbf{1}_z$, the magnetic moments will try to align with the magnetic field. This alignment is not perfect and according to quantum mechanics, only a discrete set of $2I + 1$ orientations are possible for a spin in a magnetic field. For the $1/2$ spins of interest, it results in two possible orientations : parallel (spin up) or antiparallel (spin down).

Moreover, a magnetic moment in a magnetic field experiences a torque defined by :

$$\boldsymbol{\tau} = \boldsymbol{\mu} \times \mathbf{B}. \quad (1.2)$$

As a result, a precession movement is created along the direction of the external magnetic field with a frequency given by :

$$f_0 = \gamma B_0, \quad (1.3)$$

f_0 is called the Larmor frequency (in Hz) and γ is the gyromagnetic ratio (in Hz/T) which is characteristic for the nucleus (42.58 MHz/T for ^1H). This precession movement is shown in Fig.1.1

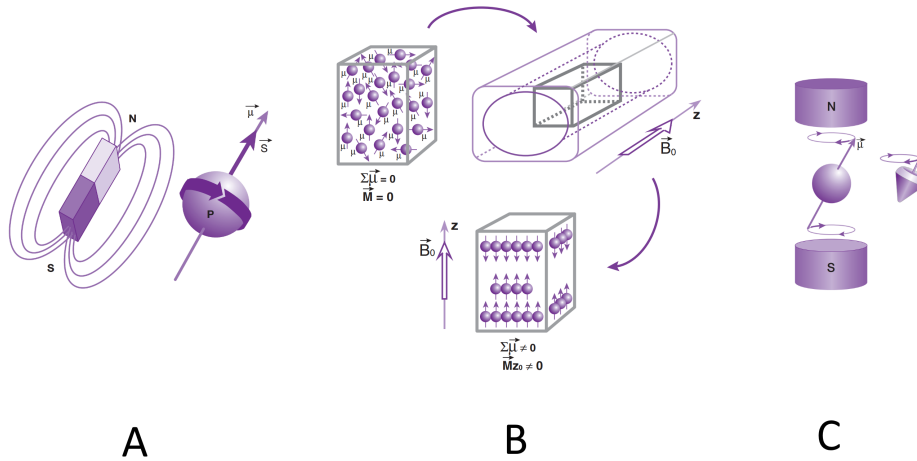


Figure 1.1: (A) Magnetic moment $\boldsymbol{\mu}$ of a rotating charged particle (B) Influence of external fields on the magnetic moment orientation (C) Precession movement of a spin in an external magnetic field, from (Kastler et al., 2011).

Overall, once an external magnetic field is applied, a macroscopic magnetization appears. This magnetization can be decomposed in a transverse and longitudinal magnetization :

$$\mathbf{M} = \frac{1}{V} \sum_i \boldsymbol{\mu}_i = M_z \mathbf{1}_z. \quad (1.4)$$

The transverse magnetization M_{xy} is zero due to the spins having a random phase for their precession movement resulting in the magnetic moments adding non coherently.

However, the longitudinal magnetization M_z is non zero. Indeed, the two possible orientations (parallel or antiparallel) correspond to two different energy levels. The difference in energy of the two states is given by :

$$\Delta E = E_{\downarrow} - E_{\uparrow} = hf_0, \quad (1.5)$$

where $h = 6.62607004 \times 10^{-34} \text{ m}^2 \text{ kg/s}$ is the Planck's constant.

The parallel orientation is at a lower energy level and is therefore more populated. More spins are oriented up than down. The ratio of the occupations between the two energy states is given by :

$$\frac{N_{\uparrow}}{N_{\downarrow}} = \exp\left(\frac{\Delta E}{kT}\right). \quad (1.6)$$

A consequence is that the resulting macroscopic longitudinal magnetization is non zero with the value :

$$M_z = M_0 = \frac{\gamma^2 h^2 \rho B_0 I(I+1)}{3kT}, \quad (1.7)$$

with ρ the spin density (Kastler et al., 2011).

1.1.2 Magnetic Resonance

For a spin in equilibrium, the magnetization $\mathbf{M} = M_0 \mathbf{1}_z$ points toward the external field $\mathbf{B}_0$. Unfortunately, the biological magnetization is small in comparison making the signal hardly measurable. In order to obtain a measurable signal, the magnetization has to be moved away from the direction of the external magnetic field. This is achieved through the use of a time varying Radiofrequency pulse (RF-pulse) that oscillates at the Larmor frequency :

$$\mathbf{B}_1(t) = 2B_1(t) \cos(2\pi f_{RF}t + \phi) \mathbf{1}_{xy}, \quad (1.8)$$

where $f_{RF} = f_0$ is the frequency of the RF-pulse (Kastler et al., 2011).

Due to the RF-pulse having the Larmor frequency, it carries an energy equal to ΔE and a phenomenon of resonance appears. The RF-pulse induces transitions from the energy level $E_{\uparrow}$ to $E_{\downarrow}$ causing the longitudinal magnetization to switch its orientation.

Moreover, the transition from one state to the other causes the spins to get in phase resulting in a transverse magnetization.

The effect of the RF-pulse on the macroscopic magnetization can be observed in the fixed frame or in the rotating frame. (see Fig.1.2)

The macroscopic magnetization vector undergoes a double precession movement (around $\mathbf{B}_0$ and $\mathbf{B}_1$). If considered in a frame rotating at the Larmor frequency, the movement becomes a simple tipping of $\mathbf{M}$. The angle between the z-direction and the magnetization is called the “flip-angle” and is defined by :

$$\alpha = \gamma \int_{T_{RF}} B_1(t) dt \stackrel{B_1(t) \text{ cst}}{=} \gamma B_1 T_{RF}, \quad (1.9)$$

with T_{RF} the duration of the RF-pulse. Two important RF-pulses to consider here are the 90° and 180° pulses.

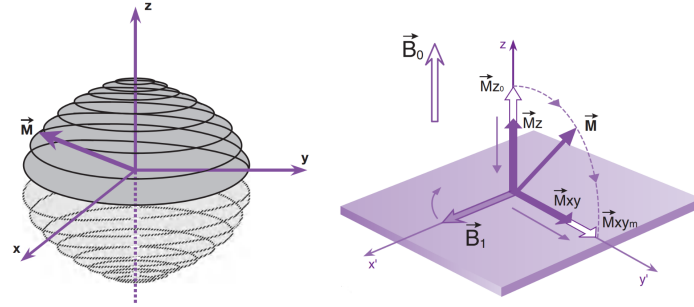


Figure 1.2: Double precession movement in the fixed frame (left) and in the rotating frame (right), from (Kastler et al., 2011).

The macroscopic magnetization is now given by :

$$\mathbf{M} = \frac{1}{V} \sum_i \boldsymbol{\mu}_i = M_z \mathbf{1}_z + M_{xy} \mathbf{1}_{xy}. \quad (1.10)$$

1.1.3 Relaxation

After a spin system is excited, it tries to go back to its equilibrium state (i.e. $\mathbf{M}_0$) meaning that the longitudinal magnetization M_z will evolve back to M_0 while the transverse magnetization M_{xy} will disappear. These are two different phenomenons. The first one is called T1 or spin-lattice relaxation and the second one is called T2 or spin-spin relaxation (see Fig.1.3).

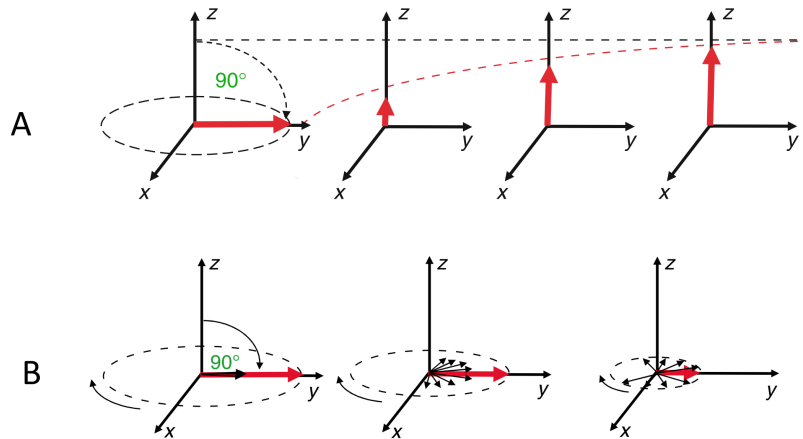


Figure 1.3: (A) T_1 relaxation after a 90°-pulse (B) T_2 relaxation after a 90°-pulse, from (Plein et al., 2011).

T1-relaxation : After the RF-pulse, through energy exchanges between the spins and the surrounding 'lattice' (thermic exchange, i.e. collisions), the spins go back to their original energy level preceding the RF-pulse excitation. As a result, the longitudinal magnetization M_z goes back to its equilibrium M_0 . This is described by the following mathematical expression :

$$M_z(t) = M_0 + [M_z(0) - M_0]e^{-\frac{t}{T_1}}. \quad (1.11)$$

$M_z(0)$ is the longitudinal magnetization after the RF-pulse and T_1 is the time constant that describes the time needed to recover 63% of the equilibrium magnetization. This is represented in Fig.1.4.

The efficiency of the energy transfer depends on the frequency of the molecular collisions f_c . If this frequency is close to the Larmor frequency, the recovery rate increases and the time constant T_1 is smaller. As the frequency f_c differs from one tissue to the other, different responses will be observed for each tissue.

T2-relaxation : After the RF-pulse, a phase shift occurs on the spins. Since they do not stay in phase, it results in the disappearance of the transverse magnetization M_{xy} . This phenomenon is the consequence of the interaction of the spins with each other and is thus called spin-spin relaxation. Protons progress in different molecular environments where small magnetic field superpose with B_0 . Due to these inhomogeneities, the spins precess at different speeds and lose their phase coherence. This is described by the following mathematical expression :

$$M_{xy}(t) = M_{xy}(0)e^{-\frac{t}{T_2}}. \quad (1.12)$$

$M_{xy}(0)$ is the transverse magnetization after the excitation and T_2 is the time constant that describes the time needed for M_{xy} to lose 63% of its initial value. This is represented in Fig.1.4.

The T_2 time is also characteristic of the tissue and different responses are obtained depending on the environment.

In practice, the transverse magnetization decays faster than what is expressed previously. This is mainly due to inhomogeneities in the external magnetic field B_0 . This can be accounted for by introducing an effective relaxation time T_2^* .

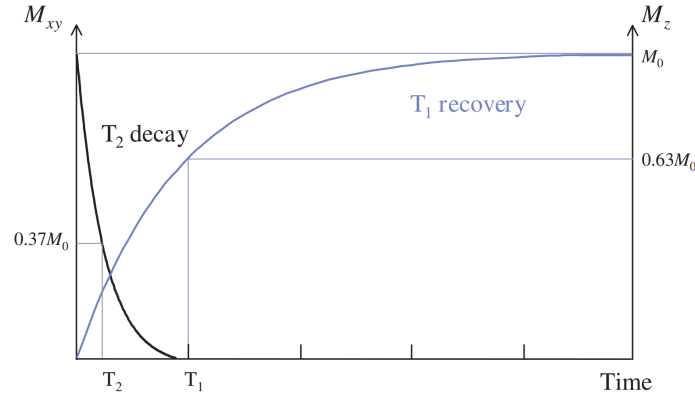


Figure 1.4: Relaxation curves for the transverse and longitudinal magnetization, from (McRobbie et al., 2004).

Supposing a 90° -pulse is applied, the transverse magnetization will go back from M_0 to zero after the excitation. In the fixed frame, this can be written :

$$M_{xy}(t) = M_0 e^{(-\frac{t}{T_2})} e^{i(-2\pi f_0 t + \phi)}, \quad (1.13)$$

where a complex notation $M_{xy} = M_x + iM_y$ has been introduced.

By using a coil situated in the transverse plane, following the Faraday's law, the oscillating behavior of the transverse magnetization induces a signal known as the Free Induction

Decay (FID). This signal is sinusoidal with an envelope decreasing exponentially according to the T_2^* constant. (see Fig.1.5)

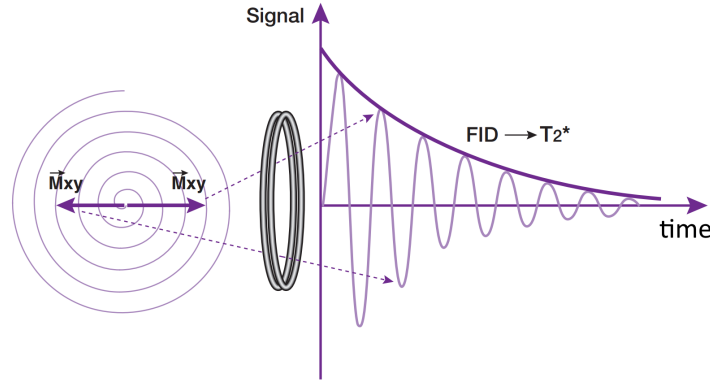


Figure 1.5: FID signal, from (Kastler et al., 2011).

The overall equation of motion of the magnetization $\mathbf{M}$ in the rotating frame is given by the Bloch equations :

$$\frac{d\mathbf{M}}{dt} = -\frac{\mathbf{M}_{xy}}{T_2} + \frac{M_0 - M_z}{T_1}\mathbf{1}_z + \gamma \times [\mathbf{B}_1 + (B_0 - \frac{f}{\gamma})\mathbf{1}_z]. \quad (1.14)$$

1.1.4 Spin echo sequence

As explained above, the field inhomogeneities of $\mathbf{B}_0$ needs to be taken into account resulting in a signal that decays with an exponential of time constant $T_2^* < T_2$. However, what is interesting to measure is T_2 which is linked to the interaction of the spins with each other (and thus the tissue) and not with the external field. As the signal decays faster, it is also more difficult to measure and the difference between the tissues are harder to observe. The effect of a T_2^* -weighting on MRI can be observed in Fig.1.6

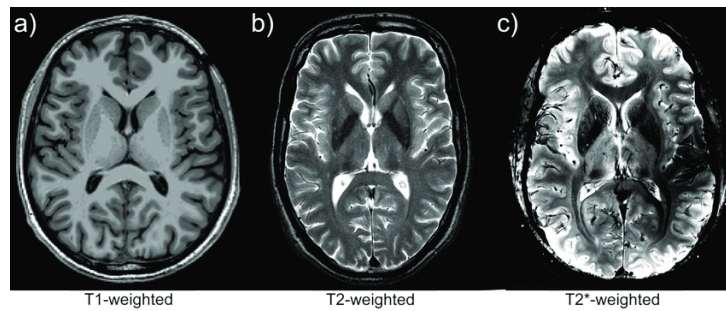


Figure 1.6: Effect of the external field inhomogeneities on the MR images, from (Plein et al., 2011).

To solve this issue, a Spin Echo (SE) method is used. The spin echo is a pulse sequence that consists of a 90° -pulse followed by a 180° -pulse. If τ denotes the time between the two, thanks to the 180° -pulse after a time 2τ the spins are refocused (rephased), only the intrinsic spin-spin dephasing remains and a SE is formed. This is illustrated in Fig.1.7

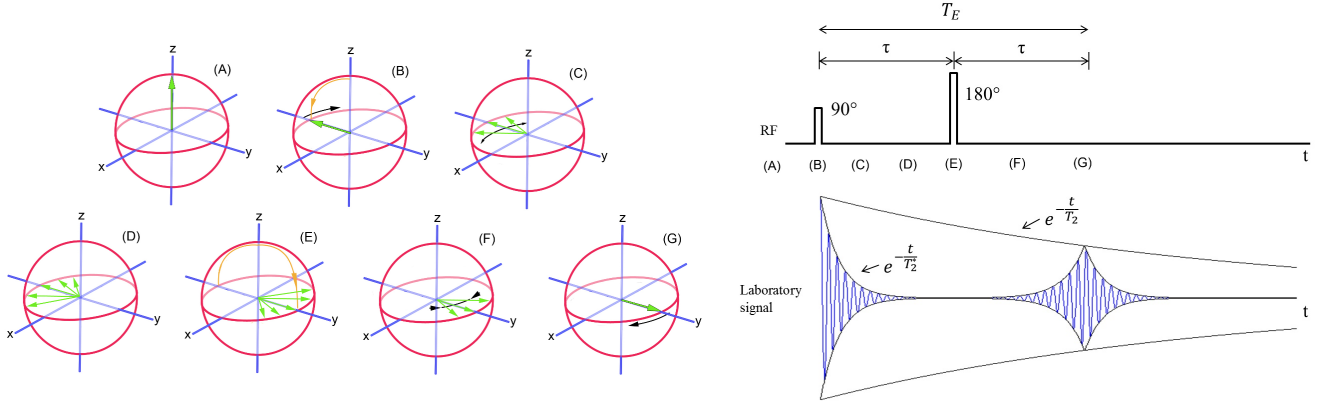


Figure 1.7: Spin echo sequence (A) Spins are at the equilibrium and aligned with the external magnetic field $\mathbf{B}_0$ (B) The longitudinal magnetization is converted into a transverse magnetization due to the 90° -pulse (C)(D) Due to the field inhomogeneities the spins rotate at different speeds and lose their coherence (E) A 180° -pulse is applied and inverse the phase (F) The spins still rotate at different speeds are progressively refocused (G) Spin echo at time $t = 2\tau$, adapted from (Filler, 2009).

In practice this sequence is repeated giving two main parameters to the SE sequence :

- Echo time (TE) : time between the 90° -pulse and the first echo ($TE = 2\tau$).
- Repetition time (TR) : time between the 90° -pulse of two adjacent spin echo sequences.

Due to relaxation the imaged spin density $\rho(x, y)$ is therefore weighted by a contrast factor c . In the case $TE \ll TR$, this factor is given by :

$$c = M_0[1 - e^{-\frac{TR}{T_1}}]e^{-\frac{TE}{T_2}}. \quad (1.15)$$

By varying the Echo Time. (TE) and the Repetition Time (TR), it is therefore possible to obtain different image contrasts :

- TE and TR short : T_1 -contrast, image weighted by the T_1 relaxation.
- TE and TR long : T_2 -contrast, image weighted by the T_2 relaxation.
- TE short and TR long : Proton density contrast, image weighted by the density of spins.

The T_1 and T_2 contrasts are illustrated in Fig.1.8. This master thesis mainly uses T_1 images as it provides a good contrast for segmentation of the white and gray matter.

1.1.5 Spatial localization

In MRI, one has to excite only a well-defined region of the patient and to localize in space the MR-signals. This is achieved using magnetic field gradients generated through gradient coils.

A gradient field is a magnetic field that varies linearly along a spatial direction. If gradients are applied along all three spatial directions in the presence of a static field $\mathbf{B}_0$, the total field becomes :

$$\mathbf{B} = \mathbf{B}_0 + (\mathbf{G} \cdot \mathbf{r})\mathbf{1}_z = \mathbf{B}_0 + (G_x x + G_y y + G_z z)\mathbf{1}_z. \quad (1.16)$$

As a consequence, the Larmor frequency $f_0 = \gamma B$ becomes position dependent and varies linearly with the spatial location.

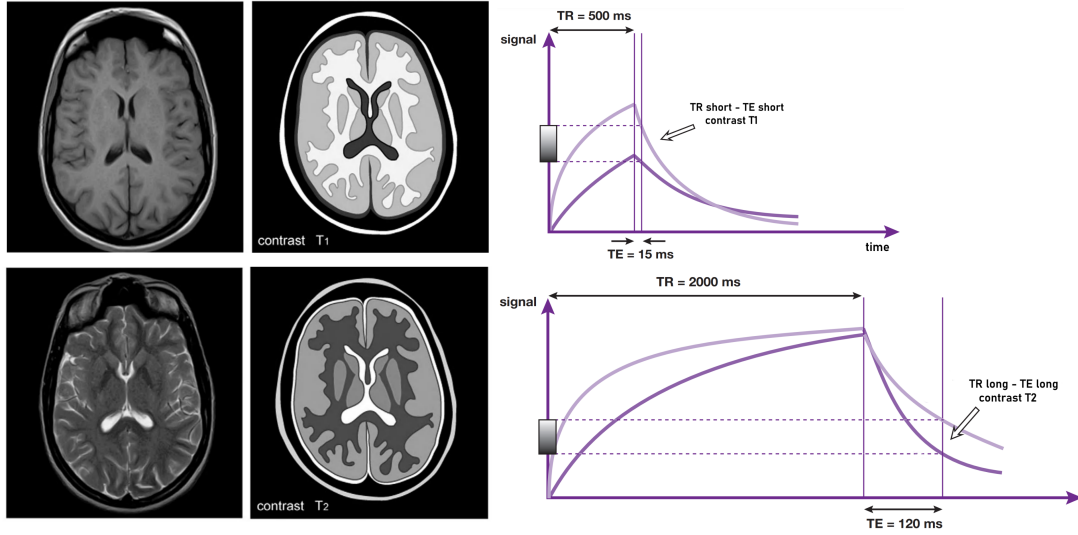


Figure 1.8: T_1 and T_2 contrasts, adapted from (Kastler et al., 2011).

Slice selection gradient (G_z) : This magnetic gradient varies linearly in the z -direction. As a result, the Larmor frequency is given by:

$$f(z) = \gamma[B_0 + G_z z] \quad (1.17)$$

It is therefore possible to excite only the spins within a slice of thickness Δz located at z_0 using a RF-pulse of bandwidth $\Delta f = \gamma G_z \Delta z$ and center frequency $f_c = f_0 + \gamma G_z z_0$:

$$B_1(t) = A \operatorname{sinc}(\pi \Delta f t) e^{-i2\pi f_c t}. \quad (1.18)$$

In practice, sinc-pulses of infinite duration cannot be applied and a consequence of their truncation is the introduction of a phase shift. To cope with this issue a refocusing gradient is used (see Fig.1.9).

Phase encoding gradient (G_y) : In order to encode the spatial information in the y -direction, after the 90° -pulse, a gradient G_y is turned on along the y -direction for a time T_y . The transverse magnetization thus acquires a phase shift :

$$\phi(y) = -\gamma G_y y T_y. \quad (1.19)$$

The direction along which the phase encoding gradient is applied is more generally called the PE-direction. It is usually the $+y$ or $-y$ direction (i.e. posterior-to-anterior PA-direction or anterior-to-posterior AP-direction).

Frequency encoding gradient (G_x) : In order to encode the spatial information in the x -direction, during the acquisition, a gradient G_x is turned on along the x -direction. The transverse magnetization thus acquires a phase shift with time (integral of frequency) :

$$\phi(x) = -\gamma G_x x t. \quad (1.20)$$

In order to avoid a net phase shift at the center of the echo, a focusing gradient with half duration is applied before the 180° -pulse to compensate the phase shift (see Fig.1.9). The direction along which the frequency encoding gradient is applied is more generally called the FE-direction. It is usually the $+x$ or $-x$ direction.

Overall, thanks to the frequency and phase encoding, the spins are uniquely defined with a phase (y-direction) and a frequency (x-direction) and can be located.

The Magnetic Resonance (MR) signal is given by :

$$S(k_x, k_y) = \int \int \rho(x, y) e^{-i2\pi(k_x x + k_y y)}, \quad (1.21)$$

where k_x and k_y are the indexes of a so called k-space, defined for a SE sequence as :

$$\begin{cases} k_x = \gamma G_x t \\ k_y = \gamma G_y T_y. \end{cases} \quad (1.22)$$

These expressions clearly show the relationship with the 2D Fourier transform. Each experiment with a specific phase encoding gradient fills one line of the k-space $S(k_x, k_y)$. The experiment is then repeated multiple times with different values of phase encoding gradient before reconstructing the spatial image $\rho(x, y)$ using an inverse Fourier transform.

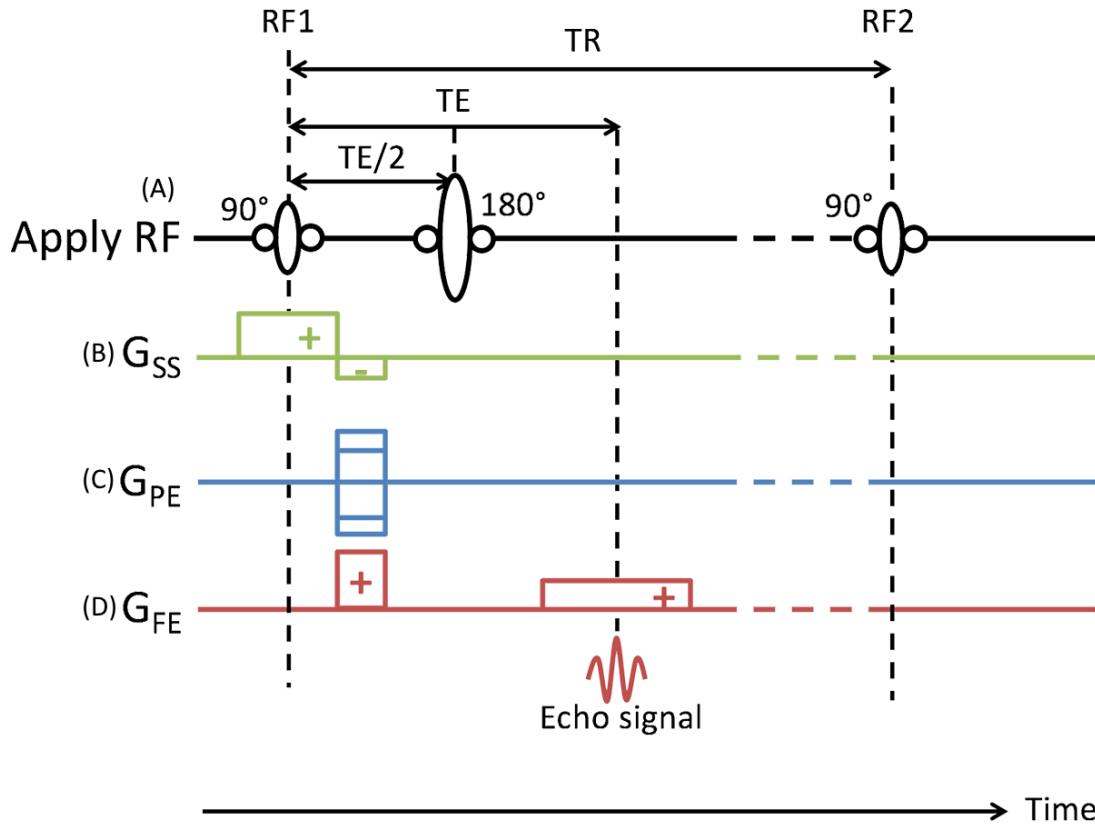


Figure 1.9: Spin echo timing sequence (A) RF-pulses (B) slice selection gradient G_z (+) and its refocusing gradient (-) (C) phase encoding gradient G_y (D) frequency encoding gradient G_x .

In Fig.1.9, the echo signal is represented. When it appears, a readout gradient is switched on and the Analog-Digital-Converter (ADC) samples this signal. The time between each sample of the ADC is called the dwell time Δt . The total sampling time is $T_s = N\Delta t$, with N the number of phase encoding steps. The Bandwidth (BW) in the phase encoding direction is the reciprocal of the total sampling time and has the unit of Hz/pixel. The bandwidth basically describes which frequency range from the analyzed echo signal is transferred into one pixel along the phase encoding direction. The readout BW, also called

receiver BW, is the reciprocal of the dwell time, it is analog to the Phase Encoding (PE) BW but in the readout direction (i.e. frequency encoding direction). For a spin echo sequence, a typical value for the bandwidth in the phase encoding direction is of the order of $200\text{Hz}/Px$ while the readout bandwidth is in the $k\text{Hz}$ range.

1.2 Diffusion Weighted Magnetic Resonance Imaging

Diffusion-weighted magnetic resonance imaging (DW-MRI) is an MRI technique that allows to use the diffusion process of molecules, mainly water, to generate contrast in a non invasive manner. Water molecules experience a random thermal movement called Brownian motion which is not free but is restricted by the environment such as macromolecules, fibers, and membranes. Water molecule diffusion can therefore reveal microstructural changes in tissues, making it a powerful tool for early detections of pathological tissues, with a higher specificity than conventional MRI sequences as it enables to investigate the influence of brain connections (Huisman, 2010).

1.2.1 Pulse Gradient Spin Echo

In order to weight the images with the diffusion of water molecules, the most commonly used sequence is the Pulse Gradient Spin Echo (PGSE). This modified spin echo sequence is composed of two supplementary diffusion gradients (see Fig.1.10). The first gradient is applied after the 90° -pulse and is used to dephase the magnetization while the second one is applied after the 180° -pulse and allows to rephase the magnetization.

The PGSE sequence has several parameters :

- Δ : time between the two diffusion gradients.
- δ : duration of the diffusion gradients.
- G : magnitude of the diffusion gradients.

These parameters are summarized under a single scalar that characterizes the PGSE sequence, the b-value (Huisman, 2010) :

$$b_{PGSE} = (\gamma G \delta)^2 (\Delta - \frac{\delta}{3}) [s/mm^2]. \quad (1.23)$$

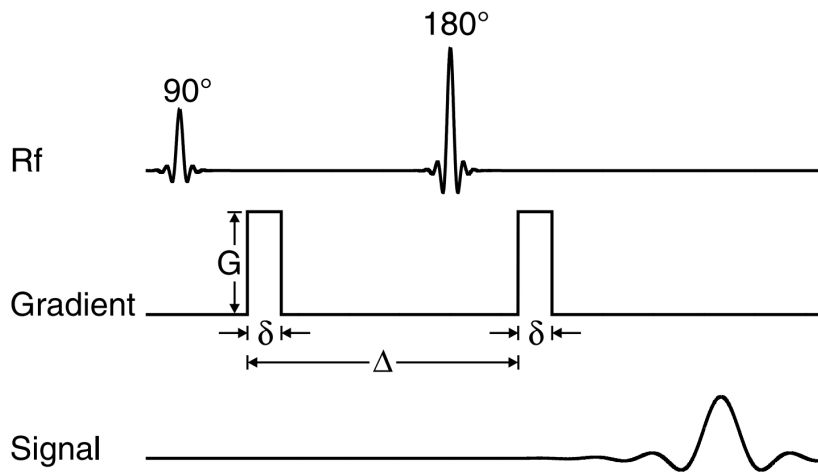


Figure 1.10: Pulse gradient spin echo sequence (G_x , G_y and G_z are omitted on the figure), from (Johansen-Berg and Behrens, 2013).

In Fig.1.11 is represented the effect of the diffusion gradients on the spins. The first diffusion gradient dephases the spins proportionally to their position while the second gradient will rephase them accordingly. However, if the spins have moved due to diffusion, they will not be correctly rephased and the signal will be attenuated (i.e. appearing darker on the diffusion weighted image).

The DW images are weighted by the diffusion of water molecules along the specific direction of the diffusion gradients. Therefore, the imaging process is usually repeated multiple times with different directions to deal with anisotropic diffusion. While in the Cerebrospinal Fluid (CSF) the diffusion is globally isotropic and the signal is the same for any orientation of the diffusion gradients, in axons, the diffusion becomes highly anisotropic. The diffusivity of a water molecule in an axon is higher along the direction of the axon compared to a direction perpendicular to it.

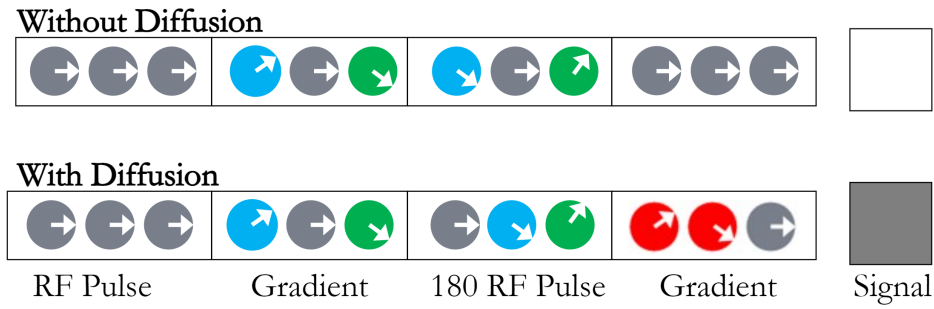


Figure 1.11: Effect of diffusion gradients on spins without and with water molecules diffusion, from (ME Mercredi, 2016).

1.2.2 Diffusion tensor imaging

Once the DW-MRI data is acquired, models are needed to retrieve the information about tissue microstructure embedded in the measured diffusion signal. The most commonly used model is called Diffusion Tensor Imaging (DTI). The idea of DTI is to model the diffusion of water molecules in each voxel with a 3×3 diffusion tensor $\mathbf{D}$ (symmetric positive semi-definite) allowing to analyze the three-dimensional diffusion of water molecules.

$$\mathbf{D} = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{bmatrix}. \quad (1.24)$$

The diagonal elements of this matrix correspond to diffusivities along the three orthogonal axes of the scanner, while the off-diagonal elements correspond to the correlation between displacements along those orthogonal axes (Johansen-Berg and Behrens, 2013).

This matrix can be diagonalized :

$$\mathbf{D} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}, \text{ with } \mathbf{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_2 \end{bmatrix}. \quad (1.25)$$

The diffusion tensor is often thought in terms of an ellipsoid, a surface representing the distance that a molecule will diffuse to with equal probability from the origin. The diffusion tensor formalism provides an “internal reference frame”, namely, the eigensystem. The

principal axes of the ellipsoid are given by the eigenvectors, and the lengths are given by the diffusion distance in a given time, t (proportional to the square root of the diffusivity) (Johansen-Berg and Behrens, 2013). This is represented in Fig.1.12.

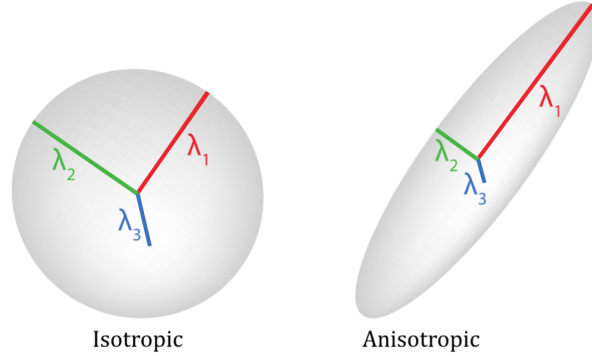


Figure 1.12: Ellipsoid representation of the diffusion tensor in the isotropic and anisotropic cases.

The link between the diffusion tensor and the signal attenuation, due to the water molecules diffusion, in a PGSE sequence of b-value b_{PGSE} is given by the Stejskal-Tanner equation :

$$\frac{S(b_{PGSE}, \hat{\mathbf{g}})}{S_0} = e^{-b_{PGSE} \hat{\mathbf{g}}^T \mathbf{D} \hat{\mathbf{g}}}, \quad (1.26)$$

where $S(b_{PGSE}, \hat{\mathbf{g}})$ is the signal obtained for a gradient in unit direction $\hat{\mathbf{g}}$ and b-value b_{PGSE} , S_0 is the signal without any diffusion gradient.

To estimate the diffusion tensor, since it is symmetric, a minimum of six gradient directions are required in addition to the base signal (i.e. without diffusion) called the **b0 volume** since it has a b-value $b = 0[s/mm^2]$. In practice, more than six gradients and one b0 are acquired as having more information allows to better correct for the large number of artifacts and noise affecting diffusion datasets (see section 1.4 and chapter 2).

Metrics derived in DTI : From the diffusion tensor, several metrics can be extracted to represent the state of the brain microstructure.

The mean diffusivity (MD) is given by the mean of the eigenvalues of the diffusion tensor and represents how much the water molecules are restricted in a given voxel

$$MD = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3}. \quad (1.27)$$

In order to obtain anisotropy indices, the axial diffusivity (AD) and the radial diffusivity (RD) can be computed and compared. They represent respectively the mean diffusivity in the axial and radial directions.

$$AD = \lambda_1 \quad (1.28)$$

$$RD = (\lambda_2 + \lambda_3)/2. \quad (1.29)$$

However, the most popular DTI metric is the fractional anisotropy (FA). The FA values goes from 0 to 1, 0 meaning that all eigenvalues are equal $\lambda_1 = \lambda_2 = \lambda_3$ and that the diffusion is isotropic (e.g. in the Cerebrospinal Fluid (CSF)) and 1 meaning that one

direction λ_1 is largely dominant for the diffusivity and the diffusion is strongly anisotropic.

$$FA = \sqrt{\frac{3}{2}} \frac{\sqrt{(\lambda_1 - MD)^2 + (\lambda_2 - MD)^2 + (\lambda_3 - MD)^2}}{\sqrt{\lambda_1^2 + \lambda_2^2 + \lambda_3^2}}. \quad (1.30)$$

Other possible representations exist such as the colored FA, a map where a color is attributed to each voxel depending on the direction of the first eigenvalue and the intensity of the color depends on the FA value.

Moreover, from the directional information and the diffusivity, it is possible to estimate the white matter tracts.

Advantages of DTI :

- Fairly simple model which can be rapidly fitted.
- Provides useful clinical metrics related to biological features, such as the myelination or the tissue organization.

Limitations of DTI :

- Difficult to interpret, a change in a DTI statistic cannot be attributed to a specific change in microstructure. The metrics give only indirect measures of the brain microstructure and several structures with different properties could present the same DTI properties. DTI is a phenomenological and not a microstructural model meaning that it fits a mathematical expression to the signal with parameters not directly representing the biological microstructure.
- Does not account for crossing fascicles while it occurs in 60% to 90% of the voxels (Jeurissen et al., 2013). The model cannot capture complex configurations of fascicles.

1.2.3 Tractography

DTI metrics only give voxel wise estimates of the white matter microstructure. Tractography is a method that integrates these local estimates of diffusion direction information to identify white matter pathways in the brain. The idea is that when the orientation of fibers is known in every voxel, it is possible to reconstruct an approximation of anatomical tracts. The fiber direction is estimated using the assumption that when a number of axons align themselves along a common axis, the diffusion of water molecules will be hindered to a greater extent across this axis than along it.

The most simple form of tractography consist in defining a seed location as a starting point and to follow the preferred direction until a new voxel is reached. The new voxel main direction is then chosen as the preferred direction until the entire tract is reconstructed. Following the Frenet–Serret formulas (Frenet, 1852), the path taken by the fiber streamlines can be formulated as a curve $\mathbf{r}$ parameterized by the arc length s of the tract.

$$\frac{d\mathbf{r}(s)}{ds} = \mathbf{t}(s),$$

where $\mathbf{t}(s)$ is the tangent vector of the curve (Basser et al., 2000). The reconstructed diffusion tensor $\mathbf{D}$ can be treated as a matrix. By equating the eigenvector ε_1 corresponding to the largest eigenvalue with the direction of the curve, the equation can be solved :

$$\frac{d\mathbf{r}(s)}{ds} = \varepsilon_1(\mathbf{r}(s)).$$

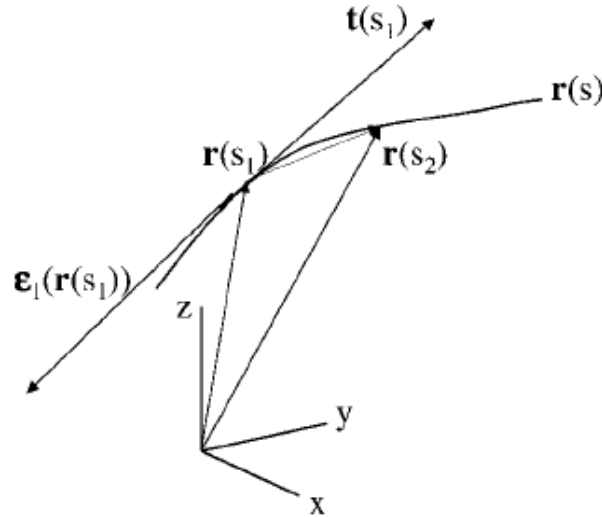


Figure 1.13: Representation of a white matter fiber trajectory as a space curve, $\mathbf{r}(s)$. The local tangent vector $\mathbf{t}(s_1)$, is identified with the eigenvector, $\epsilon_1(\mathbf{r}(s_1))$, associated with the largest eigenvalue of the diffusion tensor $\mathbf{D}$ at position $\mathbf{r}(s_1)$. From (Basser et al., 2000).

The system of three implicit vector differential equations for the fiber tract trajectory can be solved using numerical analyzes methods. The initial condition is $\mathbf{r}(0) = \mathbf{r}_0$, where $\mathbf{r}_0$ is the seed point.

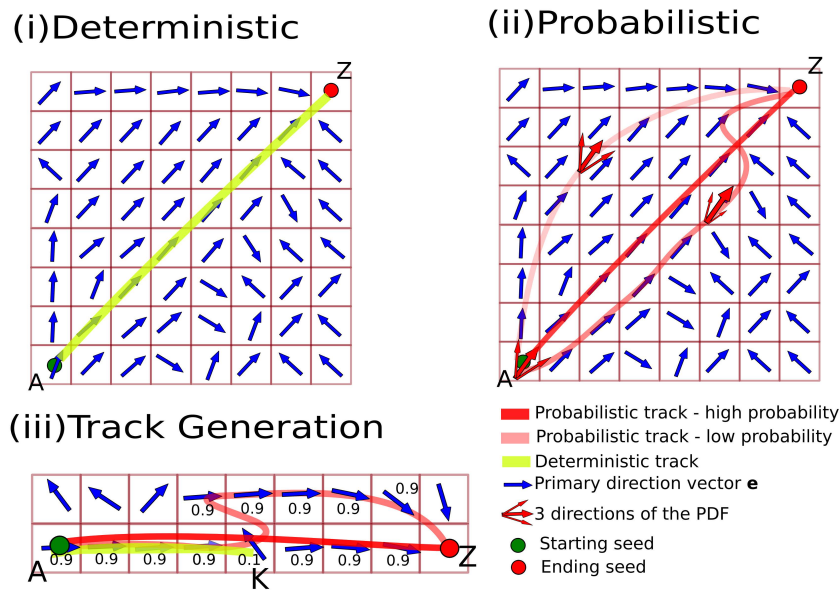


Figure 1.14: A simplified example showing in (i) and (ii) the same data set. (i) The yellow line shows the result of deterministic tractography which is given by a single trajectory and in (ii) is given by connectivity matrix depicting in red the probability of different pathways throughout the whole slice. For the ease of understanding, only three possible pathways are depicted. Finally, in (iii) an example is given where it is shown that probabilistic tractography weights more closer connections. However, it can track further deep than deterministic tractography. From (Garyfallidis, 2013).

Two classes of tractographic algorithms are commonly used for tract propagation. Probabilistic algorithms which includes uncertainty in their model and deterministic algorithms which does not assume any uncertainty along the path of the tract. The simplified 2D example in Fig. 1.14 depict the change among the two approaches (Garyfallidis, 2013).

Deterministic algorithms propagate streamlines based on a sequence of distinct locally optimum decisions. The streamlines are generated from a specific initial direction at a specific location in the volume. The starting location is known as the seed. When a low anisotropy region is found, the streamline ends its propagation. This prevent the generation of tracts inside the CSF where anatomical tracts do not exist or within gray matter where tracking can be ambiguous. Deterministic algorithm also discard propagation direction that made large angular changes (e.g. angle exceeding 90 degree) between consecutive steps to elude unrealistically sharp turns. Deterministic methods can also be used when multiple fibers orientations are estimated in a single voxel (crossing fibers). While fast, deterministic algorithms are sensitive to local noise.

Probabilistic tractography algorithms are more robust to noise than deterministic ones and can propagate streamlines in regions where the tracking is not straightforward thanks to the inclusion of uncertainty on the estimation of the fibers orientation at all track points. Rather than calculate a single streamline, a spatial distribution of streamlines arising from a single seed is computed. For all the propagation step of every streamlines, a random perturbation of the underlying fiber orientation estimate is pursue. Perturbations are generated using functions that characterize the uncertainty in the fiber orientation within each voxel. The different existing probabilistic approaches mainly differ in the way the orientation uncertainty is assessed.



Figure 1.15: White matter tracts using a DTI-based tractography. Each color represent a different atlas-based segmented region, from (Huisman, 2010).

Since the white matter is very dense, the frontier between different tracts are very difficult to distinguish. However, the white matter contains pathways known as fiber tracts that

connect functional areas of the brain. Using segmentation generally based on a structural T1 image or using local and global streamline-based and connectivity-based registration and clustering, tracts can be grouped and anatomically labeled, allowing regional comparison to be made across different subjects. A visual example of tractography segmentation is available in Fig.1.15.

1.2.4 Echo Planar Imaging

In conventional spin-echo, one line of data, in the k-space, is collected in one repetition time (TR) period. The measurement time is thus equal to the number of phase encoding steps (i.e. the number of lines) multiplied by the TR for each slice. As the repetition time is fairly long (e.g. $2000ms$), this results in an unaffordable acquisition time T_{acq} for DW-MRI. Moreover, diffusion MRI with conventional acquisition strategies is susceptible to motion artifacts since the molecular motion is significantly smaller than the macroscopic motion due to breathing, involuntary head motion or pulsatile brain motion (Bhagat, 2008).

To reduce the acquisition time, Echo Planar Imaging (EPI) is used instead of conventional spin echo. The EPI sequence is a rapid acquisition method that has become the clinical mainstay of diffusion. EPI allows the collection of data required for image reconstruction within a few hundred milliseconds resulting in high temporal resolution. This allows in particular to generate almost pure diffusion weighted images and to reduce motion within each slice.

In Fig.1.16, the acquisition scheme of EPI is represented. Compared to a traditional spin echo sequence, EPI uses time varying gradients to traverse the k-space. By applying an alternating frequency encoding gradient G_x and a blipped phase encoding gradient G_y , multiple lines are acquired within a single TR. In single-shot EPI, all Fourier lines are acquired after a single RF excitation pulse speeding up the acquisition considerably.

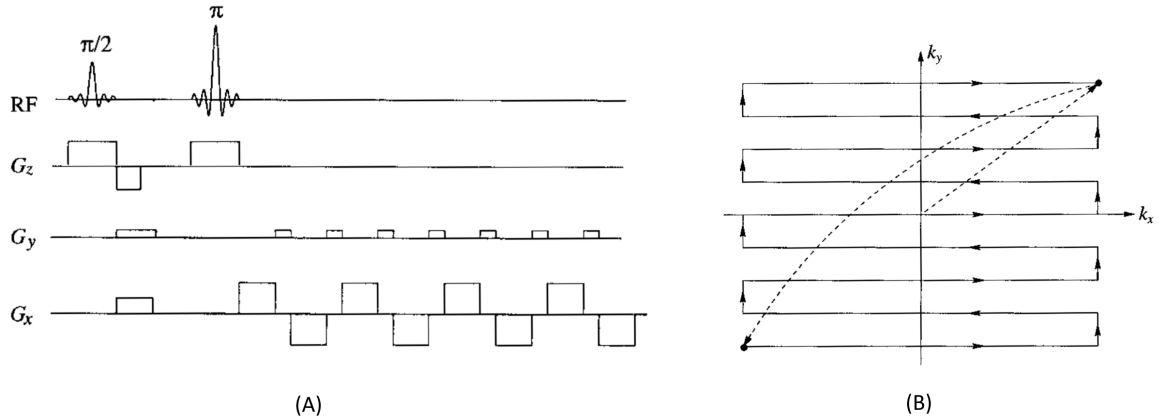


Figure 1.16: (A) Single shot blipped Echo Planar Imaging sequence (Spin echo) (B) k-space trajectory for blipped EPI. All data lines are acquired after a single excitation, each line being separated by a brief pulse of the phase encoding gradient, from (F.Peeters, 2017).

The limitations of using single-shot EPI are apparent in the quality of the derived images. Echo planar imaging suffers from increased T_2^* weighting (effect of static magnetic field

inhomogeneities), distortions and signal dropout (susceptibility effects), misregistration of the fat signal (Chemical shift artifacts) and reduced Signal-to-Noise Ratio (SNR). Moreover, EPI often provides less spatial resolution than conventional sequences due to gradient limitations (around 2mm) (Bhagat, 2008). Even though the receiver Bandwidth is high in EPI due to the rapid sampling rate along the readout axis, the continuous phase encoding scheme used in the sequence results in a relatively low bandwidth along the phase encoding axis (of the order of $30Hz/Px$). This causes severe artifacts to occur in EPI in the presence of frequency shifts. This is further discussed in section 1.4

To speed up the acquisition even more, it is also possible to do multi-band acquisition where multiple slices are acquired at the same time in an interleaved manner (in order to reduce slice cross-talk). The multi-band factor represents the number of slices acquired simultaneously. An example is shown in Fig.1.17

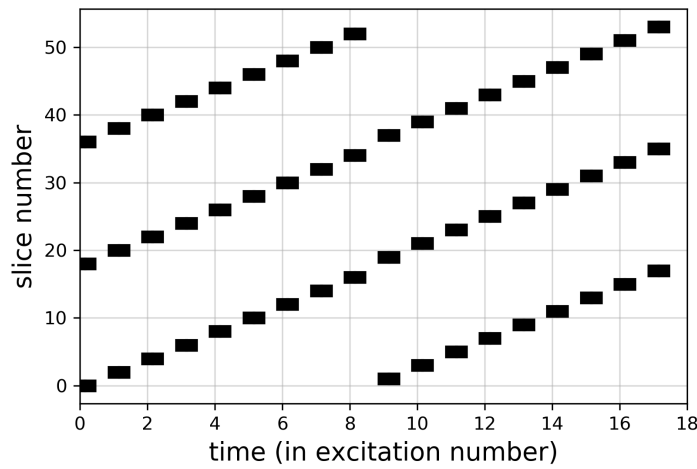


Figure 1.17: Slice acquisition scheme for 54 slices acquired in an interleaved manner with a multi-band factor of 3. Each black rectangle represents the acquisition of a specific slice.

In addition to interleave the acquisition of the slices within a volume, some very specific protocols might interleave also the acquisition of the slices across volumes of different diffusion gradients. This is called gradient cycling and is discussed more deeply in Chapter 3 as this has some particular consequences for the correction of motion artifacts (see section 1.4). The two first chapter omit the possibility of gradient cycling.

1.3 Beyond DTI, models of the brain microstructure

As explained in section 1.2.2, DTI suffers from several disadvantages and therefore more advanced diffusion models have been developed for estimating the microstructural complexity of dendrites and axons and to overcome the limitations of DTI. In this section three models are presented : Neurite Orientation Dispersion and Density Imaging (NODDI) (Zhang et al., 2012), DIstribution of 3D Anisotropic MicrOstructural eNvironment in Diffusion compartment imaging (DIAMOND) (Scherrer et al., 2016) and Microstructure Fingerprinting (MF) (Rensonnet et al., 2019). These three models are multicompartement models based on the superposition principle to infer microstructural features. This principle states that the signal measured can be decomposed as the union of individual signals emerging from mutually disjoint and independent structures with perfectly reflecting boundaries.

1.3.1 NODDI

Neurite Orientation Dispersion and Density Imaging (NODDI) is a microstructural multi-compartment model that considers in each voxel three compartments divided in two levels (see Fig.1.18) (Zhang et al., 2012) :

- **Intra-cellular** compartment : space bounded by the membrane of neurites.
- **Extra-cellular** compartment : space around the neurites, occupied by various types of glial cells.
- **CSF** compartment : space occupied by cerebrospinal fluid.

Each of those compartments is modeled independently and affects water diffusion. They give rise to separate normalized MR signals and their contributions are then combined based on the superposition principle. The total normalized signal A can be therefore written as

$$A = (1 - \nu_{iso}) (\nu_{ic} A_{ic} + (1 - \nu_{ic}) A_{ec}) + \nu_{iso} A_{iso}, \quad (1.31)$$

where A_{ic} and ν_{ic} are the normalized signal and volume fraction of the intra-cellular compartment; A_{ec} is the normalized signal of the extra-cellular compartment; and A_{iso} and ν_{iso} are the normalized signal and volume fraction of the CSF compartment (Zhang et al., 2012). Below are described the models used for each compartment.

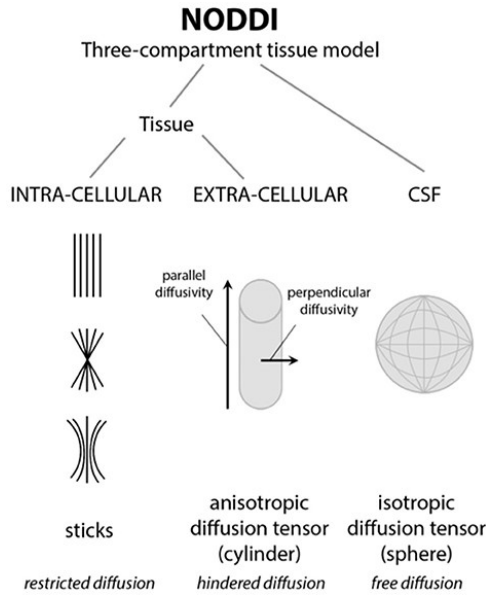


Figure 1.18: Models of diffusion for NODDI. It considers diffusion within three compartments : restricted diffusion in the intra-cellular compartment, hindered diffusion in the extra-cellular compartment, and free diffusion in cerebrospinal fluid (CSF), from (Barritt et al., 2018).

Intra-cellular model : The intra-cellular region refers to the space bounded by the membrane of neurites. It is modeled as a set of sticks/cylinders with zero radius to capture the highly restricted nature of diffusion perpendicularly to neurites and unhindered diffusion along the neurites. The total signal of this compartment over all possible sticks orientations $\mathbf{n}$ is given by

$$A_{ic} = \int_{\mathbb{S}^2} f(\mathbf{n}) \underbrace{e^{-bd_{\parallel}(\mathbf{q}, \mathbf{n})^2}}_{\text{Individual stick signal}} d\mathbf{n}, \quad (1.32)$$

where $\mathbf{q}$ and b are the diffusion gradient direction and b-value, $d_{\parallel}$ is the axial diffusivity of the intra-neurite space and $f(\mathbf{n})$ is the stick distribution that gives the probability of finding sticks along orientation $\mathbf{n}$.

To account for the fact that the sticks are not perfectly parallel $f(\mathbf{n})$ is modeled with a Watson distribution :

$$f(\mathbf{n}) = M \left(\frac{1}{2}, \frac{3}{2}, \kappa \right)^{-1} e^{\kappa(\boldsymbol{\mu} \cdot \mathbf{n})^2}. \quad (1.33)$$

This distribution is parametrized by μ , the principal orientation of the sticks and κ , the concentration parameter (i.e. spread around the mean of the stick distribution).

Extra-cellular model : The extra-cellular region refers to the space around the neurites occupied by various types of glial cells. The diffusion in this compartment is hindered by the presence of neurites but not restricted. Therefore, it is modeled with a gaussian anisotropic diffusion. The normalized attenuation signal is given by

$$A_{ec} = e^{-b\mathbf{q}^T \left(\int_{\mathbb{S}^2} f(\mathbf{n}) D(\mathbf{n}) d\mathbf{n} \right) \mathbf{q}}. \quad (1.34)$$

$D(\mathbf{n})$ is a cylindrically symmetric tensor with the principal direction of diffusion $\mathbf{n}$, diffusion coefficients $d_{\parallel}$ parallel to $\mathbf{n}$ (fixed to the same value as the intrinsic free diffusivity of the intra-cellular compartment) and $d_{\perp} = d_{\parallel} (1 - v_{ic})$ perpendicular to $\mathbf{n}$. The diffusion is described by a tensor since the diffusion perpendicularly to the axon is not neglected anymore for this compartment.

CSF model : The CSF region refers to the space occupied by cerebrospinal fluid. It is modeled as an isotropic gaussian diffusion with diffusivity d_{iso} . The normalized attenuation signal is given by

$$A_{iso} = e^{-bd_{iso}}, \quad (1.35)$$

with d_{iso} the isotropic diffusivity.

Parameters : The overall parameters to be estimated by NODDI are : the intra-cellular volume fraction ν_{ic} , the isotropic volume fraction ν_{iso} , the concentration parameter of Watson distribution κ , the principal orientation of Watson distribution μ , the isotropic diffusivity d_{iso} and the intra-neurite diffusivity $d_{\parallel}$.

The diffusivities are fixed to their respective typical values in vivo : $d_{\parallel} = 1.7 \times 10^{-3} mm^2 s^{-1}$ and $d_{iso} = 3.0 \times 10^{-3} mm^2 s^{-1}$. The remaining parameters are estimated by fitting the model to the data using maximum likelihood estimates.

After estimation, another metric is computed from the κ value, the Orientation Dispersion Index (ODI).

$$ODI = \frac{2}{\pi} \arctan(1/\kappa) \quad (1.36)$$

The ODI characterizes the dispersion of the neurites in each voxel. It ranges from zero to one. A value of zero meaning perfectly parallel fibers and a value of one indicating randomly oriented fibers.

Advantages of NODDI :

- Captures useful clinical metrics related to biological features such as the neurite densities and orientations.
- Allows more insight than DTI through the compartments.

Limitations of NODDI :

- Neglects the radial diffusivity of the axons.
- Requires to prefix some parameters (i.e. $d_{\parallel}$ and d_{iso}) leading to biased values for the estimated parameters.
- Does not account for crossing fascicles while it occurs in 60% to 90% of the voxels (Jeurissen et al., 2013). The model cannot capture complex configurations of fascicles.

1.3.2 DIAMOND

DIstribution of 3D Anisotropic MicrOstructural eNvironment in Diffusion compartment imaging (DIAMOND) is a microstructural multicompartment model that considers that the measurement in each voxel comes from a great number of spin packets and that each of those spin packets underwent a globally homogeneous diffusion. In other words, DIAMOND tries to regroup the spins of each voxel in packets that experience the same diffusion.

Overall, using the superposition principle, the dMRI attenuation signal is modeled as the sum of the signal contributions of the different compartments. The diffusion of each spin packet is described using a diffusion tensor $\mathbf{D}$ that captures the restriction and hindrance of the diffusion (Scherrer et al., 2016). The contribution of each spin packet is given by

$$S_0 e^{(-b_k \mathbf{g}_k^T \mathbf{D} \mathbf{g}_k)} d\mathbf{D}, \quad (1.37)$$

where $\mathbf{g}_k$ and b_k are the diffusion gradient direction and b-value of volume k and S_0 is the b_0 baseline signal (i.e. the signal without diffusion weighted acquired at $b = 0$). The diffusion weighted signal S_k is thus expressed as

$$S_k = S_0 \int_{\mathbf{D} \in \text{Sym}^+(3)} P(\mathbf{D}) e^{-b_k \mathbf{g}_k^T \mathbf{D} \mathbf{g}_k} d\mathbf{D}. \quad (1.38)$$

$P(\mathbf{D})$ is a matrix-variate distribution over the tensor space that characterizes the fraction of spins with diffusion tensor $\mathbf{D}$. If a voxel consists of exactly one homogeneous population described by a diffusion tensor $\mathbf{D}_0$, $P(\mathbf{D})$ would be a delta function $\delta(\mathbf{D} - \mathbf{D}_0)$ and the DIAMOND model reduces back exactly to DTI. In multi-tensor model (MTM), the spin population are considered to be homogeneous and $P(\mathbf{D})$ is modeled as a sum of delta functions (each delta corresponding to a spin packet). However, DIAMOND considers a more general case by using multivariate Gamma distributions in order to take into account the heterogeneity of the packets. This is represented in Fig.1.19.

The environment usually consists of several identifiable microstructural environments with some degrees of heterogeneity and the distribution of each spin packet is modeled with a peak shaped mv- Γ distribution centered around a mean $\mathbf{D}_0$ and parameterized by a scale parameter Σ and a shape parameter κ :

$$P_{\kappa, \Sigma}(\mathbf{D}) = \frac{|\mathbf{D}|^{\kappa - \frac{p+1}{2}}}{|\Sigma|^{\kappa} \Gamma_p(\kappa)} e^{-\text{trace}(\Sigma^{-1} \mathbf{D})}, \quad (1.39)$$

with Γ_p the multivariate Gamma function. When considering all N_p spin packets, the matrix variate distribution $P(\mathbf{D})$ becomes

$$P(\mathbf{D}) = \sum_{j=1}^{N_p} f_j P_{\kappa_j, \Sigma_j}(\mathbf{D}). \quad (1.40)$$

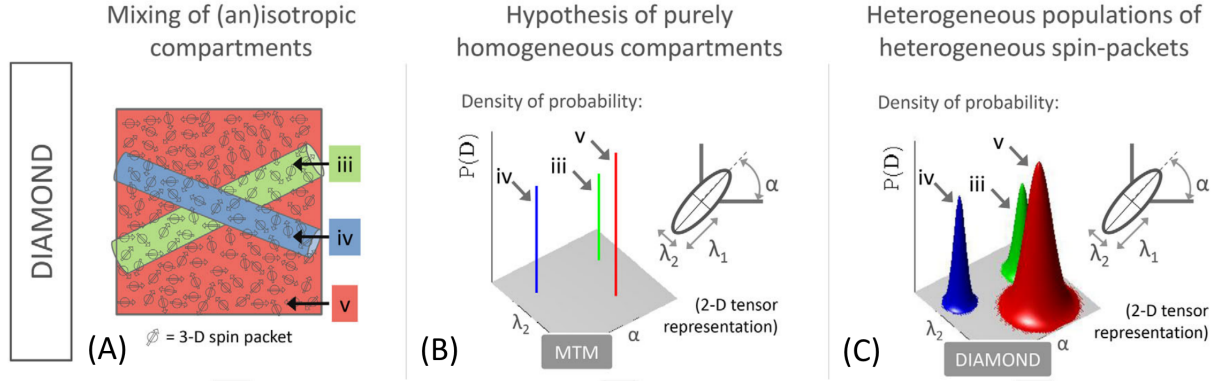


Figure 1.19: DIAMOND spin packets modelization. (A) Multicompartment model with three compartments represented in red, green and blue. (B) If considering homogeneous compartments, a multi-tensor model (MTM) is obtained for $P(\mathbf{D})$ with a delta function for each compartment. (C) DIAMOND multivariate Gamma distributions describing the diffusivity with heterogeneities in each compartment, from (Scherrer et al., 2016)

This expression can be injected in equation 1.38, and after some calculus, the expression of the signal becomes :

$$S_k = S_0 \sum_{j=1}^{N_p} f_j \mathcal{D}(\mathbf{D}_j^0, \kappa_j), \quad (1.41)$$

where $\mathbf{D}_j^0 = \kappa_j \Sigma_j$ is the expectation of the j th mv- Γ distribution and

$$\mathcal{D}(\mathbf{D}^0, \kappa) = S_0 \left(1 + \frac{b_k \mathbf{g}_k^T \mathbf{D}^0 \mathbf{g}_k}{\kappa} \right)^{-\kappa}. \quad (1.42)$$

Finally, to highlight the possible presence of an isotropic compartment, the expression can be rewritten as :

$$S_k = S_0 \left(\sum_{j=1}^{N_p} f_j \mathcal{D}(\mathbf{D}_j^0, \kappa_j) + f_{iso} e^{-b d_{iso}} \right). \quad (1.43)$$

Parameters : The overall parameters to be fitted to the data y by DIAMOND are : the shape parameters κ , the expectations of the mv- Γ distribution $\mathbf{D}_j^0$ and the fractions of the spin packets f . The parameters at each voxel are estimated using a maximum a posteriori approach $p(\mathbf{y} | \mathbf{D}^0, \kappa, f)$ considering normally distributed noise. DIAMOND also requires the estimation of the number of tissue compartments in each voxel.

Advantages of DIAMOND :

- Multicompartment model which considers several fascicles by voxel and can deal with crossing fascicles.
- DIAMOND is a generalization of DTI. Therefore, it captures the DTI metrics for each compartment (i.e. cFA, cRD, cAD, cMD) in addition to measure the compartments heterogeneity.
- Provides the fraction of each compartment and captures their microstructural changes.

Limitations of DIAMOND :

- Complex model that necessitates a computationally intensive data fitting.

- Phenomenological model which fits a mathematical expression to the signal with parameters not directly representing the biological microstructure.
- The DIAMOND metrics are estimated voxelwise meaning that in practice there are no guarantees that the fascicles labeled into a specific compartment will be labeled the same way for another brain or between adjacent voxels.

1.3.3 Microstructure Fingerprinting

Microstructure Fingerprinting (MF) is a microstructural multicompartment model that combines Monte Carlo simulations of the diffusion of water molecules with dictionary matching to find which parameters best describe the underlying microstructure in each voxel (Rensonnet, 2019). Similarly to the DIAMOND model, each voxel is considered to be composed of different structures which are combined based on the superposition principle. For a white matter voxel composed of K independent fascicles with principal unit orientations $\mathbf{u}_k$ occupying fractions ν_k of the physical volume of the voxel and for a partial volume ν_{csf} of cerebrospinal fluid (CSF), the diffusion weighted signal S is given by (Rensonnet et al., 2019)

$$S = M_0 \cdot \left[\sum_{k=1}^K \nu_k A_{fasc}(\Omega_k, \mathbf{T}_k, \mathbf{u}_k; \mathbf{g}) + \nu_{csf} A_{csf}(D_{csf}, \mathbf{T}_{csf}; \mathbf{g}) \right] \quad (1.44)$$

$$= \sum_{k=1}^K w_k A_k + w_{csf} A_{csf}. \quad (1.45)$$

M_0 is the net initial transverse magnetization of the voxel and $A_k(\Omega_k, \mathbf{T}_k, \mathbf{u}_k; \mathbf{g})$ is the normalized DW-MRI signal of the k -th fascicle modeled by Monte Carlo simulation for an orientation $\mathbf{u}_k$, microstructural parameters Ω_k , NMR relaxation parameters T_k (e.g. T_1 , T_2) and gradient profile $\mathbf{g}$. Water is assumed to diffuse freely in the CSF compartment and is therefore modeled with a scalar diffusivity D_{CSF} . The weights w_k are defined as $w_k = M_0 \nu_k$.

The set of $[A_{fasc}(\Omega, \mathbf{T}, \mathbf{u}; \mathbf{g})]_{i=1}^M$ is defined as a *fingerprint* (Rensonnet et al., 2019).

During runtime, the inverse problem is solved. The precomputed Monte Carlo DW-MRI fingerprints are used to find the optimal configuration of fingerprints that best explains the diffusion signal in every voxel. This process can be described as follows (Rensonnet et al., 2019) :

Precomputing steps (before any data is even acquired)

1. A discrete sampling of N points of the Ω -space of microstructural parameters is performed with parameters justified by tissue biology. This model represents the fascicles with a hexagonal packing of impermeable cylinders with radius r and spacing s .
2. A single fascicles dictionary $\mathbf{F}^0 \in \mathbb{R}^{M \times N}$ is built containing the fingerprints of the N selected microstructural configurations for a single fascicle along a fixed direction u_0

$$\mathbf{F}^0 = [\mathbf{A}_1^0, \dots, \mathbf{A}_N^0], \text{ with } \mathbf{A}_j^0 = [A_{fasc}(\Omega_j, \mathbf{T}_0, \mathbf{u}_0; \mathbf{g}_i)]_{i=1}^M \quad (1.46)$$

Runtime steps

1. An external routine is used to estimate in each voxel the number k of fascicles and their orientations.

2. From $\mathbf{F}^0$ in direction $\mathbf{u}_0$, a routine is applied to compute single fascicle dictionaries $\mathbf{F}^k$ in directions $\mathbf{u}_k$, $1 \leq k \leq K$ with K estimated in the previous step.
3. The optimization problem is solved under sparsity constraints which ensures that only one fascicle configuration per dictionary $\mathbf{F}^k$ contributes to the signal $\mathbf{y}$

$$(\hat{j}_1, \dots, \hat{j}_K) = \underset{1 \leq j_1, \dots, j_K \leq N}{\operatorname{argmin}} \min_{\mathbf{w} \geq 0} \left\| \mathbf{y} - [\mathbf{A}_{j_1}^1 | \dots | \mathbf{A}_{j_K}^K | \mathbf{A}_{\text{csf}}] \cdot \begin{bmatrix} w_1 \\ \vdots \\ w_K \\ w_{\text{csf}} \end{bmatrix} \right\|_2^2 \quad (1.47)$$

The solution of this problem provides the optimal fingerprints j_k for the dictionaries $\mathbf{F}^k$ that best explain the measured signal.

The fractions of occupancy ν_k are estimated from the optimal weights w_k

$$\nu_k = \frac{w_k}{\sum_{k=1}^{K+1} w_k}. \quad (1.48)$$

Using the fact that axons are modeled using a hexagonal packing, the fiber volume fraction which represents the intra-axonal volume fraction can also be computed as

$$fv_{\text{fasc}} = \frac{2\pi}{\sqrt{3}} \left(\frac{r}{s} \right)^2. \quad (1.49)$$

The total fiber volume fraction represents the total fraction of water inside axons in the voxel and is obtained by computing (for two fascicles) :

$$fv_{\text{tot}} = f_1 * fv_{f_1} + f_2 * fv_{f_2}. \quad (1.50)$$

Advantages of MF :

- Multicompartment model which considers several fascicles by voxel and can deal with crossing fascicles.
- Monte Carlo simulations of the random walk of water molecules is known to provide near groundtruth for DW-MRI signals and experiments have shown that it exhibits results generally more specific and interpretable than metrics obtained with forward signal models such as NODDI or DIAMOND (Rensonnet, 2019).
- Provides reliable and physically interpretable tissue parameters in a wide range of synthetic, in vivo and ex vivo experiments.
- Thanks to the superposition approximation for crossing fascicles and accelerated methods based on DNN, increased model complexity are still computationally doable.

Limitations of MF :

- Computational cost for the creation of the dictionary (i.e. precomputing steps).
- Membrane permeability, axonal undulation and glial cells are not yet incorporated in the model.
- The MF metrics are estimated voxelwise meaning that in practice there are no guarantees that the fascicles labeled into a specific compartment will be labeled the same way for another brain or between adjacent voxels.

1.4 Artifacts in Magnetic Resonance Imaging

As explained in section 1.2.4, diffusion images are often subject to low SNRs and a multitude of artifacts. This section aims at explaining the characteristics of the most common artifacts found in DW-MRI.

Motion artifacts : In MRI, a major source of artifacts is the patient motion. Due to the large time window required for the acquisition, it is difficult for patients to stay completely still (head motion, pulsating flow, cardiac motion, etc). This proves to be especially harmful in diffusion MRI as the molecular motion is negligible compared to the possible macroscopic motion making the signal of interest non measurable. Moreover, when the motion is not synchronous with the phase encoding cycle, ghosting artifacts will appear (Le Bihan et al., 2006). As explained in section 1.2.4, the use of EPI sequences solves both of these issues. An image is acquired within a single shot (100 msec), and hence motion is effectively frozen during the acquisition of each slice at the cost of a lower spatial resolution.

Although motion-induced ghosting is totally cured with EPI, motion may still act in other ways on the data. Indeed, in diffusion MRI, a series of images are acquired at different b-values. The time between these images is non negligible and in the presence of motion, a given voxel within the image is not exactly at the same location throughout the data set. This type of artifact is called **volume-to-volume** (V2V) motion and can be seen as a rigid movement between each acquired volume. This artifact causes problems when estimating diffusion metrics since corresponding voxels are not aligned which causes signal intensity variation in the voxels.

A common assumption made is that the subject remains motionless during the several seconds it takes to acquire all slices in a volume, and that any movement occurs in the brief moment between acquiring the last slice of one volume and the first slice of the next. However, in cases of large motion, this assumption does not hold and other artifacts appears than the volume-to-volume motion. This is called **slice-to-volume** (S2V) motion. It is responsible for signal losses (when a movement coincides exactly in time with the diffusion encoding part of the sequence) and slices irregularly sampled in space (when movement occurs between the acquisition of slices within a volume) (Andersson et al., 2017).

Eddy current distortions : Eddy currents are electric currents that are generated in the body when the magnetic gradients are switched on and off. These currents generate their own magnetic field that combine with the imaging gradient pulses such that the actual gradients experienced by the spins are not exactly the same as those that were programmed to produce the image. As a result, the image is affected by geometric distortions such as contraction, dilatation and overall shift and shear (Le Bihan et al., 2006).

These eddy current distortions become even more relevant in diffusion MRI. Indeed, due to the rapid gradient switching in EPI, DW-MRI suffers from increased distortions. Moreover, since diffusion images are calculated with different diffusion sensitization (i.e. b-values), the distortion varies between the images.

The eddy current distortions are usually minimized during the acquisition using shielding and pre-emphasis techniques but still residual eddy currents remain and a postprocessing correction is needed.

Susceptibility artifacts : Susceptibility artifacts are created due to the local internal distortions of the magnetic field. All substances get magnetized when placed in a magnetic field. This is characterized by their magnetic susceptibility (χ). The magnetic induction,

called magnetic field in MRI, is thus given by :

$$\mathbf{B} = \mu \mathbf{H} = \mu_0(1 + \chi) \mathbf{H}, \quad (1.51)$$

where $\mathbf{H}$ is the 'real' magnetic field (in contrast to $\mathbf{B}$ which is the magnetic flux density, called magnetic field in MRI as an abuse of language), μ_0 is the permeability of the vacuum and μ the permeability of the substance.

Susceptibility artifacts occur at interfaces of different χ (e.g. air-tissue interfaces). These differences in susceptibilities create distortions in the local field causing dephasing of spins (signal dropouts) and geometric distortions of the image (F.Peeters, 2017).

Compared to the Eddy current distortions which vary with the b-value, the susceptibility distortion is as a first approximation constant during a diffusion scan. The Spin Echo - Echo Planar Imaging (SE-EPI) sequence is particularly sensitive to this type of artifact because of its small bandwidth along the PE direction. A typical bandwidth in the PE-direction is of the order of $30Hz/pixel$, which means that even a (very small) off-resonance field of $30Hz$ leads to a one pixel displacement (Bastiani et al., 2019). The direction of the phase encoding gradient or blips (see Fig.1.16) therefore dramatically affects the main direction of the distortion on the final image. By performing SE-EPI for a volume with phase encoding gradients having opposite polarities (blip-up and blip-down), it is thus possible to accurately estimate the susceptibility induced off resonance field. This property is used to correct for the artifact in section 2.2.2.5.

Chemical shift : The chemical shift artifact is caused by the difference in resonance frequency experienced in various chemical environments. Due to the differences of Larmor frequency, a spatial misregistration artifact appears at interfaces between tissues of different compositions such as fat and water. Signals from water and fat protons in the same spatial location will be assigned to different locations in the image (Hernando et al., 2011)(F.Peeters, 2017). Diffusion MRI is especially sensitive to chemical shift due to the low bandwidth of the EPI sequence and its high sensitivity to off-resonance effects. At $1.5T$ the fat/water chemical shift is about $220Hz$ while the bandwidth per pixel in the phase encoding direction for EPI is of the order of $30Hz/pixel$. This is translated into a noticeable artifact that can be up to several pixels wide (depending on the chosen parameters) (Jezzard and Balaban, 1995).

The main difference between chemical shift and susceptibility artifacts is the physical origin of the distortions. Chemical shift results from the differences in gyromagnetic ratio γ while susceptibility artifacts result from differences in magnetic susceptibility. Another main difference is that fat suppression techniques can be used during the acquisition to eliminate the chemical shift misregistration, making postprocessing techniques usually not necessary. Other methods to reduce chemical shift include increasing the readout gradient or decreasing the pixel size.

Aliasing : If the image is undersampled in the k-space, following the Nyquist-Shannon theorem, aliasing will appear. Aliasing causes data lying outside the Field Of View (FOV) to be wrapped back into the image. An easy way to solve this issue is to increase the FOV.

Gibbs Ringing : Gibbs ringing is a common artifact that arises due to the use of the Fourier Transform. The entire k-space cannot be acquired and is therefore truncated.

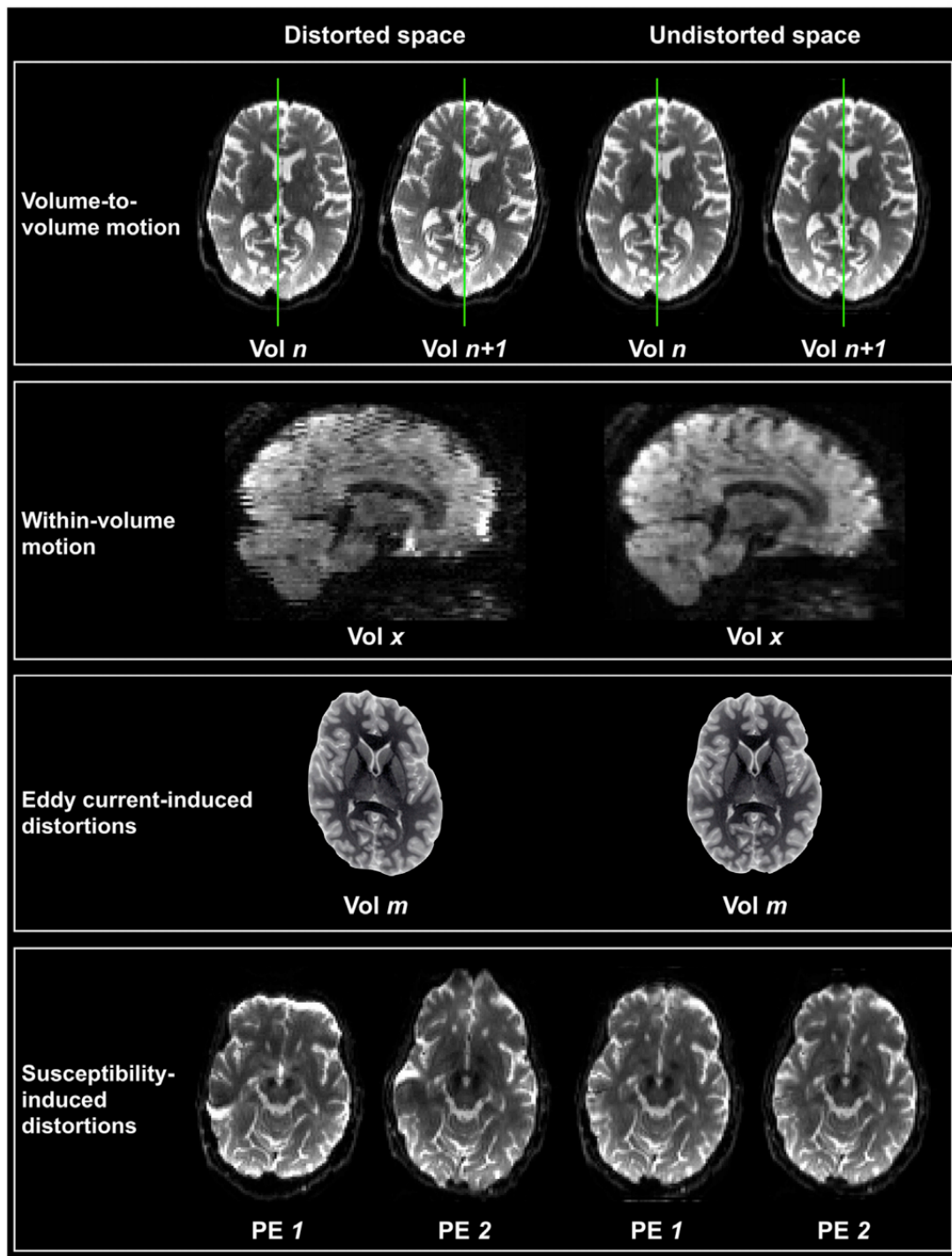


Figure 1.20: Representation of the motion artifacts, Eddy current distortions and susceptibility induced distortions in the distorted and undistorted space, from (Bastiani et al., 2019).

Truncation with a rectangular window in the k-space results in a convolution with a sinc function in the reconstructed image. This appears on the final image as spurious ringing around sharp edges. Gibbs ringing can be reduced by filtering with a smoothly decreasing (e.g. Hamming) window prior to processing or by increasing the size of the truncation window in the k-space for the same FOV in the image space. However, this comes at the cost of a lower spatial resolution and SNR.

Thermal Noise : The image is affected by thermal noise mainly caused by the patient himself and the coils. Its distribution is complex gaussian in the k-space resulting in a thermal Ricean noise in the image space. One can show that the Signal-to-Noise Ratio in MRI is proportional to (F.Peeters, 2017) :

$$SNR \propto \Delta V \sqrt{\frac{N_{ex} N_x N_y}{BW_r}}, \quad (1.52)$$

where ΔV is the voxel volume, N_{ex} is the averaging, $N_x N_y$ are the number of points in k-space and $BW_r = 1/\Delta t$ is the readout bandwidth.

For EPI, the readout bandwidth is much larger than for a classical spin echo sequence resulting in a significant decrease of SNR. Moreover, due to diffusion the signal is attenuated letting the noise becoming more prevalent especially at higher b-values.

Bias field : Inhomogeneity of the magnetic field inside the MR scanner is inescapable due to the intrinsic field inhomogeneity of MRI magnet caused by high-density receiver coils and inhomogeneity in the magnetic properties of the subject. This causes a spatial bias in the signal intensity of MR image data called Bias field. It is a low frequency undesirable signal that affects the MRI images. Bias field blurs images and thus reduces the high frequency contents of the image such as edges and contours and changes the intensity values of image pixels so that the same tissue has different graylevel distribution across the image (Juntu et al., 2005). This variability of the signal in tissues of the same type can affect microstructural metrics computation and brain segmentation algorithms (Zhang et al., 2001).

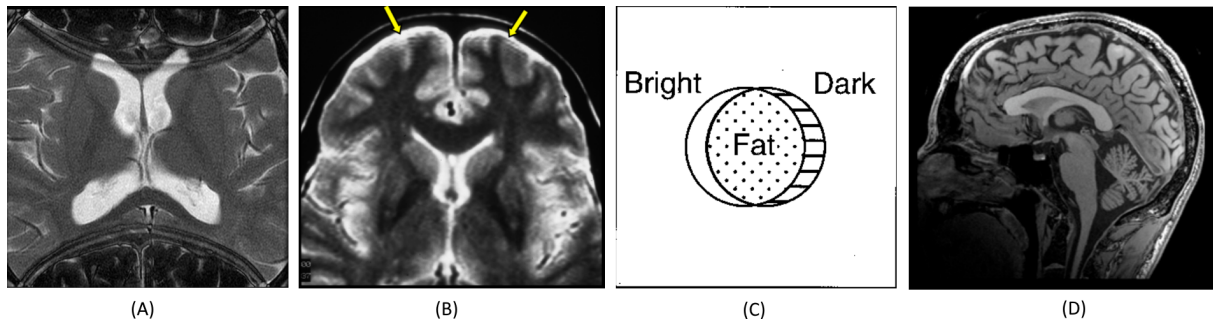


Figure 1.21: (A) Aliasing artifact causing a wrapping of the data (B) Gibbs ringing (C) Representation of the chemical shift between water and fat (D) Bias field, from (A. Haouimi, 2009) (Block et al., 2008) (Jaume et al., 2011).

Chapter 2

ElikoPy pipeline

As explained in section 1.4, DWI images are often subject to low SNRs and a multitude of artifacts. The echo planar imaging acquisition can result in susceptibility-induced and eddy current-induced distortions as well as slice-wise signal dropout. Imaging the brain repeatedly to acquire volumes along different gradient directions can result in long acquisition times and increased subject movement causing possible inter-volume motion artifacts. Overall, beyond MRI sequence and acquisition parameters, various sources of distortions influence the resulting diffusion metrics. However, uncovering the signal variability that is specific to diffusion process and its interference with the microstructure is crucial to ensure the validity of inference and the interpretability of results. This highlights the need for a complete processing pipeline with a clear and structured organization, to allow consistency and reproducibility of the studies.

This chapter covers the main state of the art pipelines for preprocessing diffusion-weighted magnetic resonance imaging data. Moreover, the novel preprocessing pipeline developed during this Master's thesis is described and validated throughout this chapter. The ElikoPy pipeline expands state of the art pipeline frameworks by providing a complete quality assessment and quality reports for each processed subject, providing a standardized framework to ensure reproducibility and consistency, reducing error propagation and improve sensitivity. It also grants the possibility for clinicians to perform fast preprocessing for a large variety of studies with minimal knowledge.

2.1 State of the art

dMRI technology has enormous potential, but it suffers from a unique and complex set of image quality problems, limiting the sensitivity of dMRI studies and reducing the accuracy of findings. Presently, several software packages and tools exist to correct those dMRI artifacts.

The FMRIB Software Library (FSL) (Jenkinson et al., 2012) offers a diffusion toolbox which includes tools for Eddy current correction and motion correction along brain extraction. Topup can estimate from a pair of data encoded in reversed phase-encoded blips the susceptibility-induced off-resonance field (Andersson et al., 2003). Eddy from FSL, a tool to correct for Eddy current-induced distortions and subject movements, simultaneously models the effects on diffusion of Eddy currents and movements on the image (Andersson and Sotiropoulos, 2016). BET (Brain Extraction Tool) deletes non brain tissue from an image of the whole head (Smith, 2002). The inner and outer skull surfaces along with

the outer scalp surface can also be estimated when having good quality T1 and T2 input images (Jenkinson et al., 2005).

DiPy, Diffusion Imaging in Python (Garyfallidis et al., 2014), contains a wide variety of generic methods for spatial normalization, signal processing, machine learning, statistical analysis and visualization of medical images. Additionally, it contains specialized methods for computational anatomy including diffusion and structural imaging. It can perform noise correction using Marchenko-Pastur Principal Component Analysis (MPPCA) algorithm (Veraart et al., 2016) and Gibbs ringing correction (Kellner et al., 2016). Along DiPy, the Diffusion Microstructure Imaging in Python (Dmipy) software package (Fick et al., 2019) facilitates the reproducible estimation of diffusion MRI-based microstructure features, especially those based on NODDI (Zhang et al., 2012).

MRtrix3 provides a set of tools to perform various types of diffusion MRI analyzes (Tournier et al., 2019). Like DiPy, MRtrix implements a dMRI noise level estimation and denoising based on MPPCA (Veraart et al., 2016) and Gibbs ringing correction using local sub-voxel shift (Kellner et al., 2016). Along these preprocessing functionalities, MRtrix3 also offers the possibility to compute a wide variety of microstructural metrics and perform tractography.

Unfortunately, most of these research-oriented software solutions are often not generally applicable to data from multi-site, multi-scanner studies with non uniform data formats, acquisition protocols and more. The processing of microstructural metrics for large datasets requires fully automatic pipelines that performs artifact and outlier removal and automated quality control prior to the calculation of microstructural metrics. Pipelines more adapted for these kinds of tasks integrating these softwares have thus been developed. There are a few customized tools performing preprocessing operations and quality report on large multi-site and/or multi-scanner dMRI datasets. A comparison of the steps performed by these pipelines is available in Fig.2.1 in regards to the steps performed by the ElikoPy pipeline.

The pipeline DTIPrep (Oguz et al., 2014) performs data format check, Rician noise removal through an linear minimum mean square error (LMMSE) filter and Eddy-current combined with motion artifact correction using FSL. However, DTIPrep does not fully integrate motion artifact correction and susceptibility-induced distortion correction. DTIPrep also performs DTI with subsequent directional artifact detection and correction.

Another pipeline, PreQual (Cai et al., 2021), performs denoising and intensities normalization on the raw data along with susceptibility-induced distortion correction and volume-to-volume motion correction. When no complementary phase encoded b0 images are supplied, an innovative approach based on Synb0-DisCo (Schilling et al., 2019) is used by PreQual to perform susceptibility-induced distortion correction. Synb0-DisCo utilize deep learning to synthesize an undistorted diffusion weighted image from the structural image, and use the non distorted synthetic image as an anatomical target for distortion correction. PreQual also performs quality assurance (QA) check and provides results in a QA report.

The Diffusion parameter EStimation with Gibbs and Noise Removal (DESIGNER) pipeline (Ades-Aron et al., 2018) has been developed to identify and minimize common sources of methodological variability such as thermal noise, Gibbs ringing artifacts, Rician bias, EPI and Eddy current induced spatial distortions, and motion-related artifacts.

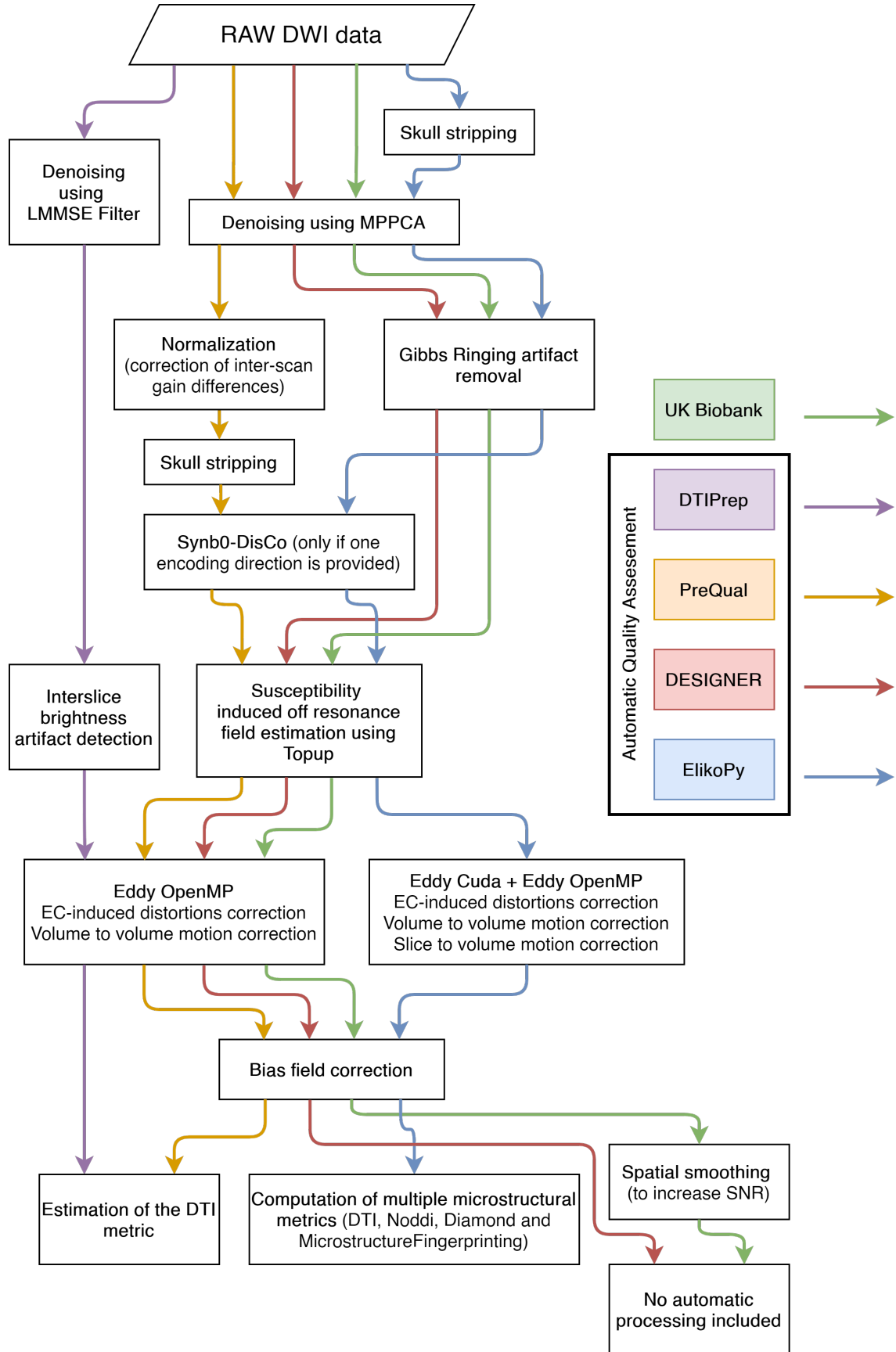


Figure 2.1: Schematic description of the state of the art Biobank (Maximov et al., 2019), DTIPrep (Oguz et al., 2014), PreQual (Cai et al., 2021) and DESIGNER (Ades-Aron et al., 2018) pipelines compared to the proposed ElikoPy pipeline.

One of the pipeline applied on the UK-Biobank (Maximov et al., 2019) performs the same succession of pipeline steps as visible in Fig.2.1 along with spatial smoothing to increase SNR before diffusion metric estimations.

Finally, all four pipelines discussed above lack the ability to easily process a large amount of subjects while performing complete motion correction, Gibbs ringing removal and denoising. Moreover, with the exception of the pipeline designed for the UK-Biobank and the DESIGNER pipeline, all these pipelines have been designed with DTI in mind. This precludes the use of newer microstructural models such as NODDI (Zhang et al., 2012), DIAMOND (Scherrer et al., 2016) or the novel Microstructure Fingerprinting (Rensonnet et al., 2019).

2.2 The ElikoPy pipeline

2.2.1 Introduction

ElikoPy, the novel pipeline developed during this Master’s thesis is a dedicated diffusion MRI (dMRI) toolbox created to correct all common known dMRI artifacts with a special focus on the adaptability of the pipeline to large-scale population studies. The implementation is based on Python and mainly utilize the Diffusion Imaging in Python (DiPy) package (Garyfallidis et al., 2014) but also rely on the Dmipy package (Fick et al., 2019) and MicrostructureFingerprinting (MF) (Rensonnet et al., 2019) package. The potential of the FSL software (Jenkinson et al., 2012), Freesurfer (Fischl, 2012), Advanced Normalization Tools (ANTs) (Avants et al., 2009), Itktools (Yushkevich et al., 2016), Mrtrix3 (Tournier et al., 2019) and Distribution of 3D Anisotropic Microstructural Environments in Diffusion Compartment Imaging (DIAMOND) (Scherrer et al., 2016) is also exploited. ElikoPy is fully open-source (AGPL3 License) and available publicly on Github (Simon and Dessain, 2021a). This pipeline supports as input the increasingly popular Brain Imaging Data Structure (BIDS) format (Gorgolewski et al., 2016) as well as classical Digital Imaging and Communications in Medicine (DICOM) and Neuroimaging Informatics Technology Initiative (Nifti) formats. Fig.2.2 illustrates the high level workflow present in ElikoPy.

Using a rich set of functions, data processed with ElikoPy are firstly audited to ensure that the image dimension match the dimension of the b-value and b-vector files along the acquisitions parameters and index files. If successful, a dedicated storage folder for each subject is generated. Afterward, preprocessing can be applied to correct and enhanced the raw image. This includes, brain extraction, reslicing, Principal Component Analysis (PCA) based denoising using the Marchenko-Pastur distribution, suppression in image space of Gibbs ringing artifacts, estimation and correction of the susceptibility induced field, modeling and correction of subject movements and finally, bias field correction. After the preprocessing, a white matter mask registered on diffusion data can be obtained using a T1 image or computed directly from diffusion data by segmenting an Anisotropic Power (AP) map (Dell’Acqua et al., 2014).

As seen in Fig.2.2, four distinct algorithms for the estimation of the microstructure are available. Subsequently, the pipeline adds the possibility to register the computed metrics in a common space given by either one of the study subjects or by an atlas. Finally, from there, ElikoPy can output statistical results for population studies by aligning metrics from multiple subjects into a common space and performing region wise, voxel wise and

cluster wise statistics.

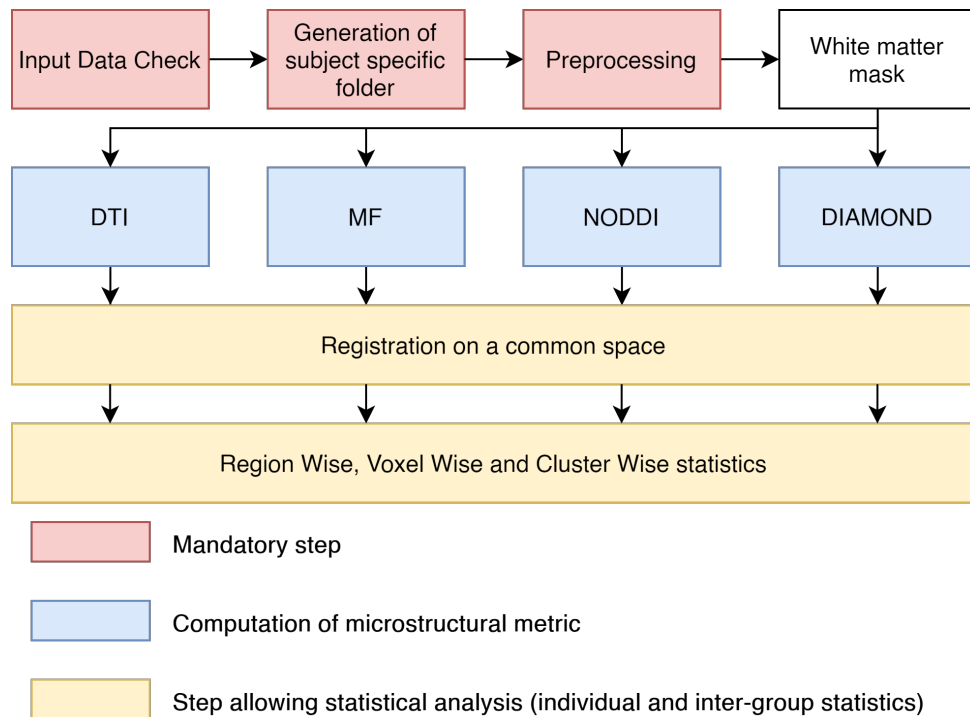


Figure 2.2: Overview of the main processing steps in the ElikoPy pipeline.

Several utility functions are also available in the library. Among them, there is :

- A data export function, which allows to obtain in a single folder specific metric for all subjects.
- A report export function which merges all the subjects reports into a single PDF file for all processing steps or for a specific stage.
- A function converting DICOM into NIfTI based on `dcm2niix` (Li et al., 2016).
- A function to quickly change the threshold value applied on the intra-cellular volume fraction (icvf) metric of Neurite Orientation Dispersion and Density Imaging (NODDI) without the needs of executing again the full NODDI core function.
- A wrapper function that applies a function given as an argument to every patients of the study. The wrapped function takes two arguments as input, the `patient_name` and the path to the root of the study.

The Elikopy toolbox follows a specific folder structure that prevents ambiguity and data losses when dealing with a large amount of subjects. Fig.2.3 illustrates the folder tree in ElikoPy.

The first type of folder present at the root of the Elikopy project are DATA folders. These folders contain all raw subject files belonging to the same class along with the associated acquisition parameters and index files. Classes are used to separate subjects that have different acquisition parameters or subjects that need to be separated from others groups of subjects. The pipeline does not have a limitation on the number of classes, the maximum number of subfolders in a folder as defined by the operating system is the actual hard limit.

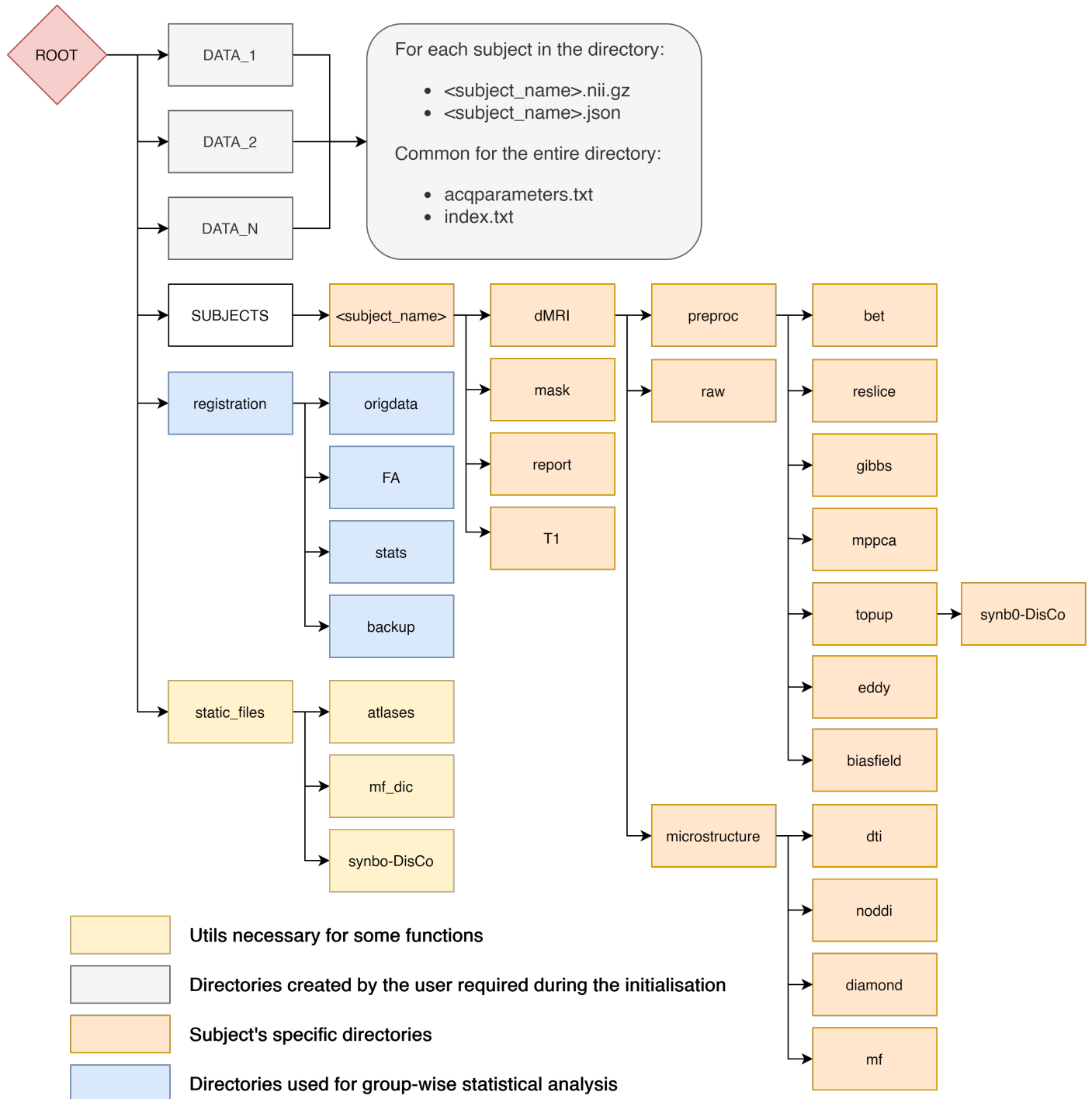


Figure 2.3: Overview of the folder structure used by the ElikoPy toolbox.

Alternatively, a path to a Brain Imaging Data Structure (BIDS) folder tree can be used to initialize the library instead of the set of DATA folders. If such a folder tree is used, ElikoPy will dynamically divide the subjects into different classes depending on their metadata (based on the acquisition parameters and the slice acquisition timing specification). ElikoPy will also identify the DW-MRI and structural T1 images to copy into the SUBJECTS folder based on the BIDS naming convention.

The SUBJECTS folder contains a subfolder for each valid subject presents in DATA folders. Along these subfolders, three json files are present. The subj_error.json file contains the list of subjects with invalid raw data, The subj_list.json file contains the list of valid subjects and the subj_type.json is a dictionary that maps each subjects subfolder to its data class.

Each subdirectory of the subjects folder contains the output of every preprocessing and processing function executed on the patient associated with the subdirectory. The output consists of NIfTI files, log files and some others files related to the specific functions.

The REGISTRATION folder contains diffusion metrics registered to a common space, group wise statistics and voxel wise statistics for each registered metric.

Finally, the STATIC_FILES folder contains files mandatory for some processing steps of the library such as MF dictionary and Synb0-DisCo atlases.

2.2.2 Preprocessing

The preprocessing stage aims at correcting the non idealities affecting the diffusion data before computing the diffusion metrics. Several correction steps are undertook as it can be seen in Fig.2.1. In this section, the different preprocessing algorithms are described. With the exception of skull stripping, all processing steps are optional and can be applied at the user discretion. They are summarized in Tab.2.1 along with their respective parameters.

2.2.2.1 Reslice

If the raw data is not in its 'native' resolution, a reslicing process is required. Usually, the MRI scanner performs automatic interpolation on the data in order to beautify the data since clinicians usually have a preference for high resolution images. However, the intrinsic resolution is not augmented by this interpolation. While somewhat useful to clinicians, the interpolation is usually not desirable for research. Using it means more computation time and uncorrelated noise becoming correlated which reduces the performances of MPPCA denoising algorithms. Moreover, interpolation is not desirable when performing Gibbs ringing correction (Kellner et al., 2016). Reslicing is therefore a way to mitigate the effect of mandatory interpolation during the acquisition. Furthermore, it allows to ensure isotropic voxels which facilitate most tractography algorithms.

It must be noted that reslicing is not always desirable when having non isotropic voxels. Increasing the voxel volume increase the SNR (see Eq. (1.52)) at the cost of reduced resolution along one (or more) given dimension(s). In spinal cord imaging where white matter tracts are quite well-aligned it is not uncommon to privilege in-plane (across the spinal cord) resolution to out-of-plane (along the spinal cord) resolution to get more signal (thus more SNR). For example, (Taquet et al., 2019) studied rats with a voxel resolution of $0.1 \times 0.1 \times 1mm^3$ to increase the SNR.

Reslicing is done using the align function of DiPy. The default interpolation method for resampling is the trilinear interpolation but can be set to any other interpolation order between zero (i.e. nearest neighbor interpolation) and five.

2.2.2.2 Brain extraction

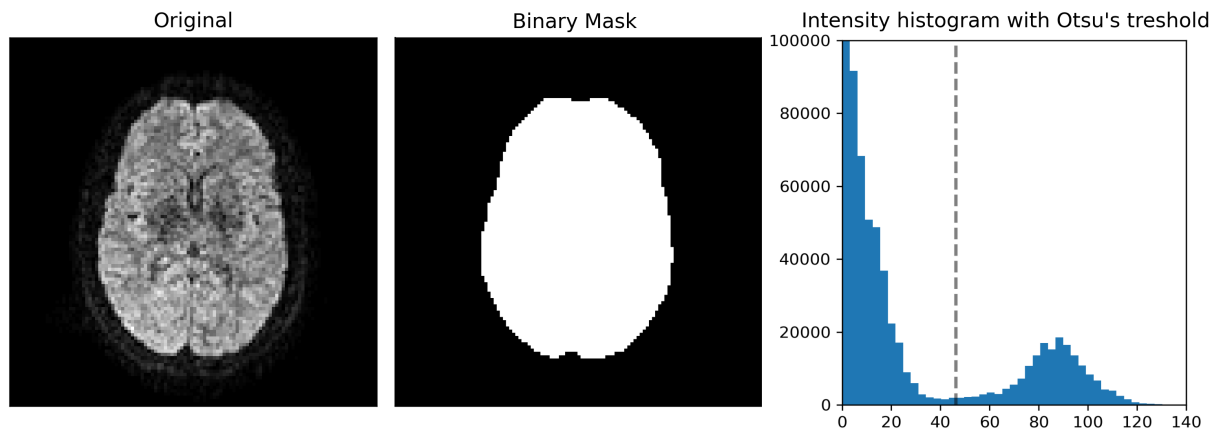


Figure 2.4: Original b0 images and binary mask obtained with using `median_otsu` are shown in the left and middle panels, while the thresholded histogram used by `median_otsu` is shown in the right panel.

The brain is extracted from the skull and other tissues surrounding the brain to increase the processing efficiency of subsequent steps. In addition to the gain of processing speed for several algorithms, removal of non brain tissues is a fundamental step in enabling the processing of diffusion images since brain regions must be skull-stripped before using other image processing algorithms (Kalavathi and Prasath, 2016). This includes, image registration and warping, inhomogeneity correction or tissue classification.

At the end of the preprocessing, a final brain mask readjusted in regard of all the applied preprocessing steps is also provided as output.

The mask is computed using `median_otsu` (see Fig.2.4) from DiPy. `Median_otsu` has been selected since the method is fast, robust, designed for DW-MRI images and directly available in Python. First, a non linear median filter is applied to reduce the noise and improve the brain segmentation. Second, the Otsu's method (Otsu, 1979) is used to separate the brain (foreground) from the background. Using the intensity histogram of the brain along multiple volumes, the algorithm returns an intensity threshold that separates the voxels between foreground and background. This threshold is computed by maximizing the inter-class variance. The skull gives rise to a weak signal in diffusion MRI making this technique possible, however it is not applicable on T1 images. It is noteworthy that the performance and quality of the image extraction tools depend on the quality of the MR images, therefore the first computed mask (before correction) is dilated using mathematical morphology.

2.2.2.3 MPPCA Denoising

To reduce Rician noise typically found in MR images, the input images are denoised using the Marchenko-Pastur PCA technique as implemented in DiPy. Since the noise in

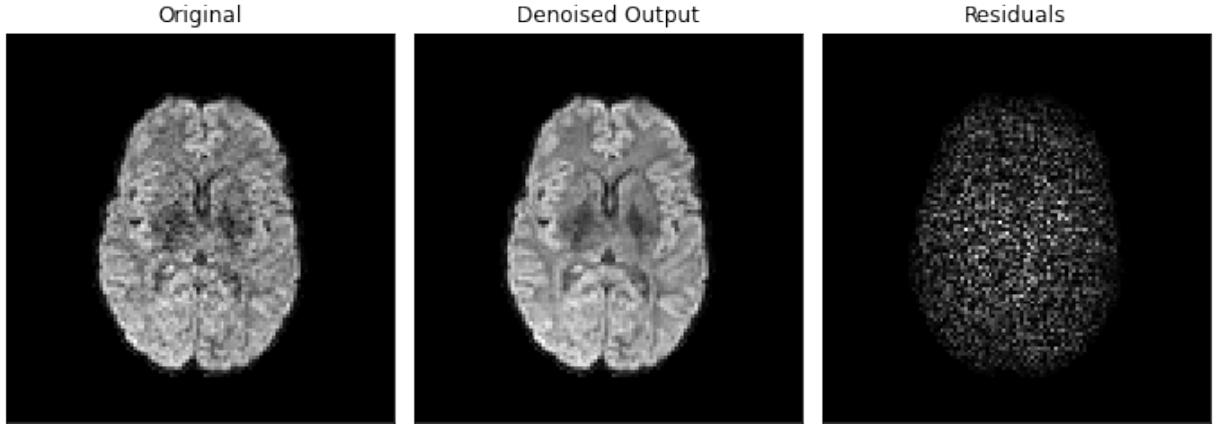


Figure 2.5: Original and denoised b0 images are shown in the left and middle panels, while the difference between these images is shown in the right panel. An unstructured spatial distribution of the right image indicates extraction of random thermal noise.

diffusion data is spatially dependent in the case of multichannel receive coils (Aja-Fernández et al., 2013), Principal component analysis of Marchenko-Pastur (MPPCA) noise-only distribution provides an accurate and fast method of noise evaluation and reduction (Veraart et al., 2016). This methods has been chosen since it is a fast denoising algorithm that does not blur the image or create artifact. As demonstrated in (Veraart et al., 2016), MPPCA provides better results than two conventional state of the art denoising techniques for DW-MRI images : adaptive non local means (Manjón et al., 2013) (ANLM) and second order total generalized variation (Knoll et al., 2011) (TGV) which both create image blurring and present anatomical structures in their residual maps.

It is vital that the denoising step is performed at the primary stage of the pipeline since it depends on the noise being uncorrelated both spatially and among progressive acquisitions. Performing this step after a step that utilize interpolation to recreate pictures would result in correlated noise and a failure of the assumptions that MPPCA removes only thermal noise without modifying anatomical features. This assumption is performed under the random matrix theory-based approach to PCA denoising.

The denoising process relies on Principal Component analysis of a $M \times N$ matrix X , with rows representing M voxels within a sliding window, and columns representing N weighted magnitude MR measurements in each voxel. This matrix is assumed large (i.e. $R = \min(M, N) \gg 1$).

$$\Sigma_X = \frac{1}{N}XX^T = U\Lambda^2U^T \quad (2.1)$$

Σ_x denotes the covariance matrix of X , U is an orthonormal matrix which allows to obtain uncorrelated data, Λ is the matrix of eigenvalues of Σ_x and is the correlation matrix in the uncorrelated data space.

The dMRI data is highly redundant due to correlations between the diffusion weighted signals and can be seen as a linear combination of a few $P \ll R$ linearly independant sources. Therefore, in a noise-free situation, PCA would output only P non zero eigenvalues.

In practice, there is no noise free data. The noise propagates through PCA and all R eigenvalues are non zero (i.e. the matrix Σ is of rank R). However, while the noise is random, its contribution to the histogram of Σ eigenvalues becomes deterministic **in the limit** $P \ll R$. The part of PCA eigenvalues due to Gaussian noise with variance σ^2 and

zero mean is given by the Marchenko-Pastur distribution :

$$p(\lambda|\sigma, \gamma) = \begin{cases} \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{2\pi\gamma\lambda\sigma^2} & , \lambda_- \leq \lambda \leq \lambda_+ \\ 0 & , \text{Otherwise} \end{cases} \quad (2.2)$$

where $\lambda_{\pm} = \sigma^2(1 \pm \sqrt{\gamma})^2$ with $\gamma = M/N$ the matrix size ratio.

The overall eigenvalues histogram that can be obtained is represented in Fig.2.6. By using the fact that the noise contribution is bounded by $\lambda_{\pm}$ and that noise-only eigenvalues are expected to obey the universal Marchenko-Pastur law, it is possible to zero out the noise component and to effectively **denoise** the data after PCA reconstruction by omitting the corresponding eigenvalues. It is to note that depending on the noise level, some of the P significant eigenvalues can fall into the MP bulk and become indistinguishable from noise (Veraart et al., 2016).

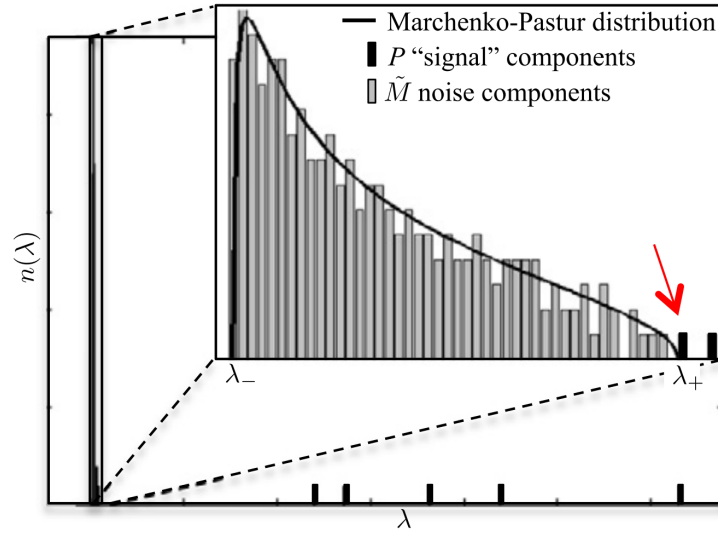


Figure 2.6: Histogram of eigenvalues after Principal Component Analysis. The noise and signal eigenvalues are represented respectively in gray and black. The Marchenko-Pastur distribution is represented on top of the noise component, from (Veraart et al., 2016).

As the MP distribution is parameterized by the variance σ and the matrix ratio γ , the noise level is estimated by the algorithm by fitting iteratively the MP distribution to the eigenvalue histogram for the lowest eigenvalues. Practically, the noise is also not Gaussian but Rician (i.e. module of a complex gaussian). It thus follows a non zero mean χ distribution with L degrees of freedom. To correct for this bias in the estimation, the inversion technique described in (Koay and Basser, 2006) is applied.

Assuming the $\tilde{M}$ lowest eigenvalues have been nullified, the noise variance after denoising can be estimated using PCA theory

$$\tilde{\sigma}^2 = \sigma^2 - P_{\sigma} = \sigma^2 - \sum_{i=P+1}^M \frac{\lambda_i}{M} \approx (1 - \frac{\tilde{M}}{M})\sigma^2 = \frac{P}{M}\sigma^2. \quad (2.3)$$

The SNR after denoising thus scales in $\sqrt{M/P}$. The value of P is not known a priori and will depend on the redundancy of the data.

	Parameters	Notes
Reslice	Voxel size	Target size for the voxels of the output
Brain extraction	Median filter radius	Size of the median filter applied before Otsu
	Median filter number of pass	Number of time the median filter is applied
	Dilate	The size of the kernel to dilate the final mask with
MPPCA	Patch radius	Radius of the local patch to be taken around each voxel, sliding window size
Gibbs ringing correction	/	(non parametric method)
Susceptibility field estimation	Topup config	The path to a Topup config file containing some or all of the parameters that can be specified for Topup. The predefined config file <i>b02b0.cnf</i> is used by default
	Force Synb0-DisCo	Force the estimation of the susceptibility field from a T1
Motion and EC induced distortions correction	Acquisition parameters	PE direction and bandwidth in this direction for each volume
	Eddy model	Order of the model to use for modeling Eddy currents
	Number of iterations	Specify the number of iterations of the Eddy algorithm
	Interpolation method	Interpolation method to use during the estimation
	Number of std	Number of standard deviation away a slice must be to be considered an outlier
	Mporder	Temporal order of the DCT model used to model slice-to-volume motion
	Regularization	Strength of the temporal regularization for the slice-to-volume motion
	Slice specifications	Specification on the slice acquisition order of the volumes
Bias field correction	B-spline fitting	Provides the initial mesh resolution in mm and the spline order. This value is optimized for human adult data and needs to be adjusted for other data
	Convergence	Define the maximum number of iteration and the convergences threshold

Table 2.1: Summary of the processing methods applied during the preprocessing and their optional parameters.

2.2.2.4 Gibbs ringing correction

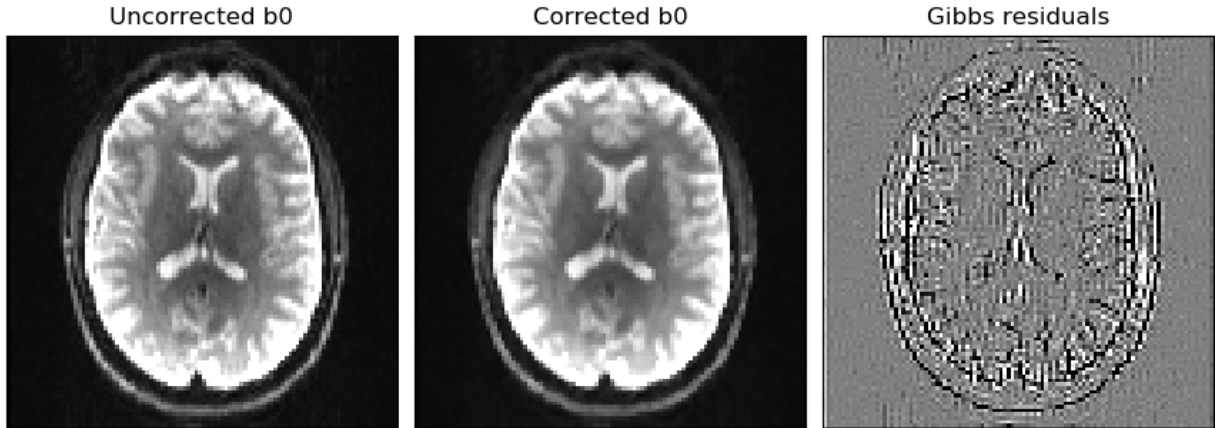


Figure 2.7: Gibbs ringing correction, uncorrected and b0 images corrected for Gibbs ringing are shown in the left and middle panels, while the difference between these images is shown in the right panel. Gibbs ringing artifacts typically occur at interfaces with sharp changes in intensity.

In general, in the context of diffusion-weighted imaging, derived diffusion-based estimates are greatly affected by Gibbs oscillations (Perrone et al., 2015). To correct for this, `gibbs_removal` from DiPy is used. This algorithm models the truncation of k-space as a convolution with a sinc-function in the image space. The severity of ringing artifacts thus depends on how the sampling of the sinc function occurs. The `gibbs_removal` function reinterpolate the image based on local, subvoxel-shifts to sample the ringing pattern at the zero-crossings of the oscillating sinc-function (Kellner et al., 2016).

The image is shifted using the Fourier shift theorem :

$$\mathbf{I}(a - a_0) = FT^{-1}[\mathbf{H}(k_a)e^{-\frac{2\pi j k_a a_0}{N_a}}] \quad (2.4)$$

The optimal resampling shift is obtained by testing multiple shifts and assessing which value minimizes the total variation for a positive (a_p) and negative (a_n) displacement. The image is then reconstructed using linear interpolation for the two shifts.

$$\mathbf{I}(a) = \frac{\mathbf{I}(a - a_n) - \mathbf{I}(a - a_p)}{a_p - a_n}a_n + \mathbf{I}(a - a_n) \quad (2.5)$$

The computation is made slice by slice separately along the x and y axes to produce the corrected $\mathbf{I}_x$ and $\mathbf{I}_y$. The final image $\mathbf{I}_c$ is computed as :

$$\mathbf{I}_c = FT^{-1}[FT(\mathbf{I}_x)\mathbf{G}_x + FT(\mathbf{I}_y)\mathbf{G}_y], \quad (2.6)$$

where $\mathbf{G}_x$ and $\mathbf{G}_y$ are weights computed with the expressions

$$\mathbf{G}_x = \frac{1 + \cos(k_y)}{2 + \cos(k_y) + \cos(k_x)} \quad \mathbf{G}_y = \frac{1 + \cos(k_x)}{2 + \cos(k_x) + \cos(k_y)}. \quad (2.7)$$

2.2.2.5 Motion, Eddy current and susceptibility distortion correction

Motion, susceptibility and Eddy current induced distortions are artifacts with different origins but a similar effect i.e the displacement and deformation of the brain. They can therefore be jointly corrected. This is achieved by first estimating the susceptibility field using FSL Topup and then using FSL Eddy to correct for motion and the two distortion fields (susceptibility and Eddy Current (EC) induced). The fields are incorporated in a way that reflects their comportment with movement. The method is based on registering the individual volumes to a model free prediction of what each volume should look like (Andersson and Sotiropoulos, 2016).

Susceptibility field estimation : As explained in chapter 1.4, an off-resonance field is induced by differences in magnetic susceptibility near junctions of tissues of differing magnetic susceptibility. This field causes geometric distortions in the data when images are acquired with the spin-echo (SE) echo-planar imaging (EPI) sequence typically used in DW-MRI. In order to correct from these distortions, the field needs to be estimated. The susceptibility estimation later used by Eddy to correct images distortion from susceptibility induced local gradients is done in ElikoPy using Topup (Andersson et al., 2003).

Topup aims to estimate the susceptibility field causing these distortions using two opposed phase encoded images for each diffusion gradient, one with a bottom-up and one with a top-down traversal of k-space in the phase-encode direction. These two images have an identical magnitude of distortions in opposing directions (Jezzard and Balaban, 1995). Alternatively, if only one encoding direction is available, a synthetic reversal direction of encoding can be generated from a T1 using a Synthesized b0 for diffusion Distortion Correction (Synb0-DisCo) (Schilling et al., 2019).

In order to correct for EPI distortions, most of the state of the art methods requires sequence modification or acquisition of additional data. These methods include

- b0 mapping (Jezzard and Balaban, 1995) which characterize the field inhomogeneities from a field map,
- point spread function estimation (Zaitsev et al., 2004) which extends the field map approach in a more robust way by relying on the application of an additional gradient encoding to the EPI readout,
- registration of a distorted diffusion images to an undistorted anatomical target (Tao et al., 2009) (Wu et al., 2008),
- reverse gradient acquisitions (Andersson and Skare, 2010).

All these approach suffers from drawback that limits their practical usability. Motion during acquisition and artifacts can corrupt the field map of the b0 inhomogeneity which can cause the geometric correction based on the field map to actually decrease the image quality. Along these limitations, point spread function estimation relies on empirical estimation of the image properties. Correction based on registration can have difficulties to handle the very different contrasts between DW-MRI images and the standard structural scans to which they are registered. The reverse gradient acquisitions strategy were initially sensitive to noise and motion artifacts since these strategy did not consider that the underlying field should be continuous and smooth. The model utilized by Topup improves this framework by modeling the field as a linear combination of smooth basis functions and by including subject motion into the correction.

Using the two volumes acquired with a blip-up and blip-down acquisitions and the

acquisition parameters, the susceptibility field is estimated by Topup by finding the field that, when applied to the two volumes, maximizes the similarity of the unwrapped volumes. The similarity criterion is assessed by the sum of squared differences between the unwrapped images. Using Gauss-Newton, Topup jointly finds the susceptibility field and movement that may have occurred between the two acquisitions.

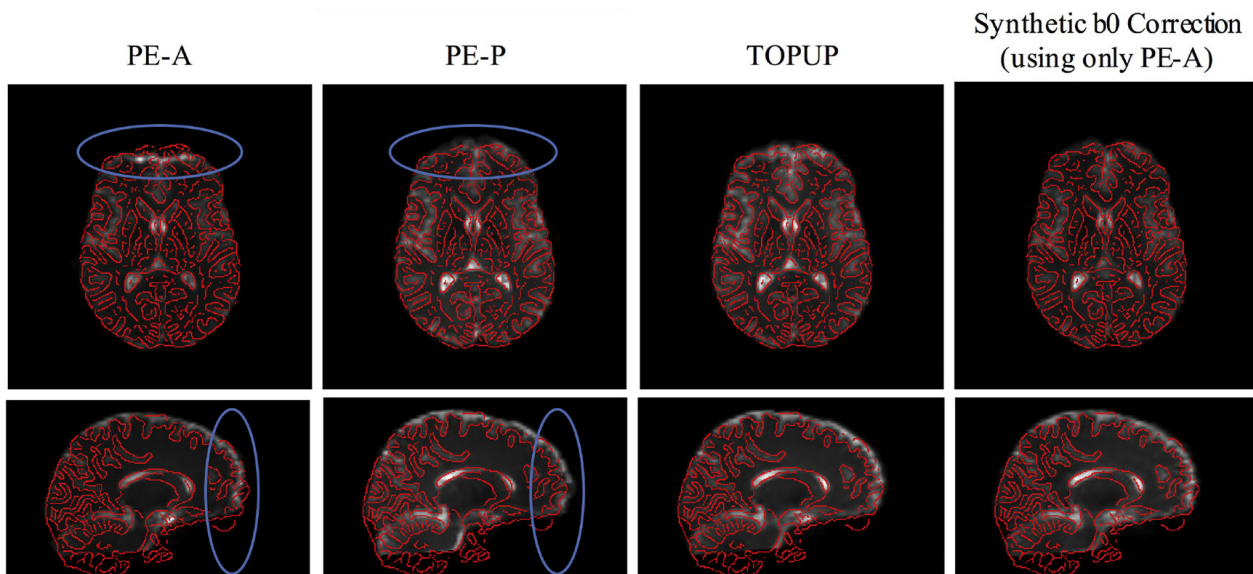


Figure 2.8: Synthetic b0 correction results in reasonable distortion corrections, qualitatively similar to state of the art (Topup). Axial and Sagittal slices of the volumes with phase encoding in the anterior and posterior directions, the Topup corrected image, and that corrected using the synthetic b0 image are shown for both b0 image, shown in T1 space. From (Schilling et al., 2019).

When only one encoding direction is available, the state of the art susceptibility and motion correction capabilities of Topup cannot be directly exploited. Fortunately, Synb0-DisCo can be used to generate an undistorted non diffusion weighted image synthesized from an anatomical scan. Fig.2.8 illustrates the differences between the original images in anterior phase encoding directions and posterior phase encoding directions along Topup and SynB0-DisCo correction.

The method used to synthesize an undistorted EPI image with ElikoPy is the following :

1. The T1 weighed structural image is bias field corrected and normalized using Ants (Avants et al., 2009). BET (Jenkinson et al., 2005) is then used to skull strip the T1.
2. A skull stripped distorted b0 is extracted from the diffusion 4D images and registered on the skull stripped T1 image.
3. Using Ants, the original T1 is registered into an atlas space in three stages. The first stage consist of a rigid transform, the second stage consist of an affine transform and the last stage consist of a deformable transform.
4. The linear and non linear transform resulting from the preceding step is applied on the normalized T1 and the registered distorted b0.
5. A Generative Adversarial Network (GAN) trained following the modalities presented in Fig.2.9 is used to generate an undistorted b0 in the atlas space.
6. The three different orientations generated by the GAN are combined by median filtering, resulting in a smoothed synthesized b0 image.

7. The inverse transform is applied to the undistorted b0 to put it back from the atlas space to the diffusion space.
8. The original distorted b0 is smoothed to match the smoothing of the synthesized b0. The undistorted b0 is then merged with the smoothed distorted b0.
9. The susceptibility field is estimated using Topup with the 4D images containing all b0. The “synthesized” b0 images is defined as having an infinite bandwidth in the PE direction to only allows the warping of the real b0 into the synthesized one. Correcting the geometry of the real b0 using the structure of the undistorted synthesized image.

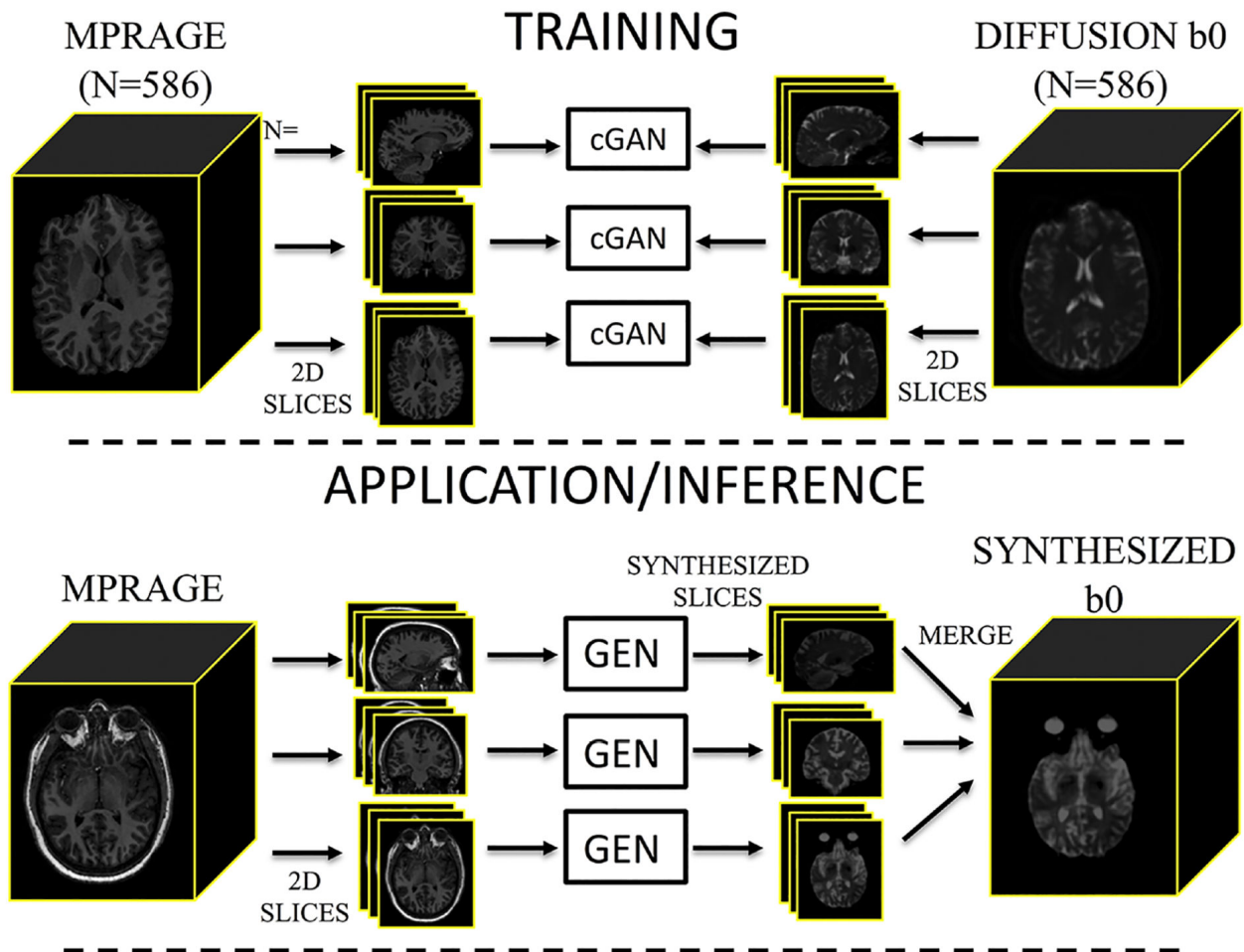


Figure 2.9: Image synthesis training (top) and application of the network (bottom). Paired MPRAGE and diffusion b0 images are sliced into groups of orthogonal slices to act as input to three independent conditional GANs (cGAN). Upon application of the network, an MPRAGE is used to synthesize an undistorted b0 contrast. The MPRAGE is sliced into orthogonal stacks, input into the trained networks (the generator – GEN) which synthesize orthogonal slices, and merged into the synthesized b0. From (Schilling et al., 2019).

Eddy FSL corrections : Once the susceptibility field is known, to correct for the artifacts, Eddy FSL is used. It works by making a prediction about how each diffusion volume should look. Then, it compares the prediction to the observed data and uses the difference to update the estimation on the subject position and EC-induced field. The

prediction is calculated using a Gaussian Process (GP) for which the hyperparameters are estimated directly from the data (Andersson and Sotiropoulos, 2016). To simplify the discussion, the algorithm is first only explained for volume-to-volume motion, it is later generalized to account for slice-to-volume motion.

The following notations are used :

- s : reference/undistorted space.
- f_i : observed/distorted space for the i^{th} volume of the data.
- $\mathbf{a}_i$: acquisition parameters for the i^{th} volume of the data. It is a vector of size 4 which contains the PE-direction and bandwidth in the PE-direction $\mathbf{a} = [\mathbf{p}, t]$
- $\mathbf{r}_i$: rigid body motion parameters for the i^{th} volume (3 translation and 3 rotation parameters).
- $e(\beta_i)$: Eddy current induced off resonance field (Hz) for the i^{th} volume, computed from the Eddy current parameters β . The EC-induced field model can be linear, quadratic or cubic e.g. for the linear model $\beta = [x, y, z, \text{off}]$ the first three elements specify linear gradients (Hz/mm) in the x y and z directions with the zero crossing at the center of the volume, the fourth element is the offset (Hz).
- $\omega(h, \mathbf{r}_i)$: Field (Hz) obtained by rigid body transform of the Topup estimated Susceptibility induced off resonance field h .
- $\mathbf{d}(\psi, \mathbf{a})$: Voxel displacement field given a field ψ in Hz and the acquisition parameters $\mathbf{a} = [\mathbf{p}, t]$. At a given voxel $\mathbf{x}$ the displacement is given by

$$\mathbf{d}_x = [p_1 t \psi_x, p_2 t \psi_x, p_2 t \psi_x]^T. \quad (2.8)$$

The overall Eddy algorithm is summarized in Fig.2.10. It is an iterative process made of two steps : (1) prediction (2) estimation

(1) Estimation step : During the estimation step the image is transformed from the observed space to the estimated undistorted space using the evaluated motion, susceptibility and Eddy parameters (initialized to zero for the first iteration). If $\mathbf{x}$ is a coordinate in the reference space s and $\mathbf{x}'$ in the observed space f_i :

$$\mathbf{x}' = \mathbf{P}(\mathbf{r}_i)^{-1} \mathbf{x} + \mathbf{d}_x(h + \omega(e(\beta_i), \mathbf{r}_i), \mathbf{a}_i) \quad (2.9)$$

$\mathbf{P}(\mathbf{r}_i)$ is used to represent in a compact way the rigid body transformation. In homogeneous coordinate, the mapping between two coordinates for a rigid transform is given by :

$$\begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix} = \underbrace{\mathbf{M}^{-1} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix}^{-1} \mathbf{M}}_{\mathbf{P}(\mathbf{r})} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \quad (2.10)$$

where $\mathbf{t}$ is a vector with the first three elements of $\mathbf{r}$ and $\mathbf{R} = \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z$ is the three dimensional rotation matrix computed using the last three elements of $\mathbf{r}$.

Each volume is associated with a set of parameters and undergoes the mapping. The mapping is then used to resample the image using cubic spline interpolation to obtain the estimated undistorted volumes $\hat{s}_i$.

The estimation is afterward loaded as training data for Gaussian Process. GP is a model that assumes that the signal varies smoothly, as a function of b-value and diffusion gradient direction, and uses linear combinations of the observed data to make predictions. The smoothness is determined from hyperparameters whose values are determined directly

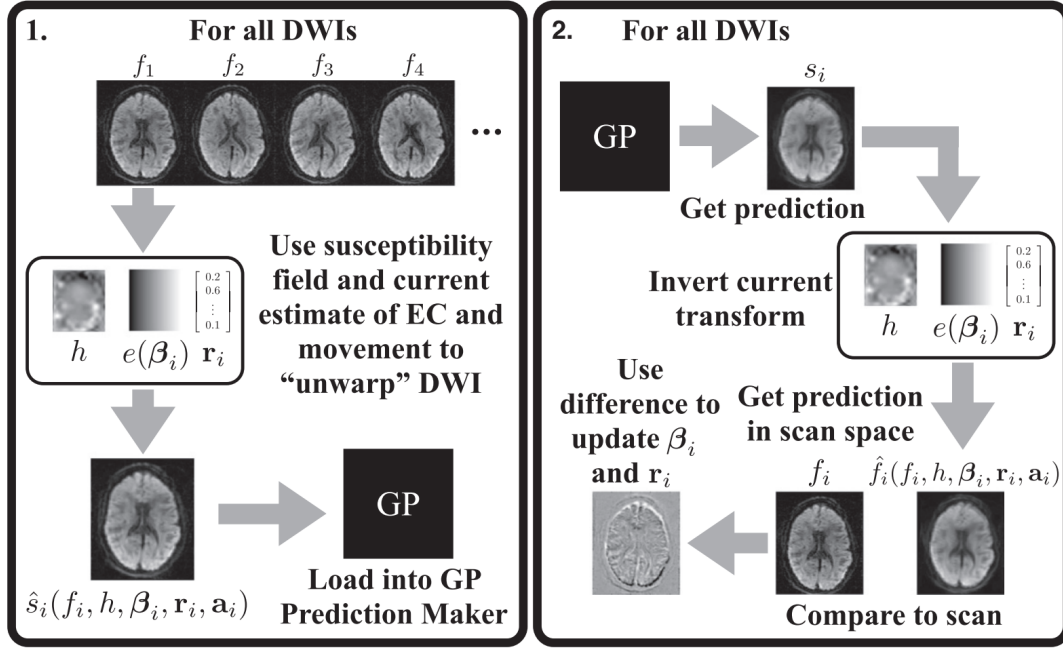


Figure 2.10: Correction algorithm followed by Eddy of FSL. It is an iterative process that consists of a (1) prediction and (2) estimation step, (Andersson and Sotiropoulos, 2016)

from the data (Andersson and Sotiropoulos, 2015). Intuitively, if one takes two volumes with opposite gradient directions, the EC induced distortions will be opposite. The GP makes the average of both those gradient images resulting in an overall prediction closer to the true undistorted space than the current estimate of the corrected observed image.

(2) Prediction step : After step (1), using the Gaussian Process, a prediction of the undistorted space is obtained. This prediction is projected back into the observed space using the inverse transform.

$$\mathbf{x}' = \mathbf{P}(\mathbf{r}_i)(\mathbf{x} + \mathbf{d}_x(\omega(h, \mathbf{r}_i) + e(\beta_i), \mathbf{a}_i)^{-1}) \quad (2.11)$$

After interpolation, it results in predictions in the observed scan space $\hat{f}_i$.

Finally, The difference between the prediction in the observed space and the actual data is computed and used to update the EC and motion parameters. The cost used for the update is the sum-of-squared differences and each update is a non linear optimization Gauss–Newton step.

$$\begin{bmatrix} \beta_i^{k+1} \\ \mathbf{r}_i^{k+1} \end{bmatrix} = \begin{bmatrix} \beta_i^k \\ \mathbf{r}_i^k \end{bmatrix} - (\mathbf{D}\mathbf{D}^T)^{-1} \mathbf{D}^T(\hat{f}_i - f_i), \quad (2.12)$$

with $\mathbf{D}$ the matrix of partial derivatives.

The two Eddy steps are run iteratively, pushing the estimation towards an undistorted image for motion, susceptibility and EC induced displacements. The algorithm is run for a fixed number of iteration and does not check for convergence since the same model is not fit at every iteration.

Extension for slice-to-volume motion : Up to this point only the volume-to-volume motion has been considered. Several modifications are required in order to account for the slice-to-volume movement (Andersson et al., 2017).

Previously each volume had a set of rigid body motion parameters $\mathbf{r} = [\Delta_x, \Delta_y, \Delta_z, \phi_x, \phi_y, \phi_z]$ from which the transformation $\mathbf{P}(\mathbf{r})$ is computed and applied on all voxels of the volume. Now a set of $\mathbf{r}(t)$ parameters are required for each volume. To ensure the continuity and differentiability of the motion, it is modeled with a Discrete Cosine Transform (DCT) and regularized during the update with a parameter λ .

$$\mathbf{r}(t_n) = \begin{bmatrix} 1 & \cos(\frac{n\pi}{N-1}) & \dots & \cos(\frac{Mn\pi}{N-1}) \end{bmatrix} \begin{bmatrix} \Delta_x(0) & \Delta_y(0) & \dots & \phi_z(0) \\ \Delta_x(1) & \Delta_y(1) & \dots & \phi_z(1) \\ \dots & \dots & \dots & \dots \\ \Delta_x(M) & \Delta_y(M) & \dots & \phi_z(M) \end{bmatrix} \quad (2.13)$$

M is the order of the DCT expansion, it characterizes the degree of freedom that is let to the intra-volume motion, decreasing M results in a smoother modelization of the motion. N is the number of time steps, it is assumed the slices are acquired in discrete time points.

The model of $\mathbf{r}(t_n)$ is combined with a function $Z(t_n)$ which output all the slices indices acquired at time t_n . For example, if 15 slices are acquired in an interleaved manner with a multi-band (mb) factor of 3, there will be $15/3 = 5$ time steps and the function Z will output : $Z(0) = 0, 5, 10 \dots Z(5) = 3, 8, 13$. This function $Z(t_n)$ must be provided by the user as it depends on the acquisition.

Overall, the motion is computed as follows : for each volume and for each time step $t_n = \frac{\#slices}{\#mb-factor}$, the rigid body transform $\mathbf{P}(\mathbf{r}(t_n))$ is applied to all slices s in $Z(t_n)$.

Using this, equation 2.11 can be adapted directly by simply applying the transform on each slice. However, for the forward spatial transform (equation 2.9), it gets more complicated (i.e. to go from the distorted space to the undistorted one). Indeed, due to motion the samples are not on a regular grid and it might happen that no data has been acquired in some portion of the undistorted brain complicating its reconstruction, there is no one-to-one correspondence. This is illustrated on Fig.2.11.

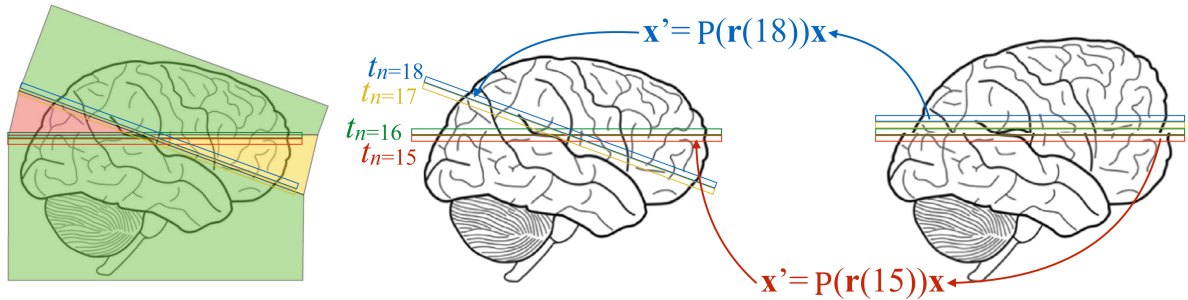


Figure 2.11: Illustration of the problems encountered when trying to interpolate from the observed space (right) to the undistorted space (left). It has been assumed on this representation that the patient has rotated his head between the acquisition of the slice 16 and 17. However, it is not known and the slices in the observed space will still be stacked up normally giving rise to the distorted brain (right). Due the motion, the green part of the volume has been acquired once, on a regular grid and standard interpolation would be feasible. Yet, the yellow part has been acquired twice and it might happen that voxel values from two different slices coincide spatially. The red part of the brain has not been acquired at all and the interpolation needs to fill this region (Andersson et al., 2017).

Therefore, to resample the irregularly acquired slices on a regular grid, an hybrid 2D+1D resampling strategy is applied. The idea is that instead of making the mapping on a

regular grid for the target space, a correspondence is made for each xy-position of the undistorted space and each axial slice of the distorted space. This results in a target space which is irregularly sampled in the z-direction. This is illustrated on Fig.2.12. The data is then spline interpolated onto a regular xy-grid in the target space, followed by an irregular 1D spline resampling along the resulting z-columns.

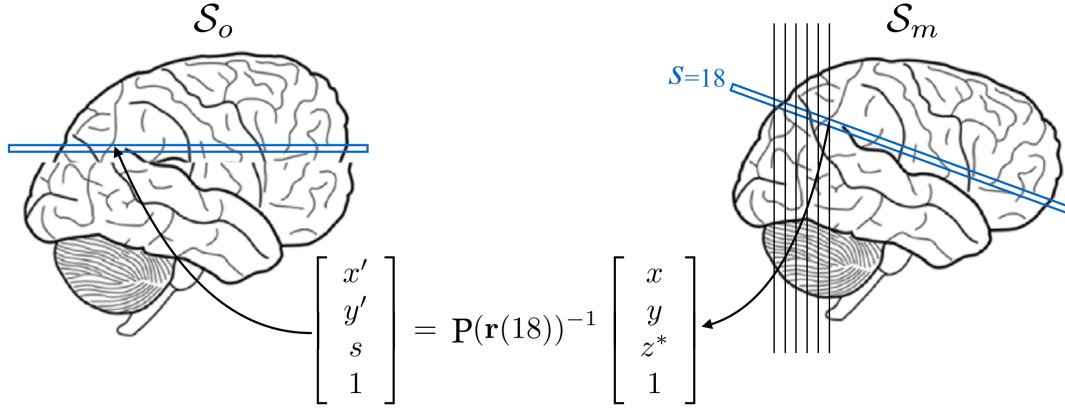


Figure 2.12: S_o and S_m denotes respectively the image in the observed/distorted and target/undistorted spaces. In order to reconstruct the volume S_m from S_o , for each slice s in S_o , the $x' y'$ position is found corresponding to the regular coordinates $x y$ of S_m . The non integer coordinate z^* in S_m corresponding to the $x' y' s$ combination is calculated. This results in the equation shown on the figure (EC and susceptibility induced distortions have been neglected) with unknowns $x' y'$ and z^* on both sides. The volume S_m obtained is sampled on a regular xy grid but irregularly along the z-direction (Andersson et al., 2017).

As explained above (see Fig.2.11), due to motion, there are gaps in the data and the spline interpolation may fail to recover the content of the gaps. To cope with this issue, missing data is therefore replaced by the predictions from the Gaussian Process. These predictions are weighted such that whenever there are observations, they always take precedence. This process is shown on Fig.2.13.

Outlier replacement for signal dropout : Due to motion during the diffusion encoding part of the imaging sequence, signal dropout arises. It is characterized by a drop of intensity for a slice or a group of slices and leads to a set of unusable measurements.

Using the framework of Eddy FSL, it is possible to correct the signal dropout. The idea is to detect the outlier slices by comparing each of them with what it “should” look like using the Gaussian Process prediction. During each iteration of the Eddy algorithm, the list of outlier slices is updated and the corresponding slices are replaced by their GP prediction (Andersson et al., 2016).

For each volume g and slice s the following value is computed :

$$d_{gs} = \frac{1}{n_s} \sum_{i \in s} y_{gi} - \hat{y}_{gi}, \quad (2.14)$$

with n_s the number of brain voxels in s , y_{gi} the intensity of the voxel i of the volume g and $\hat{y}_{gi}$ the predicted value of the Gaussian Process.

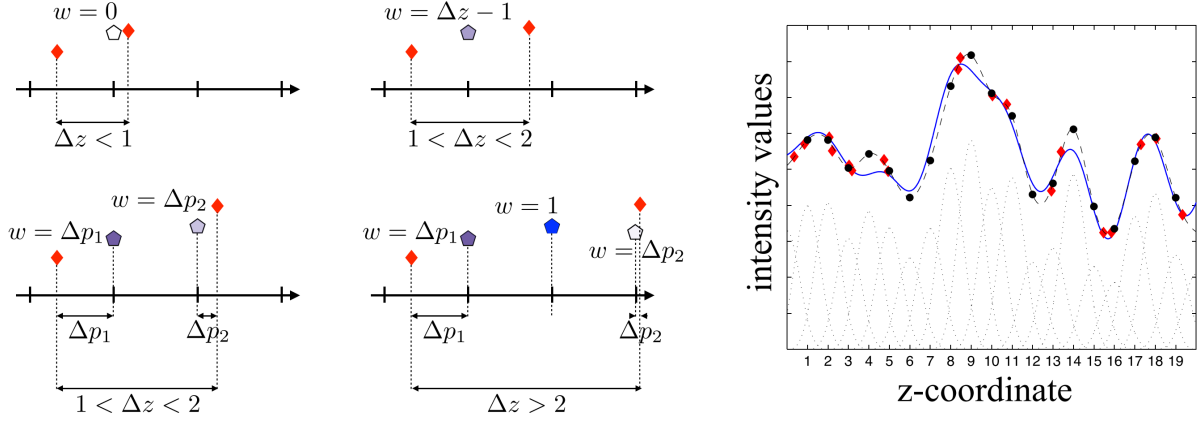


Figure 2.13: Interpolation process along the z -direction. The red diamonds are the data points, the black points are the interpolated points and the pentagones are the Gaussian Process predictions. These predictions are weighted depending on the proximity with the data as shown on the figure (Andersson et al., 2017).

Using all the d_{gs} values, the mean $\bar{d}$ and the standard deviation σ_d^2 are computed. Then, each d_{gs} value is converted to a z -score.

$$z_{gs} = \frac{\sqrt{n_s}(d_{gs} - \bar{d})}{\sigma_d} \quad (2.15)$$

The slice is considered to be an outlier if its z_{gs} value exceeds a certain user determined threshold (e.g. 3 times the standard deviation). Slices considered outliers are replaced by their GP prediction $\hat{y}_{gs}$.

2.2.2.6 Bias Field Correction

As explained in chapter 1.4, inhomogeneity of the b_0 magnetic field inside the MR scanner causes a spatial bias in the signal intensity of MR image data. This variability of the signal in tissues of the same type can affect microstructural metrics computation and brain segmentation algorithms (Zhang et al., 2001).

The artifact coming from the bias field can be corrected using the N4 Bias Field Correction algorithm (Tustison et al., 2010) which corrects low frequency intensity non uniformities present in MRI image. This algorithm is a nonparametric non uniform intensity normalization method which is independent of pulse sequence and insensitive to pathological data that might otherwise violate model assumptions. This package is available in ITK and Ants. The Ants version is integrated in the ElikoPy package. N4 is used in ElikoPy since it is the only known algorithm capable of removing the bias field of dMRI from b_0 volumes and non b_0 volumes.

The Bias Field Correction operated by N4 is a multiresolution iterative algorithm which seeks the smooth multiplicative field that maximizes the high frequency content of the distribution of tissue intensity. To obtain the corrected image, N4 use the image formation

$$v(\mathbf{x}) = u(\mathbf{x})f(\mathbf{x}) + n(\mathbf{x}),$$

where v is the source image, u is the uncorrupted image, f is the bias field, and n is the noise (assumed to be independent and Gaussian). Using the notation $\hat{u} = \log u$ and

assuming a noise free scenario, the image model becomes

$$\hat{v}(\mathbf{x}) = \hat{u}(\mathbf{x}) + \hat{f}(\mathbf{x}).$$

Using this model, an iterative solution can be derived in order to obtain the corrected image at iteration n .

$$\begin{aligned} \hat{u}^n &= \hat{u}^{n-1} - \hat{f}_r^n \\ &= \hat{u}^{n-1} - S \{ \hat{u}^{n-1} - E[\hat{u} | u^{n-1}] \}, \end{aligned} \quad (2.16)$$

where $\hat{u}^0 = \hat{v}$, $\hat{f}_r^n$ is the residual field estimate at the n th iteration, the smoothing operator $S\{\cdot\}$, is a uniform B-Spline approximation and $\hat{f}_e^n = \sum_{i=1}^n \hat{f}_r^i$, the total bias held estimate at the n th iteration is the sum of the first n residual bias fields.

In addition to the iterative algorithm, N4 uses a multi-resolution framework since direct fitting using a higher mesh resolution might miss the lower frequency components of the underlying bias field. A synthetic view of the complete algorithm can be visualized in Fig.2.14

```

1:  $\hat{\mathcal{L}}(p) \leftarrow \mathbf{0}$  {initialize control point lattice}
2:  $n \leftarrow 1$ 
3: for each resolution level do
4:   for each iteration at current resolution level do
5:      $\hat{f}_r^n \leftarrow S^* \{ \hat{u}^{n-1} - E[\hat{u} | \hat{u}^{n-1}] \}$  {calculate residual field}
6:      $\hat{u}^n \leftarrow \hat{u}^{n-1} - \hat{f}_r^n$  {update the estimate of  $\hat{u}$ }
7:      $\hat{f}_e(p) \leftarrow \hat{f}_e(p) + \hat{f}_r^n(p)$  {add residual control point lattice to the total estimate}
8:    $n \leftarrow n + 1$ 
9: end for
10:  $\hat{\mathcal{L}}(p) \leftarrow \hat{\mathcal{L}}(p)'$  {refine control point lattice}
11: end for

```

Figure 2.14: N4 iterative algorithm pseudocode, from (Tustison et al., 2010).

The estimate of the field $E[\hat{u} | u^{n-1}]$ is based on the assumption that the corruption of the low frequency bias field can be modeled as a convolution of the intensity histogram by a Gaussian.

2.2.3 Extraction of white matter mask

Providing a white matter mask is a useful step to accelerate microstructural features computation and more easily do tractography. The `white_mask` function of the ElikoPy library has been elaborated to perform this important step.

On the one hand, when a T1 image is available, a white matter mask can be computed from this data. Therefore, the T1 image is first preprocessed then segmented. Finally the

segmented white matter mask is projected into the space of the preprocessed diffusion image.

On the other hand, when no T1 images are available, the white matter mask is directly computed from a segmentation of the diffusion data using Anisotropic Power (AP) map (Dell'Acqua et al., 2014). In this case, no registrations are necessary.

T1 preprocessing : The structural T1 image is first corrected of Gibbs artifact. The method used (Kellner et al., 2016) is the same as the one used for Gibbs ringing correction for diffusion-weighted imaging described in section 2.2.2.4. The second step of the preprocessing is skull stripping using BET from FSL. As seen in Fig.2.15, the underlying algorithm relies on premasking through registration using FMRIB's Linear Image Registration Tool (FLIRT), bias field correction using FMRIB's Automated Segmentation Tool (FAST) (Smith, 2002) and scalp removing using Brain Extraction Tool 2 (BET2) (Jenkinson et al., 2005). Fig.2.16 shows a sagittal cut of the brain for different preprocessing steps.

In the last preprocessing step, in order to performs a fine skull stripping, BET2 (Jenkinson et al., 2005) exploit the fact that in an MR image, the skull appears darker than brain and other tissues with an intensity comparable to the background intensity. Since the scanner to scanner relative intensity variation of brain and scalp on T1 greatly varies, the maximum scalp intensity is constrained relative to the brain intensities to increase robustness. In lower area were tissues and skull topologies are too complex, BET2 is not used. The brain extraction for these areas relies on standard space registration.

The recursive call of BET2 is carry out until $i > 10$ or $\frac{1}{(x_{i-1}-x_i)^2+(y_{i-1}-y_i)^2+(z_{i-1}-z_i)^2} < 1$ with x_i, y_i, z_i the x, y, z positions of the center of gravity of the image after BET2 and i the number of BET2 iterations performed.

Segmentation : After obtaining a brain extracted T1 structural image, the preprocessed data needs to be segmented to identify the white matter mask. This step is performed by the TissueClassifierHMRF from DiPy (without biasfield correction). FAST from FSL also implements the same algorithm with the inclusion of the bias field correction. The theoretical model used by this segmentation's algorithm is explained in the subsequent paragraph (Zhang et al., 2001) (Avants et al., 2011).

Segmentation of MR images into different tissue classes, especially Gray Matter (GM), White Matter (WM) and Cerebrospinal Fluid (CSF) is a difficult task. In the past, the finite mixture (FM) model was one of the most frequently used statistical segmentation algorithm of MR images thanks to its mathematical simplicity. However, the FM model is a histogram based model which does not take into account spatial information. The class label is estimated using the pixel intensity since all pixel are considered as independent samples drawn from a population. Unfortunately, due to this limitation, this models commonly leads to unreliable results due to the presence of bias field distortion and multiple tissue class occupying one voxel at tissues' junction (partial-volume effect).

In order to handle these issues, the segmentation model used in ElikoPy is based on Hidden Markov Random Fields (HMRF) classifier and an associated Expectation-Maximization (EM) algorithm. HMRF model derives from Markov Random Fields (MRF) theory in which the spatial information of an image is encoded through contextual constraints of neighboring pixels leading to adjacent pixels having the same class labels and similar

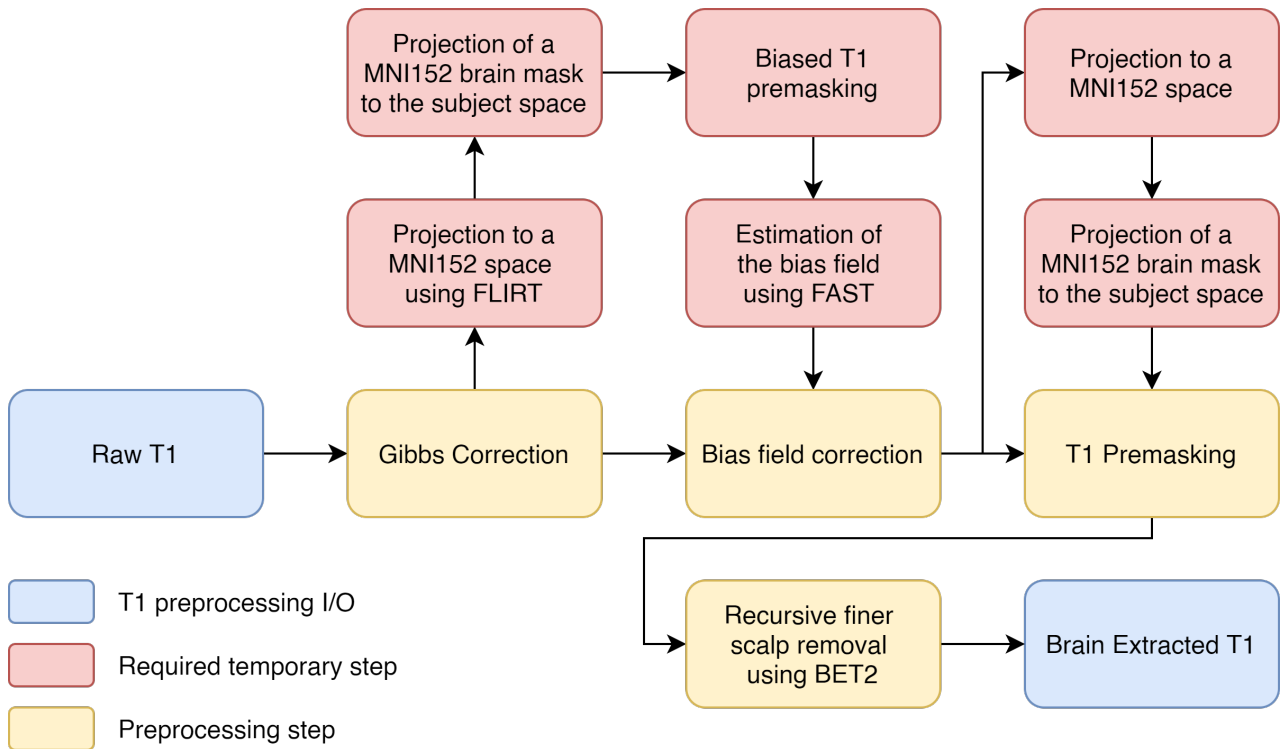


Figure 2.15: T1 preprocessing pipeline. The Gibbs corrected input image is premasked on the basis of a T1 2mm MNI152 standard space mask, that is transformed into the space of the input image, creating the biased premasked output. The bias field is then extracted from the biased premasked output using FAST. The Gibbs corrected input image is then unbiased using the bias field estimated from the biased premask. The bias field correction is applied by dividing the Gibbs corrected input image by the bias field estimation. Then, an unbiased premasked brain is extracted from the unbiased Gibbs corrected input using the same methods than the first applied on the Gibbs corrected image. At last, BET2 is applied recursively.

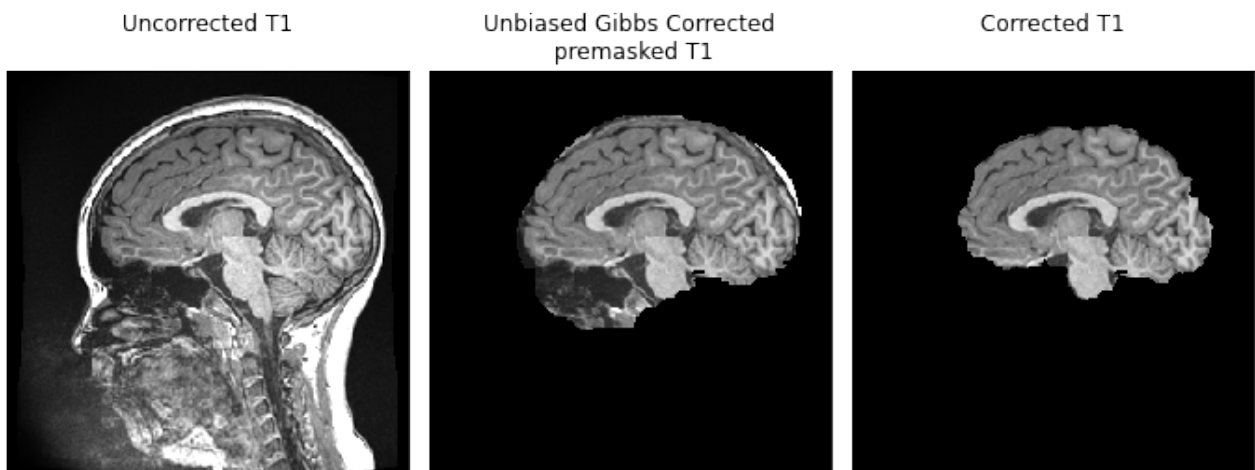


Figure 2.16: Intermediate pipeline step, uncorrected and the temporary unbiased premasked T1 images are shown in the left and middle panels, while corrected T1 is shown in the right panel. The premask used does not incorporate the lower part of the cerebellum by design.

intensities (Zhang et al., 2001). Moreover, the bias field correction algorithm presented in (Guillemaud and Brady, 1997) is incorporated into the model.

Registration : In order to project the white matter mask in the space of the dMRI data, the computation of a rigid then affine transform of the T1 image to the preprocessed diffusion data is performed using DiPy.

The global optimization method used by DiPy and FLIRT has been designed for volumetric registration of brain images. It uses a Mutual Information cost function to quantify the quality of the alignment of the two images for any given transformation. The registration problem can be expressed as the search of the transformation T^* which gives the minimum cost.

$$T^* = \arg \min_{T \in S_T} C(Y, T(X))$$

where $C(I_1, I_2)$ is the cost function, S_T is the space of all affine transformations and $T(X)$ represents the image X after it has been transformed by the transformation T .

In order to not fall in local minima, the optimization process uses a local optimization method with a multiresolution framework followed by a global search. The multiresolution framework is based on an optimization at four different decreasing voxel sizes. At each scale, the two images are resampled, after initial preblurring. To estimate the final transformation sufficiently accurately, a global search is performed at the lowest resolution (8-mm scale) where only the gross image features still exist. A coarse search of the cost function at this resolution should reflect the major changes in rotation, translation, and global scaling, allowing large misregistrations to be avoided.

Ultimately, the transform used for the registration of the structural image is applied to segmented white matter mask to obtain the final white matter mask in the space of the preprocessed diffusion data.

Anisotropic power map as T1 alternative : When no T1 is available, an Anisotropic Power (AP) map (Dell'Acqua et al., 2014) of the dMRI preprocessed image is used as a pseudo-T1 for segmentation purposes. This index is capable of quantifying and discriminating low anisotropy regions from CSF regions or noise where standard anisotropy indices fail to do so. As visible in Fig.2.17, both images are able to separate white and gray matter structures similarly. This pseudo-T1 is segmented to obtain a white matter mask. Since the AP map comes directly from the preprocessed diffusion image, no registration is needed after the segmentation step.

The computation of the AP map relies on the High Angular Resolution Diffusion Imaging (HARDI) acquisition schemes used to acquire the diffusion images. From each HARDI signal, the anisotropy information can be captured using the spherical harmonic coefficients of even order $l \geq 2$. The total anisotropic power is computed from the signal using

$$AP = \sum_{l=2,4,6,\dots} \frac{1}{2l+1} \sum_{m=-l}^l |a_{l,m}|^2,$$

where $a_{l,m}$ are the spherical harmonic coefficients of the normalized HARDI signal. Due to the exponential relation between the MR signal and diffusivity, this index is commonly expressed on a neper scale.

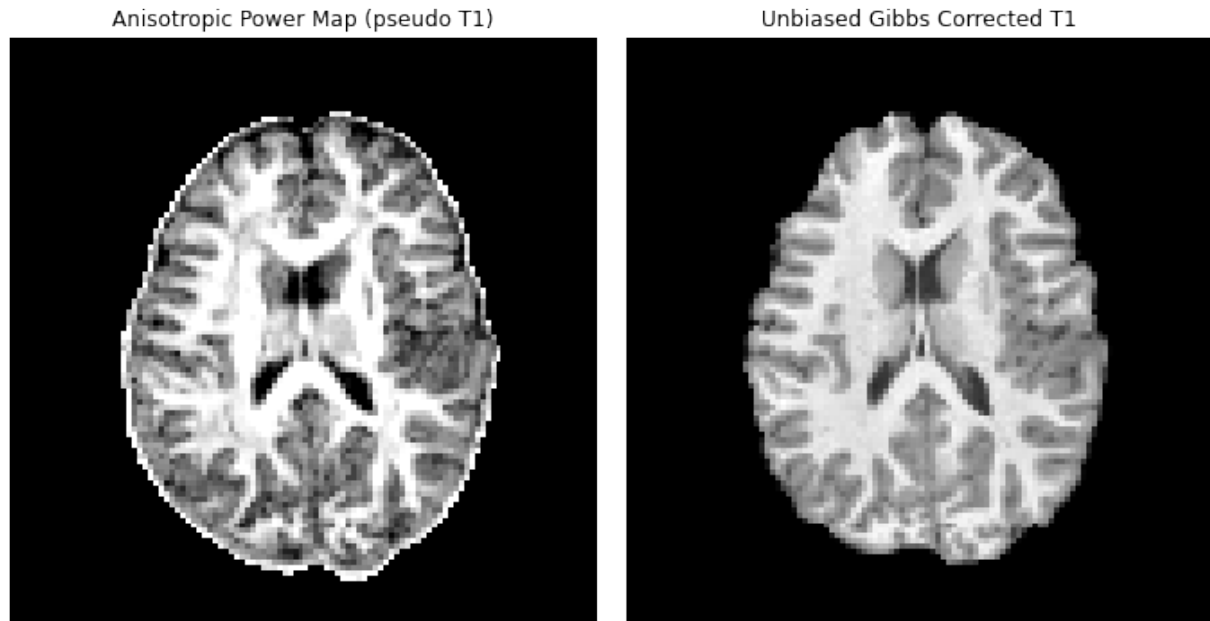


Figure 2.17: AP map is shown in the left panels, while a T1 weighted structural image obtained from the same subject is shown in the right panel.

2.2.4 Computed microstructural metrics

The metrics derived from DTI (Le Bihan et al., 2001), NODDI (Zhang et al., 2012), DIAMOND (Scherrer et al., 2016) and Microstructure Fingerprinting (Rensonnet et al., 2019) explained in the theoretical background are available for processing in the ElikoPy library. These metrics are computed from preprocessed diffusion data and can be associated with a white matter mask to decrease the processing time.

DTI metrics are computed based on the framework described in (Basser and Pierpaoli, 2011) (Basser et al., 1994) using the python package DiPy.

DIAMOND metrics are computed using a research code implemented by B. Scherrer, the main author of DIAMOND. ElikoPy estimates by default a maximum of two fascicles per voxel and uses an Unbiased Akaike Information Criterion (AICU) model selection criterion (McQuarrie et al., 1997) to determine the number of fascicles in each voxel.

NODDI metrics (Zhang et al., 2012) are evaluated using the python package Dmipy (Fick et al., 2019). By default, ElikoPy uses a Watson distribution to model the dispersion of the stick distribution. The NODDI diffusivities are fixed by default to their respective typical values in vivo ($d_{\parallel} = 1.7 \times 10^{-9} m^2 s^{-1}$ and $d_{iso} = 3.0 \times 10^{-9} m^2 s^{-1}$) but can be modified by the user. ElikoPy can use other NODDI based models implemented in the Dmipy library such as NODDI AMICO. Accelerated Microstructure Imaging via Convex Optimization (AMICO) is a method that accelerates the fit of existing techniques (e.g. NODDI) drastically by re-formulating the problem as a linear system that can then be solved using very fast algorithms (Daducci et al., 2015).

Microstructure Fingerprinting (MF) (Rensonnet et al., 2019) relies on DIAMOND tensor files in order to estimate the number of fascicles and their orientations. In the case where these files are not available, the fascicles orientations are computed using a constrained spherical deconvolution model (Tournier et al., 2007). The implementation is based on the python library coded by G.Rensonnet, the main MF author. Brain tissue properties

are estimated based on a dictionary containing the fingerprints of multiple microstructural configurations for numerous fascicle orientations. The dictionary of Monte Carlo DW-MRI fingerprints used in ElikoPy has been precomputed by the author of the MF paper and is provided as a static file in the ElikoPy library. It is similar to the heterogeneous packing dictionary described in (Rensonnet, 2019) with a fixed axon radius distribution and the addition of extra-axonal diffusivity.

After processing, the extracted microstructural features of the white matter are made available in their respective folders along with logging informations. A detailed list of the computed metrics for each model is available in Appendix A.

2.2.5 Common space registration and Statistics

The microstructural diffusion metrics estimated after preprocessing can be used to compute group comparison statistics in order to localize brain changes related to development, degeneration and disease.

In order to do these statistical analyzes, as synthesized in Fig.2.18, three ElikoPy's functions based on Tract-Based Spatial Statistics (TBSS) (Smith et al., 2006) are used :

1. **regall_FA** : This function performs a carefully tuned nonlinear registration of FA metrics, followed by a projection onto an alignment-invariant tract representation (skeletonized FA) which is then thresholded. During the registration, subjects can be labeled as part of two groups. The FA images are aligned to a 1x1x1mm standard space by default but can also be aligned to the most "typical" subject in the study. This function is a prerequisite for the regall and randomise_all functions.
2. **regall** : This function applies to other metrics the nonlinear warps and skeleton projection vectors computed by the regall_FA function. One big advantage of the regall function is that it can take any metrics in parameters and register it to a common space and a mean skeleton. Enabling the possibilities of doing statistics on these metrics.
3. **randomise_all** : This function based on randomise (Winkler et al., 2014) from FSL performs nonparametric permutation inference between the two groups defined in the regall_FA function. The user can choose to do the statistics on any aligned microstructural metrics and metric's skeleton. These permutations results in a test statistic image and sets of (1 - P-value) images that can be FWE-corrected. "FWE-corrected" means that the Family-Wise Error (FWE) rate is controlled. A list of names of significant clusters and region wise mean FA in csv form are also produced by the function.

The operations performed by the reg_all_FA function aims at avoiding the issues occurring when only doing a common space high degree of freedom registration known as Voxel-based Morphometry (VBM) (Ashburner and Friston, 2000) to perform voxel wise statistic. Indeed, several challenges can occur when doing VBM (Bookstein, 2001) (Davatzikos, 2004). The major issue is that the complete successful registration of every subject data to a common space cannot be formally ensured, both in terms of resolving topological variabilities and having an exact alignment of the very fine microstructural specificities present in the data. VBM relies on spatial smoothing to reduce the first outlined problems. However, the amount of spatial smoothing applied is arbitrary and can greatly affect the statistical results (Jones et al., 2005).

To overcome the issue of VBM, manual tractography based approach can be used to allow

for analysis of specific target white matter tracts using hand-drawn ROIs. However, this approach is time consuming since it requires a lot of human intervention and is thus not eligible for the ElikoPy library and will not be further discuss.

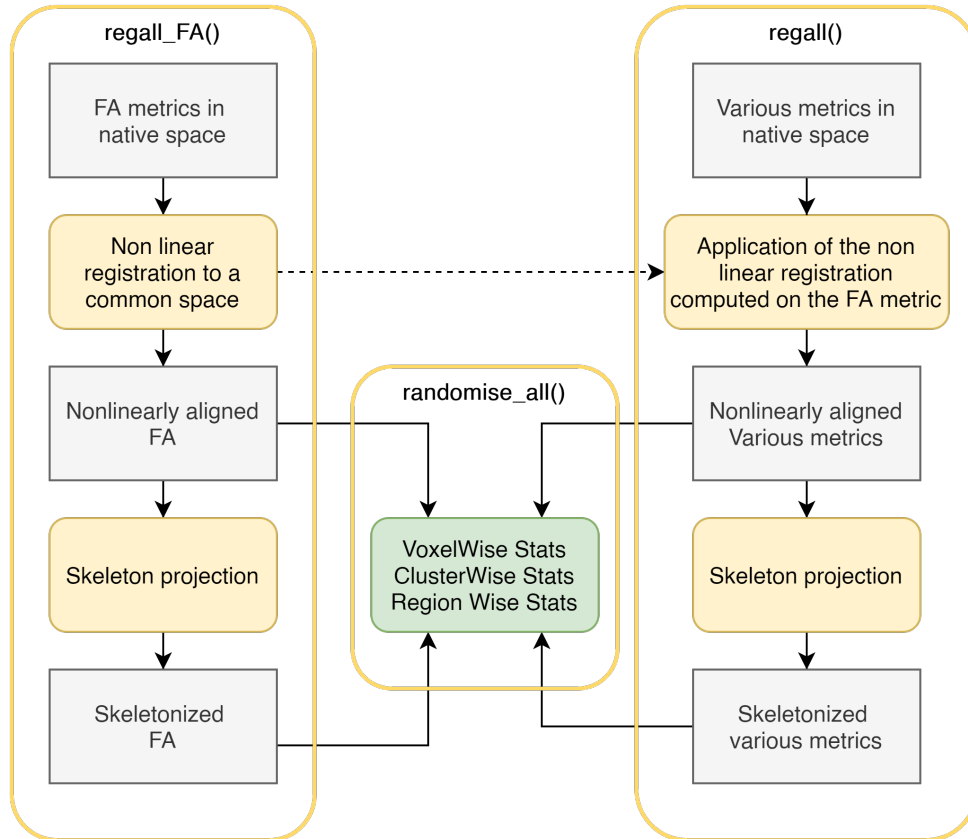


Figure 2.18: Main functionalities available with the common space registration and statistics toolset of ElikoPy.

The approach taken by TBSS and implemented in ElikoPy aims at bringing together the strengths of each method. TBSS is an automatic scalable approach investigating the whole brain like VBM. However, it also relies on information offered by tractography thanks to the use of the FA skeleton which represents the centers of all fiber bundles.

In summary, `regall_FA` performs the three following steps :

1. All subjects FA are aligned to a common space using a nonlinear registration. At this stage, perfect alignment is not expected or required.
2. Computation of the mean FA skeleton to remove the areas of low mean FA and high inter-subject variability.
3. Projection of each aligned subject FA onto the skeleton by filling the skeleton with FA values from the nearest relevant track center.

When computing voxel wise group statistics, the `randomise_all` function performs permutation inference with a general linear model (Winkler et al., 2014) using a nonparametric 2-sample t-test. The number of exhaustive permutations is given by

$$\frac{(n_1 + n_2)!}{(n_1! \cdot n_2!)},$$

where n_1 and n_2 are the number of subjects in the first and second group. Evaluating

every possible permutation is often not possible. Instead, a Conditional Monte Carlo (CMC) permutation test can be used. The CMC evaluates a random sample of possible permutations. If the number of possible permutations is large, one can show that a exhaustive P-value of p will produce P-value between $p \pm \sqrt{\frac{p(1-p)}{n}}$ about 95% of the time, where n is the number of CMC permutations (Pesarin and Salmaso, 2010).

Along voxel wise t statistics and f statistics, the `randomise_all` function generates several text file reports associating the center of mass of the significant voxel clusters (i.e. with p -values smaller than 0.05) to labeled atlas regions. For each metric and each atlas selected by the user, four different reports are created that contain the probability of the significant clusters to belong to the atlas regions where each of the two compared group diffusion metric is significantly different regarding the corrected and uncorrected t test.

In addition to compute voxel wise statistics on the whole brain or in the FA skeleton, ElikoPy also computes for each user selected diffusion metric a csv file containing the region wise mean metric value. The atlases used to compute the region wise values are probabilistic atlases and the mean are therefore computed as weighted means

$$W_r = \frac{\sum_i p_{i,r} y_i}{\sum_i p_{i,r}}, \quad (2.17)$$

with $p_{i,region}$ the probability of voxel i to belong to the region r .

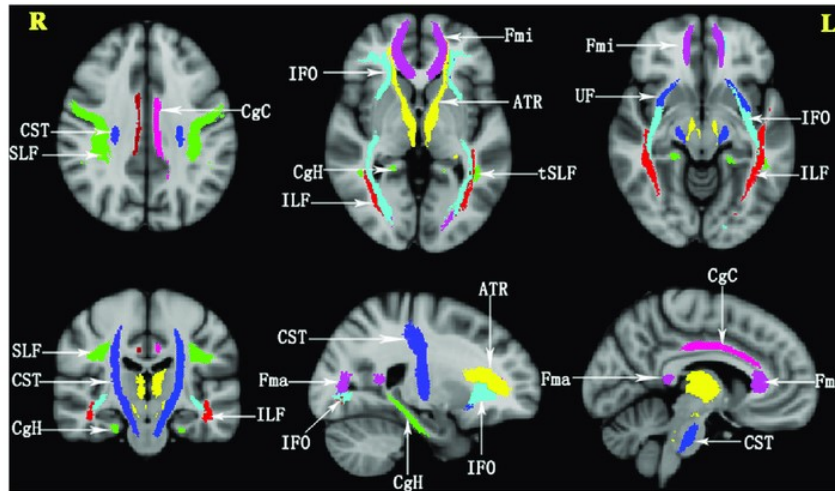


Figure 2.19: Details of the JHU White Matter (WM) Tractography Atlas. ATR, anterior thalamic radiation; CST, corticospinal tract; CgC, cingulum (cingulate gyrus); CgH, cingulum (hippocampus); Fma, forceps major; Fmi, forceps minor; IFO, inferior fronto-occipital fasciculus; ILF, inferior longitudinal fasciculus; SLF, superior longitudinal fasciculus; UF, uncinate fasciculus; tSLF, superior longitudinal fasciculus-temporal part; L, left; R, right.; from (Luo et al., 2020).

The atlases currently supported by the ElikoPy library are

- The Harvard-Oxford cortical and subcortical structural 1mm probabilistic atlases (Desikan et al., 2006)
- The JHU DTI-based white matter tractography 1mm probabilistic atlas (Mori et al., 2005) (Hua et al., 2008)
- The MNI structural 1mm probabilistic atlas (Collins et al., 1995) (Mazziotta et al., 2001)

The JHU white matter tractography atlas is shown in Fig.2.19

Finally, when T1-images are available, ElikoPy will perform a volumetric brain analyzes to give an overview of the macroscopic changes that might have occurred. Using the T1-segmentation of section 2.2.3, Elikopy outputs the volumes of CSF, white and gray matter in the brain.

2.2.6 Quality control report

In addition to all the processing capabilities of the pipeline, a powerful quality control step has been integrated to ElikoPy. This control step is motivated by the fact that applying all the processing step to DW-MRI data does not de facto ensure that data are artifact-free. On the one hand, tools used for correction are not perfect and will sometimes underperform, leading to residual data corruption. On the other hand, the majority of the tools applied in ElikoPy have been designed separately by different experts. Manipulating data between those tools presents an additional risk of data corruption and underperformance of algorithms.

The quality report comes in the form of a subject specific PDF document with dedicated sections for each preprocessing and processing step. The pipeline also saves all step-specific intermediates report in separate PDF documents. To ease the analysis for the reader and considering the high dimensionality of data, qualitative visualizations, quantitative plots and metrics are available.

The preprocessing report consists of :

- A summary table of the input parameters and subject characteristics.
- For the raw data and for each intermediate preprocessing step, several slices of multiple volumes with different b-values are displayed.
- An estimation of the SNR.
- An estimation of the MPPCA Noise SNR and standard deviation.
- A visual comparison of the effect of the Gibbs ringing correction, including a visualization of the residual.
- A visual comparison of the effect of the MPPCA denoising, including a visualization of the residual.
- A visual representation of the tSNR after and before preprocessing for several volumes with different b-values.
- A visual representation of the susceptibility field estimated by Topup.
- Tables synthesizing the estimate volume-to-volume motion, detected outlier slices and Eddy currents (obtained with Eddy QC (Bastiani et al., 2019)).
- Plots summarizing the estimated translations (mm), rotations (deg) and Eddy current (Hz) for all axis (obtained with Eddy QC).
- Plots illustrating outliers for every slice of every volumes using the distance in terms of standard deviation (obtained with Eddy QC).
- tSNR and CNR maps (obtained with Eddy QC).
- Group report plots showing the performances of the individual in contrast to the other individual of the study in terms of motion (translations, rotations), amount of outliers slice, amount of outliers volume, Eddy current, average SNR and average CNR (obtained with Eddy QC).

Report for microstructural models includes plots of the MSE, R2 along some specific diffusion metrics.

A reporting step is also available for the extraction of the white matter mask based on the T1. This reporting step includes, the raw T1, the Gibbs ringing corrected T1 and the brain extracted bias field corrected T1. Moreover, the segmented white matter mask in the DW-MRI subject space is also plotted for several slice on top of the first b0 of the DW-MRI data.

An example of a quality report is available in appendix D.

2.2.7 Research oriented design choices

As stated earlier, the ElikoPy is fully open-source (AGPL3 License) and available publicly on Github. Moreover, an extensive documentation and an API is available on <https://elikoPy.readthedocs.io/en/latest/> and in appendix E. If users confront problems, they can find commonly encountered issues in the FAQ or open an issue in the Github repository. To facilitate the installation of ElikoPy, a conda and pip package have been made available. If the user does not want to experience the hassle of installing Cuda, FSL, Freesurfer, ITK, DIAMOND, Mrtrix3 and Ants (all these programs are not always required depending on which processing steps are applied), end users can use a preconfigured containerized environment using the provided Singularity image (also compatible with docker).

Along this end user oriented installation process, the ElikoPy pipeline has also been designed with cluster ready processing in mind. The library natively supports the Slurm Workload Manager widely utilized on high performance computing unit such as the one from the CECI and CISM consortia. Moreover, in order to speed up the processing, the majority of the computing intensive tasks of the library are compatible with Cuda for GPU processing and Numba (Lam et al., 2015) for an efficient compilation of Python code in order to optimize performance.

The quality control and the extensive processing logs provided with the preprocessing can help clinicians more easily understand the steps performed in the preprocessing and which subjects of the dataset are of poor quality. Thanks to the multiple subjects class approach, the pipeline easily adapts to multiple acquisitions procedure, longitudinal studies and case-control study type. Having variability on the type of data acquired (such as the absence of T1 or multiple encoding directions for all subjects or a change in the acquisition parameters) is also not an issue.

Finally, export functions are available if one specific type of data files common in all subjects needs to be exported into one single folders.

2.2.8 Relevance for clinical data and population studies

The ElikoPy pipeline has already been used for studies involving dyslexic children, meditator and hemiplegic children.

As part of a meditator group study at the institute of neuroscience of the *Cliniques Universitaires Saint-Luc* in collaboration with Sune Darkner (university of Copenhagen), meditator and control data were preprocessed and used to compute microstructural metrics. Voxel and region wise statistical analyzes were also performed using the pipeline. This processing takes part in a multi-modal multi-scanner brain imaging study on the effect of meditation on the brain structure and function, in which participants were tested using

various imaging modalities, including dMRI (Kupers et al., 2021).

The pipeline was used for a dyslexia study, in collaboration with the Development of Literacy Lab (DeLi Lab), to preprocess and compute microstructural metrics from dyslexic and control data. The dMRI study compared dyslexic children with age-matched control subjects. The results revealed differences in WM microstructure in the arcuate fasciculus, a fiber tract involved in literacy. An abstract was submitted to the 2021 Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM) (Delinte et al., 2021).

2.3 Methods and materials

In this section, are presented the techniques implemented in order to validate the ElikoPy pipeline. The data used and the way they have been processed are also described.

2.3.1 Subjects and data

To validate the pipeline, two types of data have been used : in vivo and simulated diffusion weighted images. These two datasets and their acquisitions are described in the following sections.

2.3.1.1 In vivo DWI data

To assess the quality of the pipeline, a dataset of 35 children aged from 8 to 11 was used. From the 35 subjects, 17 are dyslexic children and the 18 remaining subjects are age-matched control patients. Anatomical and diffusion weighted images were acquired at St-Luc University Hospital on a 3T SIGNA™ Premier GE MRI scanner, equipped with a 48-channel head coil. During the analysis (section 2.3.2), the data processed and not processed are compared, it has been considered that the differences between the control and dyslexic patients are negligible compared to the differences coming from the processing.

Anatomical data : The T1 weighted anatomical images were acquired using an MPRAGE sequence with $TR = 2187.84ms$ and $TE = 2.96ms$. The matrix size is $156 \times 256 \times 256$ with isotropic voxels of size $1 \times 1 \times 1mm^3$.

Diffusion data : The diffusion MRI images were acquired using a PGSE EPI sequence with $TR = 4842ms$ and $TE = 77.4ms$. The matrix size is $256 \times 256 \times 64$ with voxels of size $0.8594 \times 0.8594 \times 2mm^3$ and no inter-slice gaps (resulting in a $220 \times 220 \times 128mm^3$ FOV). A multi-shell diffusion scheme was used and each data is made of 167 volumes :

- 7 baseline volumes with a b-value of $0[s/mm^2]$.
- 64 volumes with direction gradients and a b-value of $1000[s/mm^2]$.
- 32 volumes with direction gradients and a b-value of $2000[s/mm^2]$.
- 32 volumes with direction gradients and a b-value of $3000[s/mm^2]$.
- 32 volumes with direction gradients and a b-value of $5000[s/mm^2]$.

The shells are represented in Fig.2.20.

The gradient intensities were 30.80, 43.55, 53.34 and 68.86 $[mT/m]$ for the shells with b-values of respectively 1000, 2000, 3000 and 5000 $[s/mm^2]$. The duration of the diffusion

gradients pulses (δ) and their separation (Δ) were set to $23.824ms$ and $35.66ms$. The slices were acquired in an interleaved manner with a multi-band factor of 3 with gradient cycling (this particular acquisition scheme is discussed in depth in Chapter 3). All volumes were acquired with an Anterior-to-Posterior phase encoding. The echo spacing and EPI were set to $0.66[ms]$ and 128 respectively, resulting in a time between reading the center of the first echo and reading the center of the last echo of $0.0845[s]$ (i.e. reciprocal of the Bandwidth in the phase encoding direction).

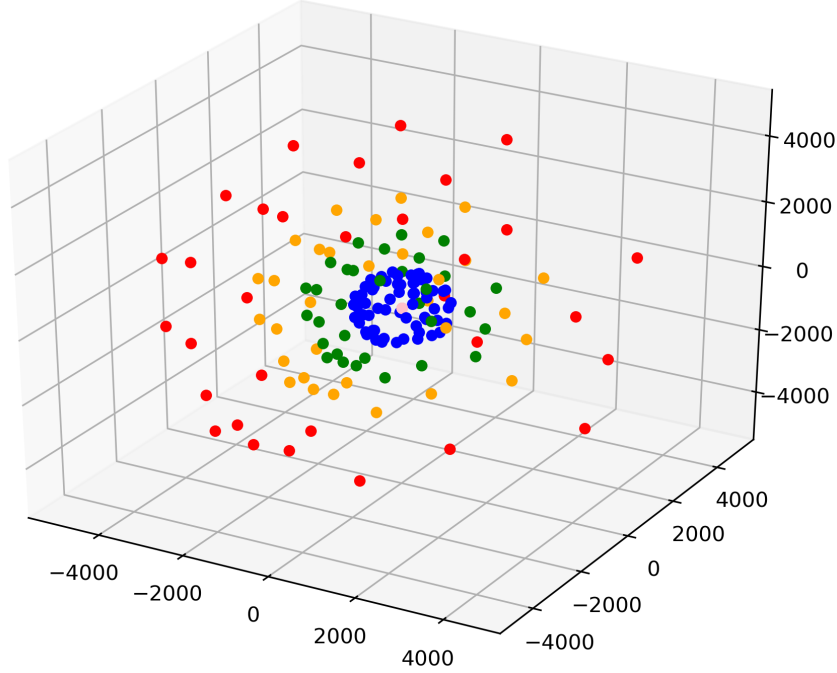


Figure 2.20: Gradient shells represented on a 3D sphere, the radius of the spheres are scaled by the b-value and the orientation of the points corresponds to the directions of the gradients. b-value of 0, 1000, 2000, 3000 and 5000 $[s/mm^2]$ are represented respectively in pink, blue, green, orange and red.

Additional data : One additional data has been acquired. This subject has followed the same acquisition protocol but with 11 additional volumes. These last 11 volumes were acquired with a reverse phase encoding (i.e. Posterior-to-Anterior). Since this subject has both a T1-image and reverse phase encoding volumes, it is possible to compare the estimation of the susceptibility off-resonance field of Topup with and without the use of Synb0-DisCo.

Processing : The dataset has been processed using the complete pipeline of Fig.2.2. The data were converted from DICOM to NIfTI and preprocessed before computing the white matter mask and the diffusion metrics. Four diffusion metrics have been registered to the common MNI space to perform statistical analysis, namely the FA (DTI), the ODI (NODDI), the kappa values (DIAMOND) and the $f_v f_{tot}$ (MF). The parameters used during the processing are summarized in Appendix B.2. For the preprocessing step it can be noted that the data had to be resliced to isotropic voxels of size $2 \times 2 \times 2mm^3$. This is due to the fact that this is the actual resolution of the image and the data provided at the output of the scanner were interpolated which is detrimental for the other preprocessing

steps and greatly increases the processing time. Unfortunately, due to the gradient cycling acquisition scheme, the slice-to-volume motion correction could not be activated as Eddy FSL is not compatible with it, this is deeply discussed in chapter 3. The susceptibility off-resonance field was estimated using the Synb0-DisCo framework (i.e. using T1-images) only for the 21TD subject due to the processing time of Synb0-DisCo. The other subjects could not be corrected for the susceptibility induced distortion because of the absence of a reverse phase encoding image, with the exception of the additional data described above.

2.3.1.2 Simulated DWI data

Simulated DWI data are generated based on a framework called DW-POSSUM (Graham et al., 2016). DW-POSSUM is an extension of POSSUM FSL (Drobnjak et al., 2006) which allows to simulate MRI data by solving the Bloch and Maxwell equations. It produces realistic DWI data and artifacts introduced during the acquisition of the MR signal in k-space, such as EC induce distortions or movement.

DW-POSSUM framework : The operations of DW-POSSUM for the simulation are summarized in Fig.2.21. Four inputs are required : a geometric 4D object where the fourth dimension is the tissue-type (WM, GM and CSF) along with the T1 and T2 values, a representation of diffusion weighting, a pulsed-gradient spin-echo (PGSE) sequence and details on the artifacts to produce (e.g. motion parameters). The PGSE sequence is converted to a SE-EPI sequence which includes EC-induced distortions. The DWI is created by taking the attenuated object (i.e. with diffusion), the pulse sequence and artifact informations and solving the Bloch's and Maxwell's equations (equation 1.14) at each point in the object, summing the resultant signal in order to generate the k-space measurements (Graham et al., 2016). A Fourier transform is then applied to obtain the final DWI.

Generated datasets : Using DW POSSUM, two datasets are generated, a ground truth DWI dataset (*GT data*) and the corresponding dataset with the following artifacts (*Possum data*) :

- Thermal noise $SNR = 20$.
- EC-induced distortions, applied with the SE-EPI sequence.
- Rigid body movement applied before the MR simulation starts in order to model volume-to-volume motion.

The profiles of the head rotations used to generate motion and the parameters of the simulation are available in reference (Graham et al., 2016). The data is composed of 32 gradients at $b = 700[s/mm^2]$, 64 at $b = 2000[s/mm^2]$ as well as 12 b_0 . All simulations were performed at $3T$ using an echo-planar imaging (EPI) pulse sequence with $TE = 109ms$, $TR = 7500ms$ and a matrix size of $72 \times 68 \times 54$ (isotropic voxel size of $2.5mm$).

Custom simulation : The dataset generated using the framework of DW-POSSUM lacks several artifacts, namely susceptibility induced distortions, bias field and slice-to-volume motion. It is possible to generate a dataset with all non idealities using DW-POSSUM and another POSSUM extension named spin-echo-POSSUM (Graham et al., 2017). However this requires a tremendous amount of processing power which is unfortunately not available. Therefore a suboptimal approach is taken for the simulation and all artifacts are added in image space instead of being added during the acquisition process. This leads to less

realistic artifacts since this method will fail to capture non geometric effects such as blurring due to EC decay (Graham et al., 2016). This constitutes the third simulated dataset (*Custom data*).

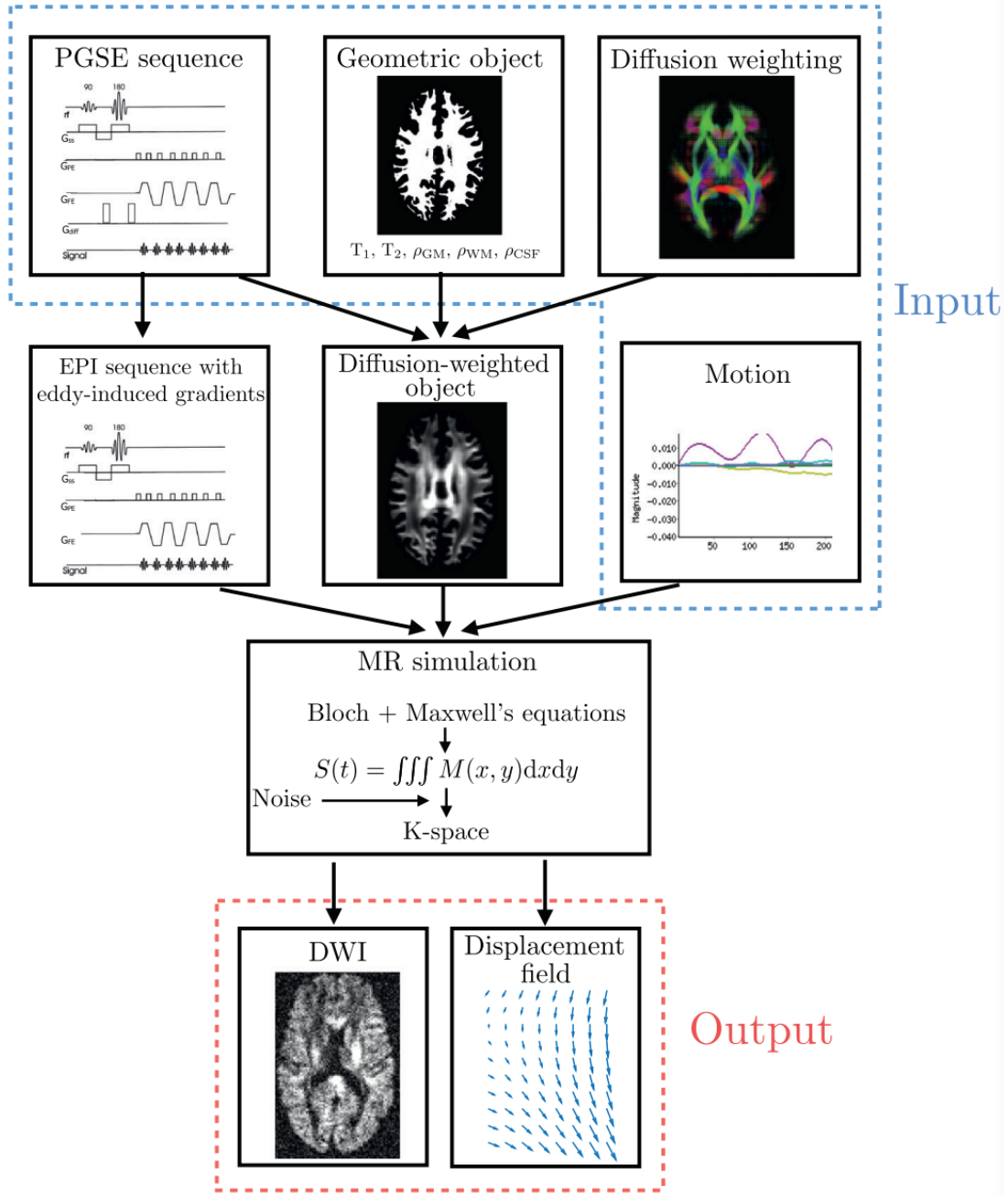


Figure 2.21: DW-POSSUM pipeline for simulating diffusion weighted images, from (Graham et al., 2016).

First, a bias field is introduced by multiplying the data with low frequencies variations centered around 1. The motion (V2V and S2V), EC-induced and susceptibility induced distortions are then added at the same time using the same approach as Eddy FSL (section 2.2.2.5). From the *GT data*, the coordinates in the distorted space are obtained, using equation 2.11 and the rigid body motion theory in homogeneous coordinate of equation 2.10. Then, the brain is interpolated to obtain the distorted image. The motion vectors

$\mathbf{r}(t_n)$ are generated using a random walk approach as it is considered that for each time step, the subject has an equal probability to move in any direction. This results in pretty sharp displacements such that the correction is performed in a "worst-case" scenario. Eddy current has been modeled the same way as Eddy FSL for a quadratic model with typical parameters values obtain from subjects. The Susceptibility field used has been generated using the FSL command "b0calc" on the air-tissue segmentation of the brain (Graham et al., 2017). The acquisition parameters considered for the addition of the artifacts are an anterior-to-posterior phase encoding except for the second b0 acquired in an posterior-to-anterior fashion. The time (in seconds) between reading the center of the first echo and reading the center of the last echo has been set to 0.1s. Motion has been added only on the 10 $\rightarrow$ 20 and 71 $\rightarrow$ 81 volumes to reduce processing time and to allow to compare the corrected slices to the other volumes. For the acquisition of the 54 slices, a multi-slice interleaved excitation scheme was simulated with a multi-band factor of 3 (see Fig.1.17).

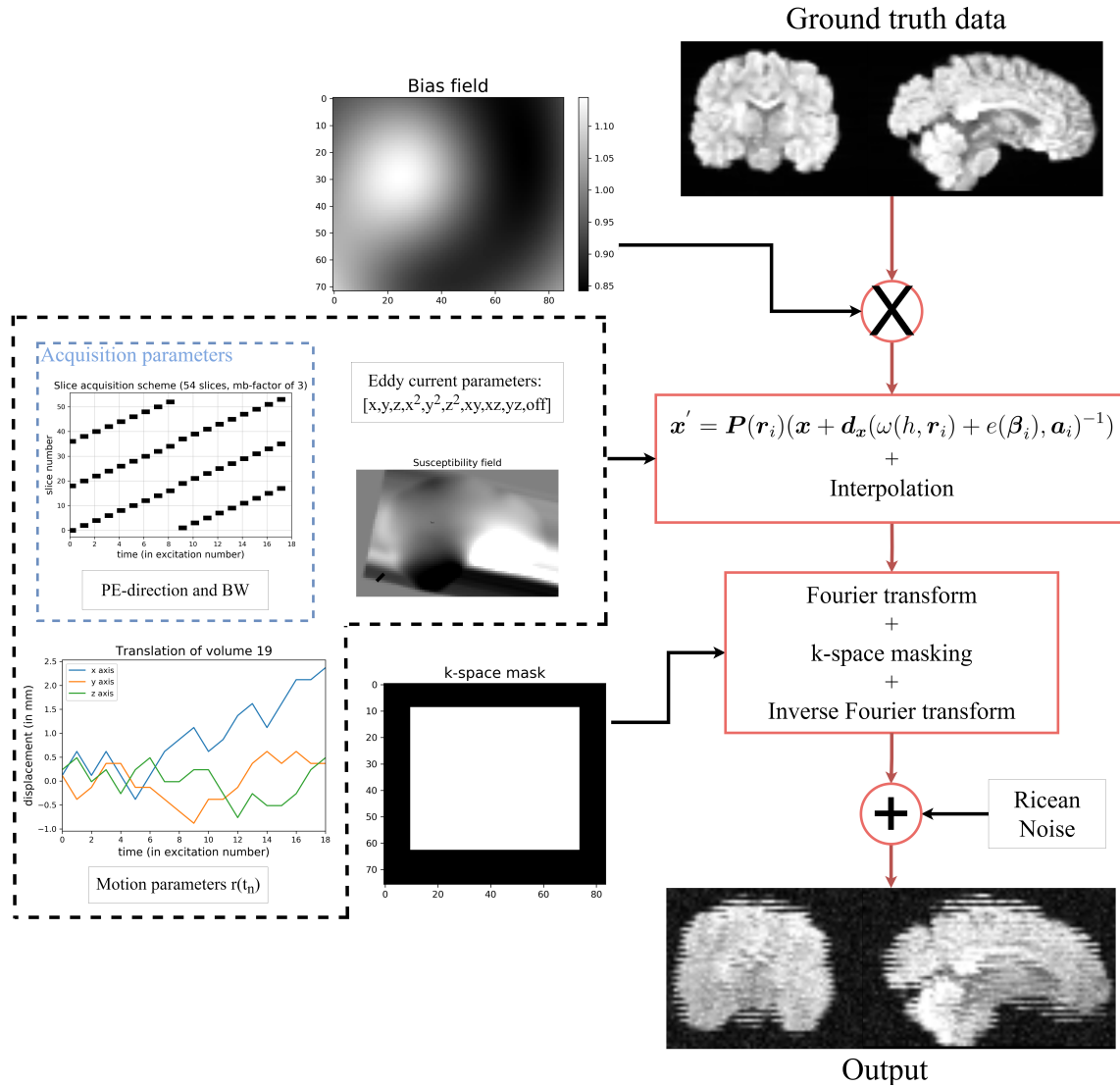


Figure 2.22: Simulation pipeline for the *Custom data*. 1) motion, susceptibility and Eddy current artifacts are created in the image space 2) Gibbs ringing is added by cropping the image in k-space 3) a bias field is multiplied with the image 4) Ricean noise is added.

Gibbs rigging is increased by masking the data in the k-space with a rectangular window in order to keep 80% of the lower frequency content. Finally, thermal noise is added to the image by adding to the data the module of a complex gaussian signal with $\sigma = 1$. The complete simulation process is summarized in Fig.2.22.

The code used to generate the data and the simulation files are available on the GitHub repository of the ElikoPy library (Simon and Dessain, 2021a).

Processing : The simulated datasets have been processed using the complete pipeline of Fig.2.2. As no T1-images are produced during the simulation, the white matter mask were computed using the Anisotropic Power map. It was also not necessary to register the data in a common space since the three datasets are the same 'subject' but with different artifacts. The *GT data* has been processed in order to assess the processing error of the pipeline. For the *GT data* and the *Possum data* it was not possible to use Topup FSL due to the absence of T1-images and reverse phase encoding (in practice this is not useful nonetheless as those two datasets are not affected by susceptibility induced distortions). All datasets have been corrected for motion using the slice-to-volume motion correction of Eddy FSL.

To compare with other existing pipelines, the datasets have also been processed using PreQual and DESIGNER (see section 2.1). The datasets have not been processed with the DTIPrep pipeline since the tool is designed to only performs DTI and DTIPrep could not output the preprocessed DW-MRI. The pipeline used for the UK-BIOBANK was also not used since its source code was not publicly disclosed. The parameters used for the processing are summarized in Appendix B.1.

2.3.2 Validation methods

To quantify the effects of the pipelines, in addition to visual inspection, several metrics and methods have been implemented.

2.3.2.1 Evaluation of the diffusion metrics

Using the diffusion models (section 1.3), it is possible to assess the quality of the data processing by computing goodness of fit metrics.

In vivo data : Using the in vivo dataset, for each diffusion model, the reconstructed signals y_{rec} of the processed and unprocessed data are extracted (i.e. the DWI reconstructed from the diffusion parameters of the model). The mean squared error (MSE) and the coefficient of determination (R2, square of the Pearson correlation) are then computed as

$$MSE = \frac{1}{M} \sum (y_i - y_{rec,i})^2 \quad (2.18)$$

$$R2 = \frac{(\sum_i (y_i - \bar{y}) (y_{rec,i} - \bar{y}_{rec}))^2}{\sum_i (y_i - \bar{y})^2 \sum (y_{rec,i} - \bar{y}_{rec})^2}, \quad (2.19)$$

where the sums are along the diffusion dimension and M is the number of diffusion measures. This results in voxel wise goodness of fit maps for each subject. The mean values are then reported along the gray and white matter (the brains have been segmented using the framework of section 2.2.3) and compared between the processed and unprocessed

data. Ideally, the MSE and R2 should not show the structure of the underlying data meaning the model does fit all part of the brain equally, however this is generally not the case in practice. Yet, it is expected that the error of the model fit should decrease with the processing of the ElikoPy library. In MSE the lower is better, while R2 has an upper bound of one, which represents a perfect model fit. The comparison is performed using the `scipy stats python` package Welch’s t-test (Virtanen et al., 2020). A Welch’s t-test is similar to a student t-test except that it does not assume equal variance. It is more robust than Student’s t-test and the results of Welch’s t-test come close to that of Student’s t-test, even when the population variances are equal and sample sizes are balanced. It can therefore be applied directly and without any substantial disadvantages to Student’s t-test (Ruxton, 2006). One can note however that the assumption of normally distributed samples is maintained.

Using the ElikoPy library functions (section 2.2.5), the voxel wise group statistics are also realized to extract regions and brain maps of the significant statistical differences between the processed and unprocessed data for the final diffusion metrics. The effect of processing has been assessed on four different diffusion metrics : FA (DTI), ODI (NODDI), kappa (DIAMOND) and fvf_{tot} (MF). These four metrics have been chosen in order to assess the improvements on various diffusion models and are commonly used metrics for each of the four models. The results are reported region -wise using the JHU DTI-based white matter atlas available in FSL (see section 2.2.5). It can be noted that the analyzes are not performed on the white matter skeletons. Using the skeletons allows to alleviate the alignment problems that may occur during the registration step. However, this means that the analysis would be performed only in highly aligned fascicles, while multi-fascicles models can investigate metrics in voxels with crossing fascicles. Moreover, the benefits of the preprocessing are thought to overshadow the possible misregistration issues.

Simulated data : For the simulated data, the ground truth diffusion estimates can be obtained using the *GT data*. The error on the final diffusion metrics can therefore be obtained directly for the processed and unprocessed *Possum data* and *Custom data*. The error is also computed for the processed *GT data* in order to assess the processing error of the pipeline.

2.3.2.2 Evaluation of the SNR

In vivo data : To quantify the effects of the pipelines using a conventional data quality metric, the temporal signal-to-noise ratio (tSNR) was estimated (Roalf et al., 2016). It is defined as

$$tSNR = \text{mean}_{\text{voxel}} / \text{std}_{\text{voxel}}, \quad (2.20)$$

where the quantities are computed along the diffusion dimension. The tSNR is estimated for each shell and provides an estimated SNR map for each subject. These maps are then used to compute statistical results and differences. In (Roalf et al., 2016), data have been visually classified between poor, good and excellent. It has been shown that the tSNR best differentiated poor data from good and excellent data with values significantly lower in the poor group, highlighting the interest of tSNR as a validation metric. Another advantage compared to a strictly b_0 -based SNR estimation is that usually a low number of b_0 images are acquired resulting in poor and unreliable estimations. The tSNR is here also computed on the weighted images and while these images are of lower SNR overall, significantly more of these images are collected, which provides a more robust estimate of SNR.

Simulated data : Using the *GT data*, it is possible for the simulated datasets to compute the 'real' signal-to-noise ratio. For a true signal I and its approximation $\hat{I}$, the SNR in decibels is given by :

$$SNR = 10 \cdot \log_{10} \left(\frac{\sum_{i,j} I_{i,j}^2}{\sum_{i,j} e_{i,j}^2} \right), \quad (2.21)$$

where $e = I - \hat{I}$ is the approximation error. This results in a value per volume. Using the term SNR is a slight abuse of notation. Indeed, the value of equation 2.21 will be influence by the noise but also the artifacts affecting the data. Therefore, the term Signal-to-Noise-Artifact Ratio (SNAR) is preferred in this report.

2.3.2.3 Comparison with ground truth parameters

For the simulated datasets, the parameters used to generate the motion, the susceptibility off resonance field, the Eddy currents, the bias field and the noise are known. The parameters estimated by the preprocessing step of the ElikoPy pipeline can therefore be directly compared to the true values.

2.3.2.4 Susceptibility off resonance field estimation

Using the additional in vivo data acquired with a reverse phase encoding, the performance of Topup with two phase encoding directions is visually assessed and compared to the performance of Synb0-DisCo with a single phase encoding directions and a T1 structural image.

2.3.2.5 Evaluation of the white matter mask

Using the in vivo dataset, the quality of the T1 structural image preprocessing, registration and segmentation performed in the white matter mask function of the ElikoPy library is visually assessed. The intermediate steps of the preprocessing are inspected and compared to graphically evaluate the quality of the Gibbs ringing artifact removal, the bias field correction and the brain extraction.

Using a graphical comparison in the preprocessed dMRI space, the Anisotropic Power (AP) map index is observed to determine his capabilities of quantifying and discriminating white matter regions from CSF, gray matter regions and regions with non brain tissues. The quality of the AP map as a pseudo-T1 for segmentation is also evaluated through a comparison of the segmentation obtained from the structural T1 image and the AP map.

2.3.2.6 Other validation methods

Age correlation : White matter anisotropy rapidly increases with age during neonatal development and continues to increase until the middle of the thirties (Barnea-Goraly et al., 2005)(Yoshida et al., 2013). Therefore, the link between the fractional anisotropy (FA) and age can be used in order to observe the effect of processing on the in vivo dataset. The FA needs to be estimated over the white matter skeleton and a linear model is fitted to predict the age. The correlation value is then reported. An increase in correlation is expected when the data is processed indicating improved relevance of the statistical results obtained with processed data. This idea has been used in (Roalf et al., 2016) and (Maximov et al., 2019).

Unfortunately, the available in vivo dataset only covers a limited number of ages and could not be used to easily validate the pipeline using this technique. The description has still been given to present the other validation methods that have been explored.

Generalization error : The Generalization Error (GE) quantifies the capability of a diffusion model to accurately predict the DW signal for unobserved gradient directions and b-values. It is based on the assumption that a diffusion model that well captures the biophysical mechanism of diffusion for the data should accurately predict the signal for new gradient directions and strenghts. It can be estimated using the 0.632 bootstrap method which provides an estimate of GE with low bias and low variance (Efron, 1983). This metric was originally designed to provide quantitative evaluations of biophysical models of the diffusion (Scherrer et al., 2014) and was used in (Scherrer et al., 2016). However, it is expected that the preprocessing should decrease the GE by allowing the models to better fit the data and the underlying biophysical diffusion mechanisms.

Unfortunately, due to the extensive computation time required by the models and the time available, this method could not be implemented.

2.4 Results

2.4.1 Visual results

Using the simulated datasets, the results of the ElikoPy processing can be visualized and compared to the ground truth brain. In Fig.2.24, the effect of motion correction is highlighted. One can observe that the motion is corrected for both volume-to-volume and slice-to-volume movement. The ability of the preprocessing to recover from slice-to-volume is also demonstrated in Fig.2.25. Due to motion, the raw data fails to match the ground truth mask and the slices are not properly aligned. This discrepancy does not appear on the processed data.

To assess the brain extraction step, in Fig.2.23 the brain mask is overlaid on the b_0 volume of the processed *Custom data*. One can observe that the brain segmentation does manage to extract correctly the brain.

Similarly to the figures for the motion, the effect of the distortions corrections are shown in Fig.2.26. It can be noted that the algorithm properly corrects for the distortions and brings the volumes in a common undistorted space.

The effect of Gibbs ringing suppression is shown in Fig.2.27. The Gibbs ringing is effectively removed and the artifact does not appear on the output of the pipeline. The results of the MPPCA denoising are also given in Fig.2.28. The residual is shown to have an unstructured spatial distribution and does not display any structure of the brain.

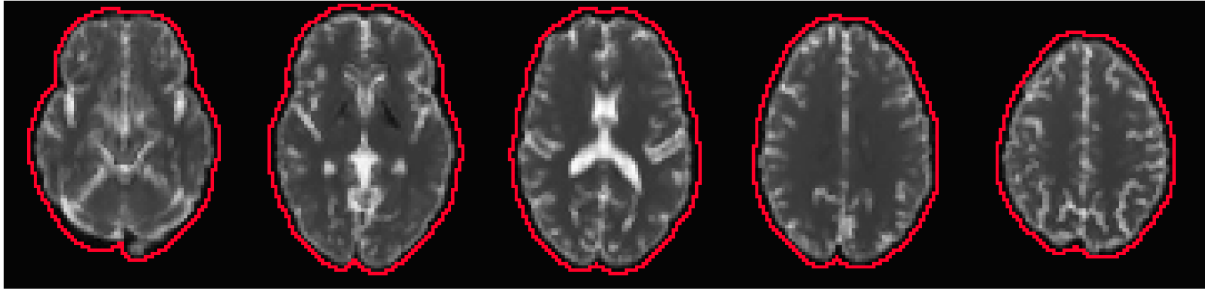


Figure 2.23: Brain extraction of the *Custom data*. The brain segmentation obtained to compute the mask is shown in red.

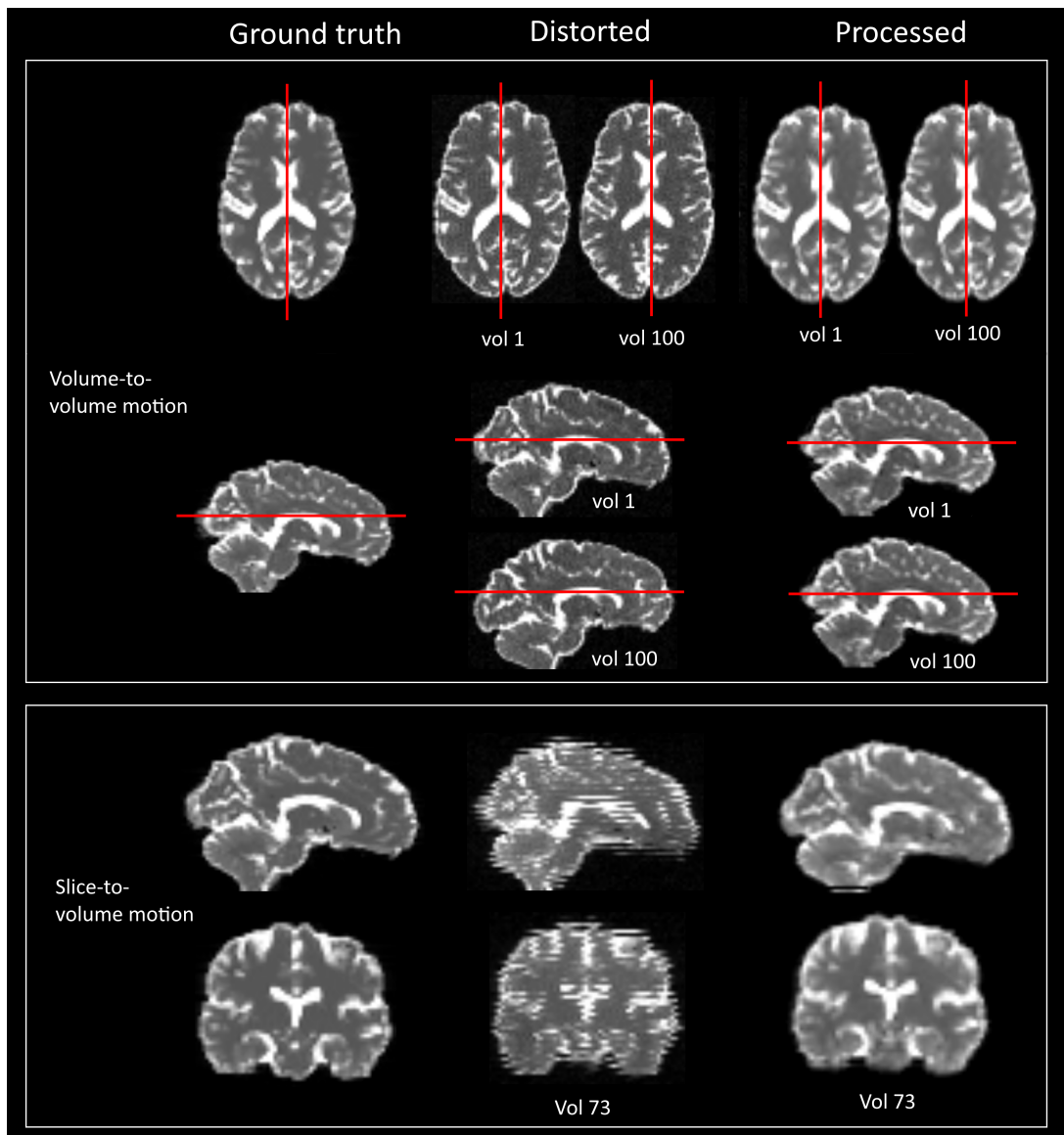


Figure 2.24: Visual inspection of the volume-to-volume and slice-to-volume motion artifacts using the *Possum* and *Custom data*. From left to right are represented the ground truth, the images in the distorted space and the images after processing with the ElikoPy pipeline. The red bars help to better observe the motion in the distorted space between the two volumes.

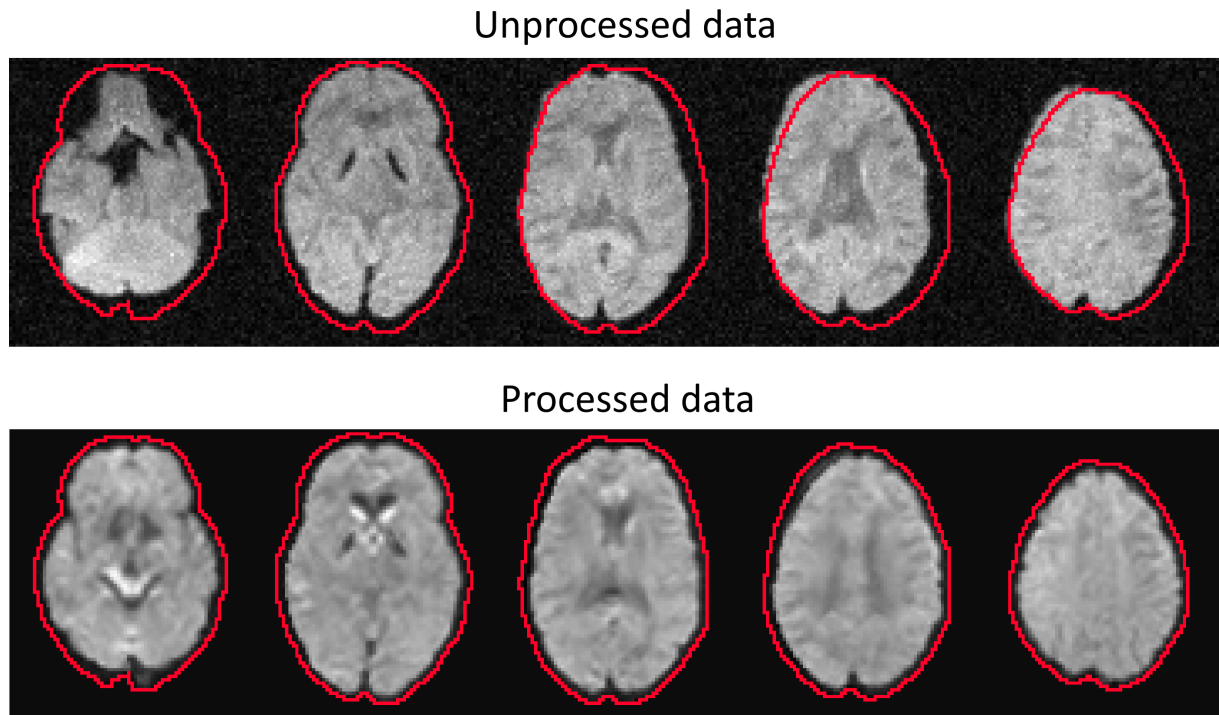


Figure 2.25: Visual results of the ElikoPy library processing for the correction of the slice-to-volume motion. Five slices of the 12th volume of the *Custom data* are shown before and after processing and are overlaid by the brain mask of the *GT data* (in red).

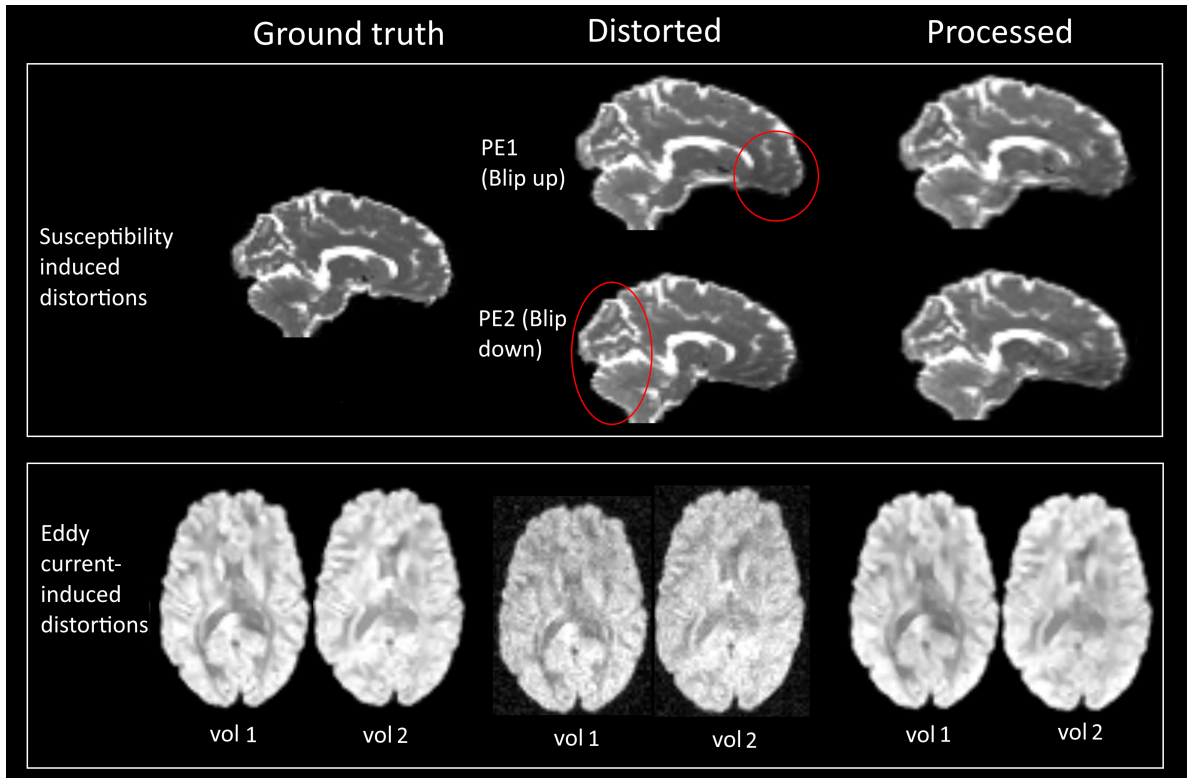


Figure 2.26: Visual results of the ElikoPy library processing for the correction of the susceptibility off resonance field and Eddy current induced distortions. For the susceptibility distortions, the results are shown for volumes with reverse phase encoding directions.

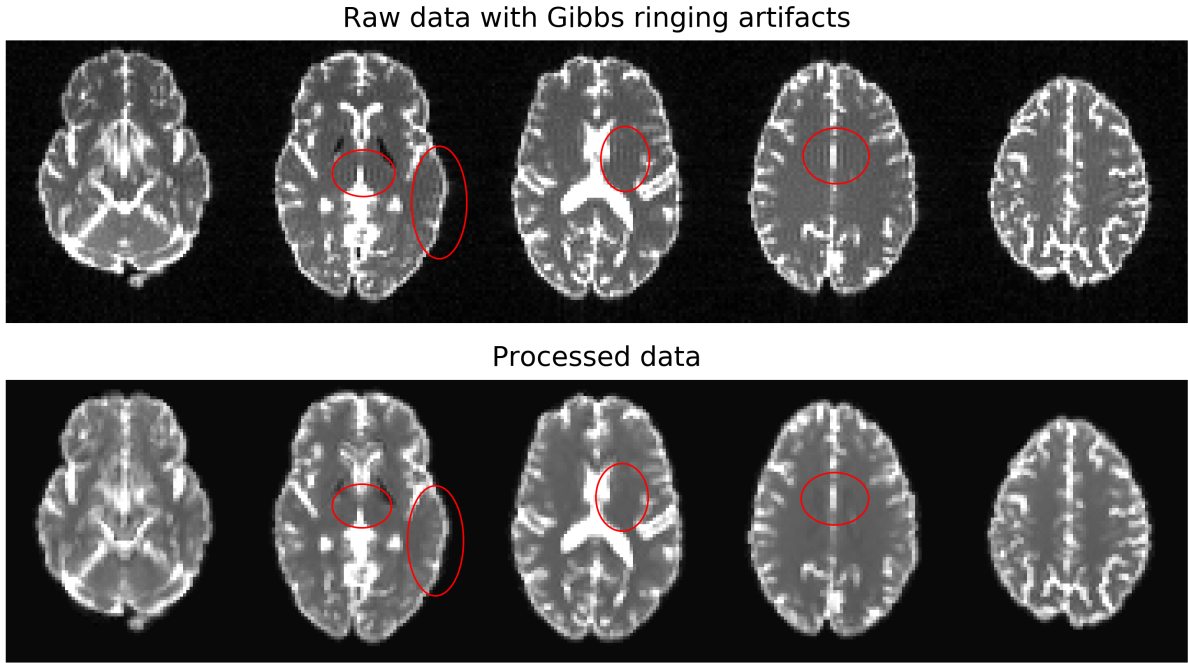


Figure 2.27: Comparison of the Gibbs ringing artifacts affecting the raw data and the data processed using the ElikoPy pipeline.

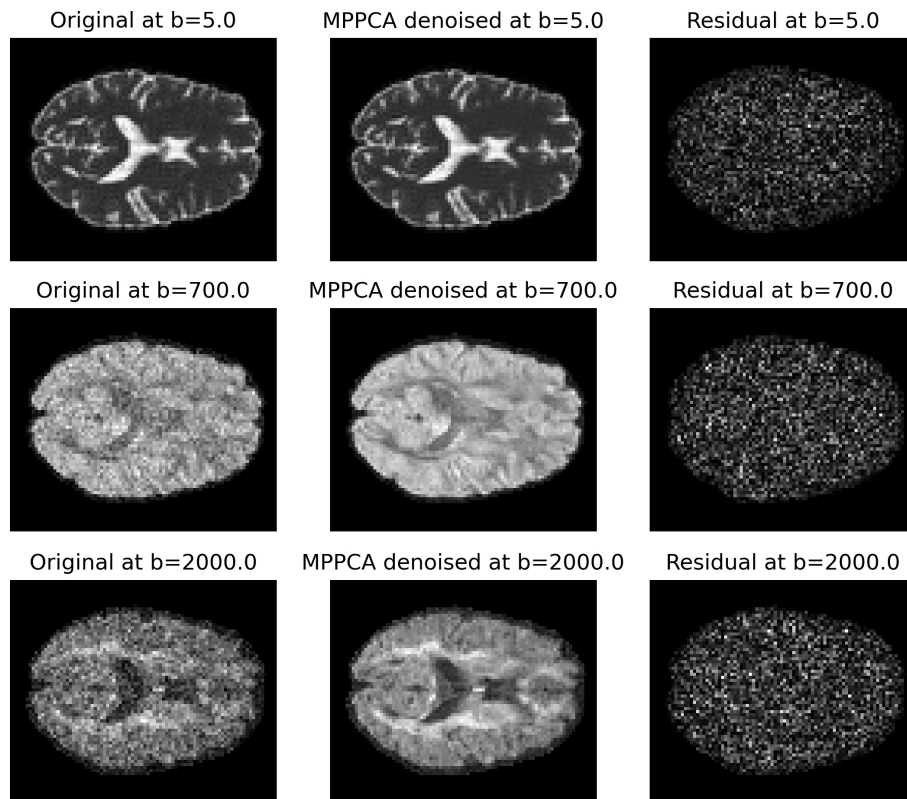


Figure 2.28: Marchenko-Pastur PCA denoising effect on the shells of the *Possum* data. The raw data is shown in the left, the processed data in the middle and the difference in the right. An unstructured spatial distribution of the residual indicates the extraction of random thermal noise.

Discussion : The results have demonstrated the ability of the pipeline to correct for the motion and the distortions artifacts. The preprocessing allowed to recover volumes similar to the ground truth images from the distorted data and to realign the images. The slice-to-volume motion correction resulted in volumes slightly blurred compared to the ground truth due to the residual motion after correction as well as the extensive use of the Gaussian Process for the reconstruction. The Gaussian Process has been explained in section 2.2.2.5 and can be seen (in a very simplified manner) as an averaging operator along the gradient directions. It is used during the motion correction in order to replace missing signal during the interpolation step. Considering the large slice-to-volume motion exhibited by the *Custom data*, the correction had to use the Gaussian process extensively to fill the missing signal points resulting in a slightly blurred output.

The brain extraction does correctly segment the brain using the Otsu method and the Gibbs ringing is effectively suppressed by resampling the data. Finally the MPPCA residuals are not structured which indicates that the noise has been properly reduced without largely affecting the underlying signal and that the nullified principal components are noise components containing limited information. The effect of denoising is predominant at higher b-values due to their lower SNR.

2.4.2 Mean squared error and Coefficient of determination

The MSE and R2 maps of four diffusion models (DTI, NODDI, DIAMOND and MF) have been computed for each in vivo data with and without processing using equations 2.19. The values have been averaged for each subject along the gray and white matter. The results for the R2 metric along the gray and white matter are reported in Fig.2.29 and 2.30 respectively while the results for the MSE metric along the gray and white matter are reported in Fig.2.31 and 2.32. One can observe a general increase in R2 on the whole brain as well as a decrease of the MSE for the processed subjects compared to the unprocessed ones. The Welch's t-test t and p-values corresponding to the boxplots are reported in Tab.2.2 showing significant differences in all cases.

For visualization, in Fig.2.33 and 2.34 are represented the MSE and R2 maps of the subject 43DYS from the in vivo dataset.

R2		DTI	NODDI	DIAMOND	MF
Gray matter	t-values	3.19	3.89	2.84	3.7287
	p-values	2.23e-03	2.41e-04	5.98e-03	4.23e-04
White matter	t-values	4.41	5.70	5.06	4.89
	p-values	4.89e-05	4.64e-07	3.64e-06	8.44e-06
MSE		DTI	NODDI	DIAMOND	MF
Gray matter	t-values	-2.23	0.50	-4.24	-4.10
	p-values	2.98e-02	6.21e-01	9.83e-05	1.50e-04
White matter	t-values	-2.14	-5.87	-5.38	-5.27
	p-values	3.58e-02	2.80e-07	1.80e-06	2.53e-06

Table 2.2: Welch's t-test comparison results between the processed and unprocessed data for the R2 and MSE in the gray and white matter.

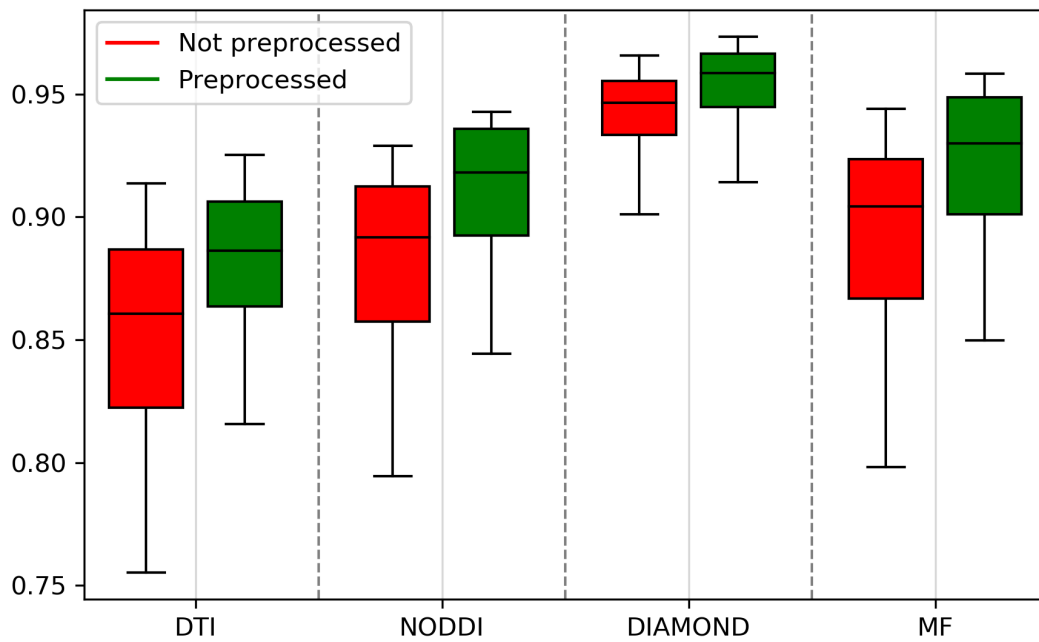


Figure 2.29: Boxplots of the coefficient of determination (R^2) metric averaged over the **gray matter** for each in vivo data before (in red) and after (in green) preprocessing. The results are displayed for four different diffusion models : DTI, NODDI, DIAMOND and MF.

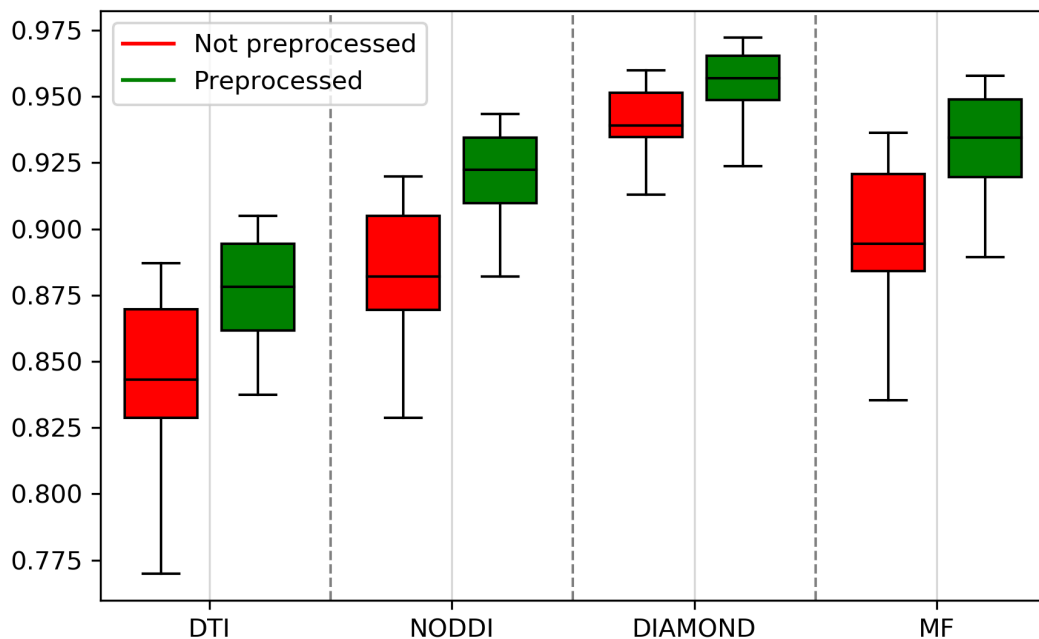


Figure 2.30: Boxplots of the coefficient of determination (R^2) metric averaged over the **white matter** for each in vivo data before (in red) and after (in green) preprocessing. The results are displayed for four different diffusion models : DTI, NODDI, DIAMOND and MF.

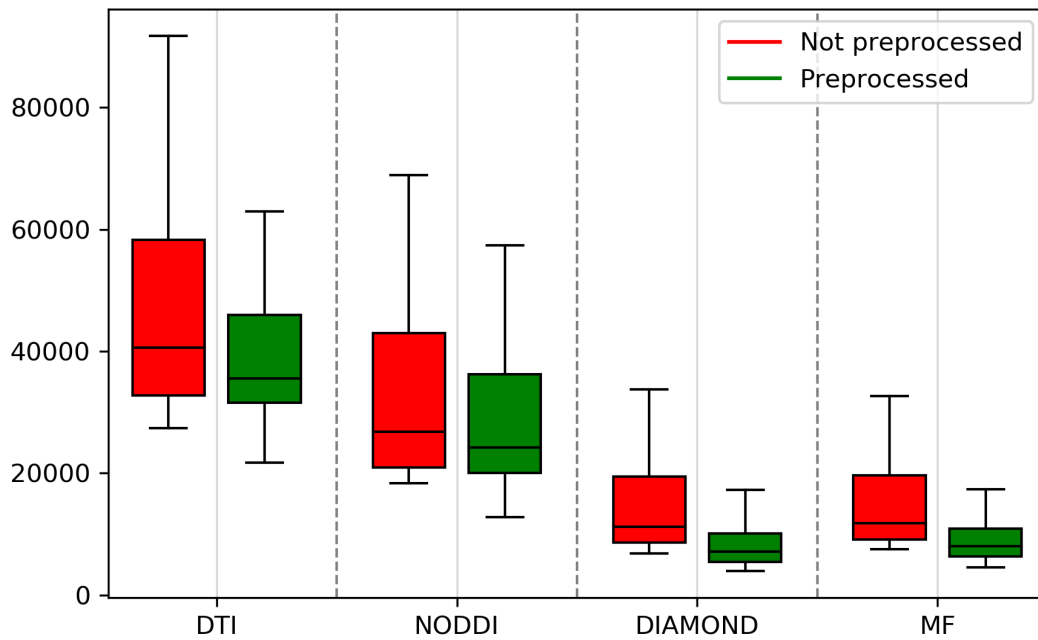


Figure 2.31: Boxplots of the mean square error (MSE) metric averaged over the **gray matter** for each in vivo data before (in red) and after (in green) preprocessing. The results are displayed for four different diffusion models : DTI, NODDI, DIAMOND and MF.

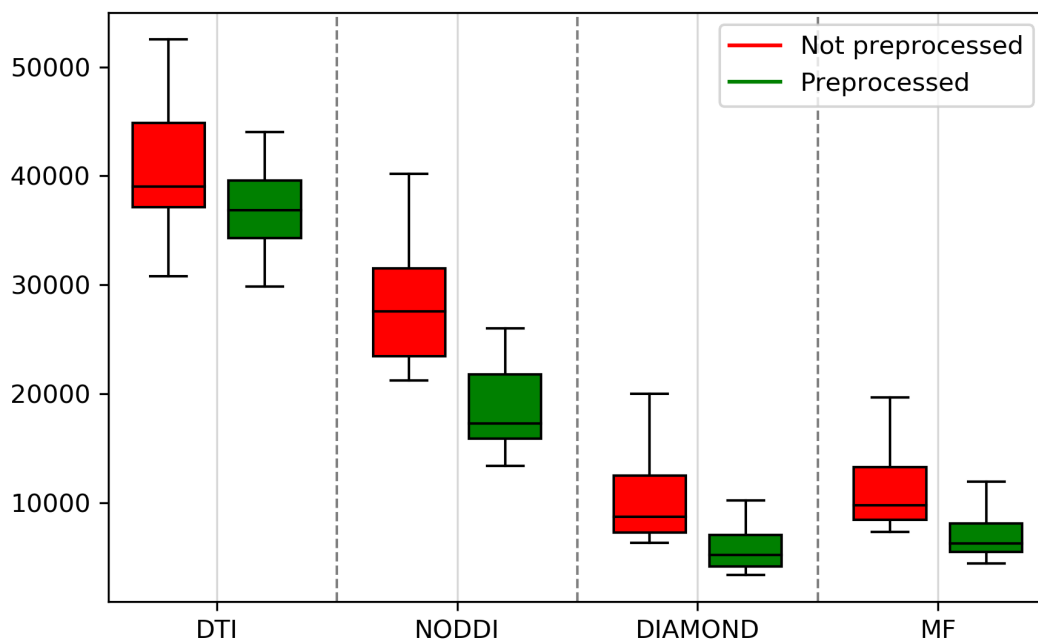


Figure 2.32: Boxplots of the mean square error (MSE) metric averaged over the **white matter** for each in vivo data before (in red) and after (in green) preprocessing. The results are displayed for four different diffusion models : DTI, NODDI, DIAMOND and MF.

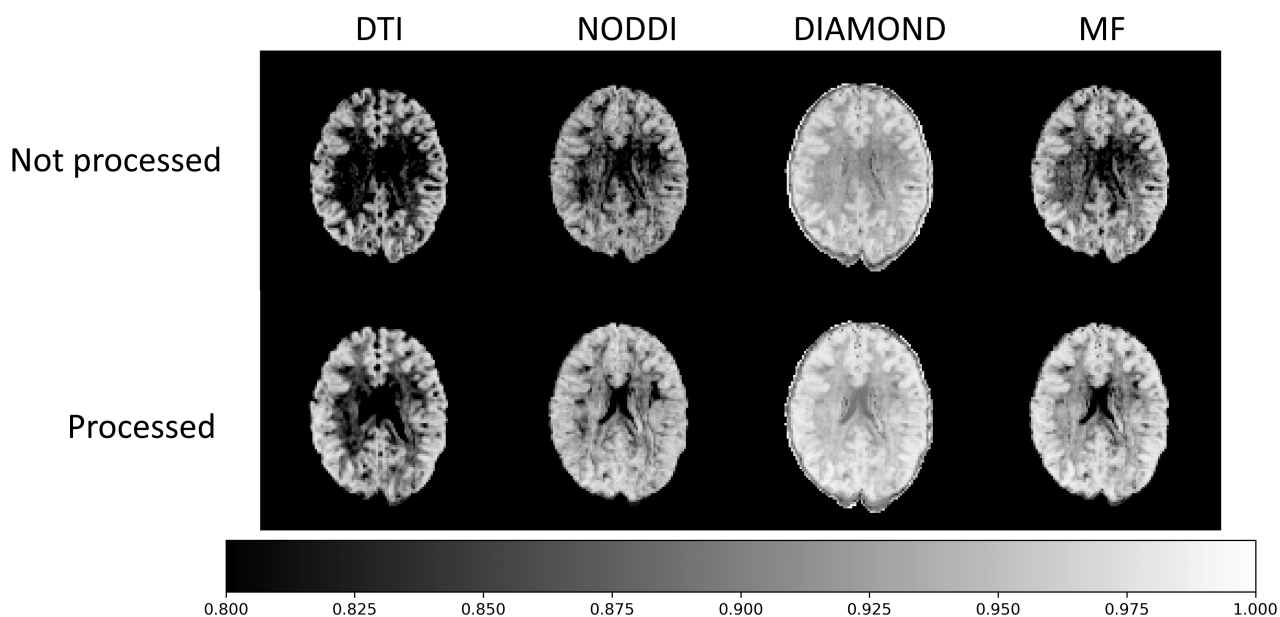


Figure 2.33: Coefficient of determination maps (R^2) of the subject 43DYS for the diffusion models DTI, NODDI, DIAMOND and MF before and after preprocessing.

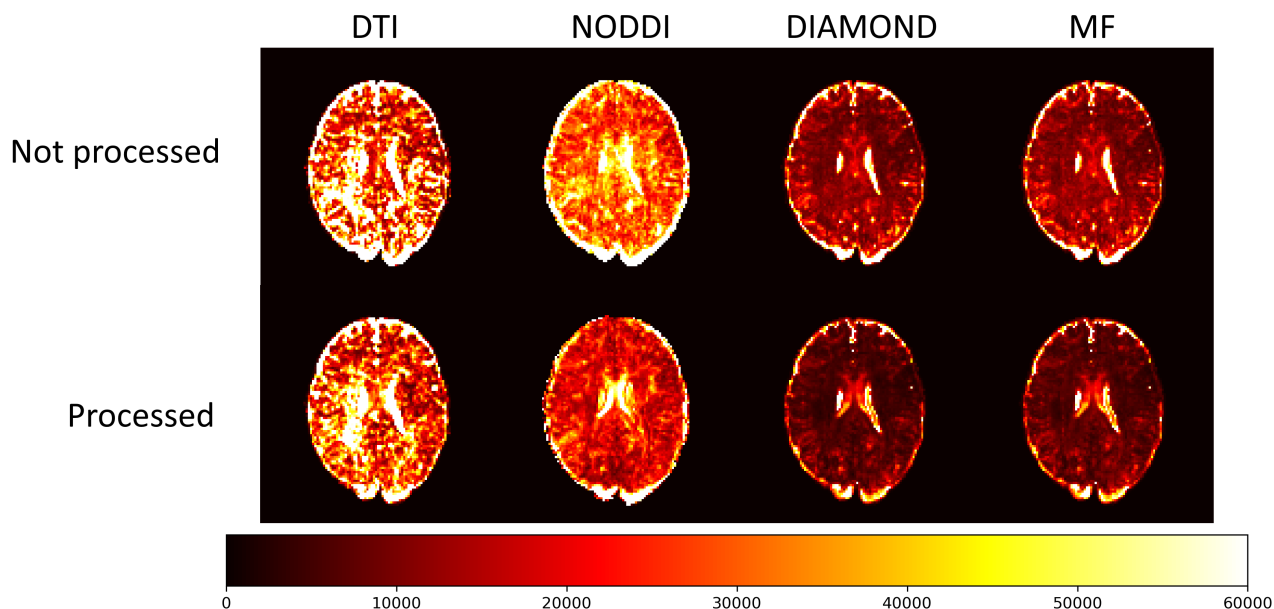


Figure 2.34: Mean squared error maps of the subject 43DYS for the diffusion models DTI, NODDI, DIAMOND and MF before and after preprocessing.

Discussion : As explained in section 2.3.2.1, the MSE and R2 are goodness of fit measures and indicate how well the diffusion models capture the information of the data. In MSE the lower is better, while R2 has an upper bound of one, which represents a perfect model fit. Overall, the decrease in MSE and increase in R2 showed significant improvements on the dataset through processing and the comparison demonstrated that the proposed ElikoPy pipeline yields to a substantially better fitting of the diffusion models compared to the raw data. It can be noted that the improvements are particularly visible in the white matter which is the region of interest for the microstructural diffusion models. This suggests that processing could increase the sensitivity of the models to better characterize the microstructure of the brain. This is supported by the fact that the more advanced multicompartment diffusion models display larger performances increase which could show that they capture finer microstructural elements and changes. The R2 and MSE also see their standard deviation across subjects decrease with processing. This reveals that the corrections properly remove the artifacts and effectively make the dataset more uniform by getting rid of the non biological variabilities across volumes.

2.4.3 Diffusion metric group statistics

The results of the ROI-based analysis for the FA (DTI), ODI (NODDI), kappa (DIAMOND) and fvf_{tot} (MF) metrics are represented in Fig.2.35, Fig.2.36, Fig.2.37 and Fig.2.38. The diffusion metrics values are reported region wise using the framework of the ElikoPy library for statistics explained in section 2.2.5. The atlas used is the JHU DTI-based probabilistic white matter atlas of FSL.

The ROI-based analysis revealed an increase in FA, kappa and fvf_{tot} as well as a decrease in ODI along all regions of the white matter compared to the unprocessed dataset. The Welch's t-test t-values are reported in Tab.2.3 showing significant differences for each region and diffusion metric.

In addition to the region wise results, voxel wise t statistics maps have been computed for each of the four diffusion metrics discussed, using the randomise function of ElikoPy (see section 2.2.5). The number of permutations was set to 10000 and the results have been FWE-corrected. The p-value maps are shown in Fig.2.39. It can be observed that significant statistical differences between the processed and unprocessed data are obtained for each metric in the white matter while the gray matter did not revealed large statistical differences.

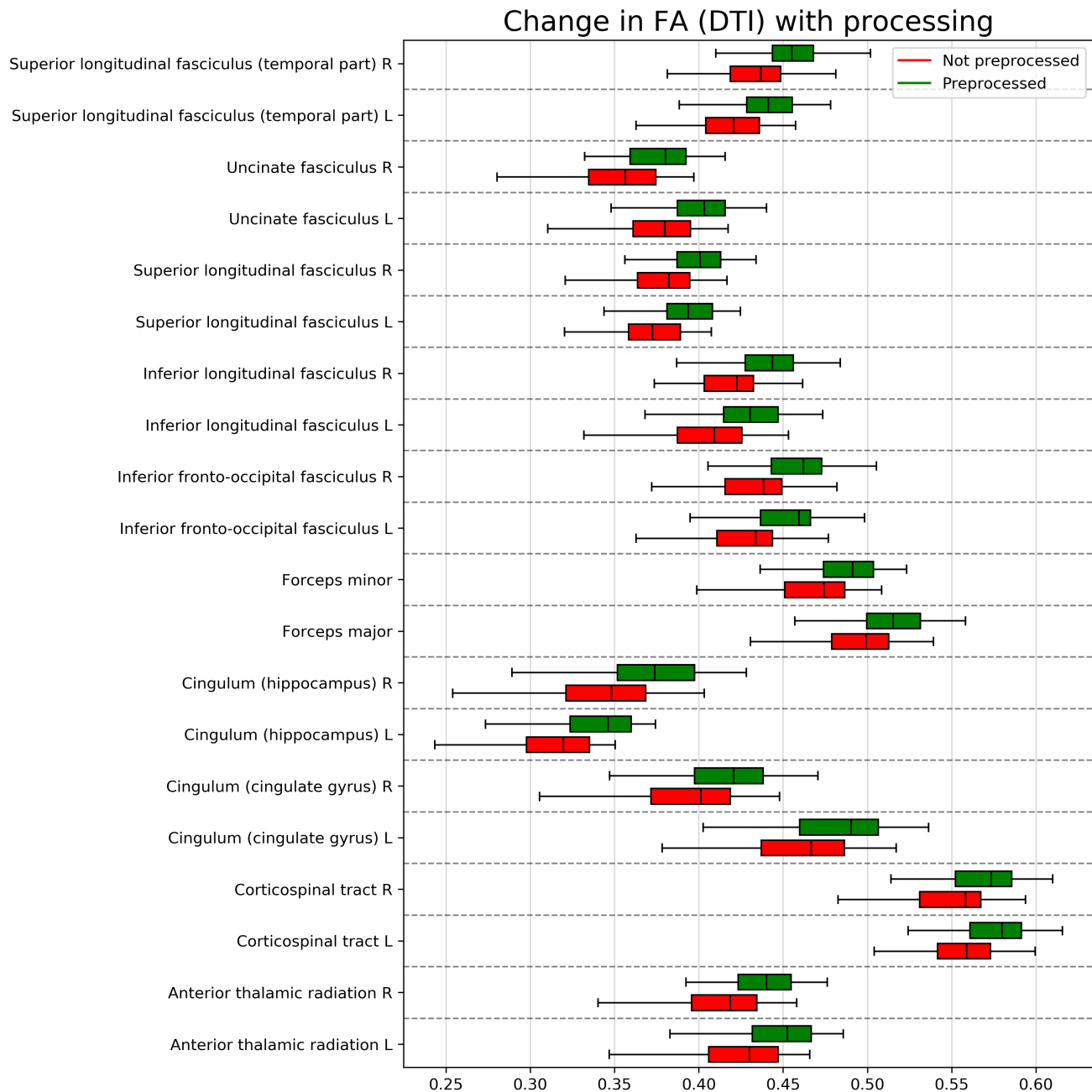


Figure 2.35: Boxplots of the white matter Fractional Anisotropy for the processed (green) and unprocessed (red) data. The values are reported along the regions of the JHU DTI-based probabilist white matter atlas.

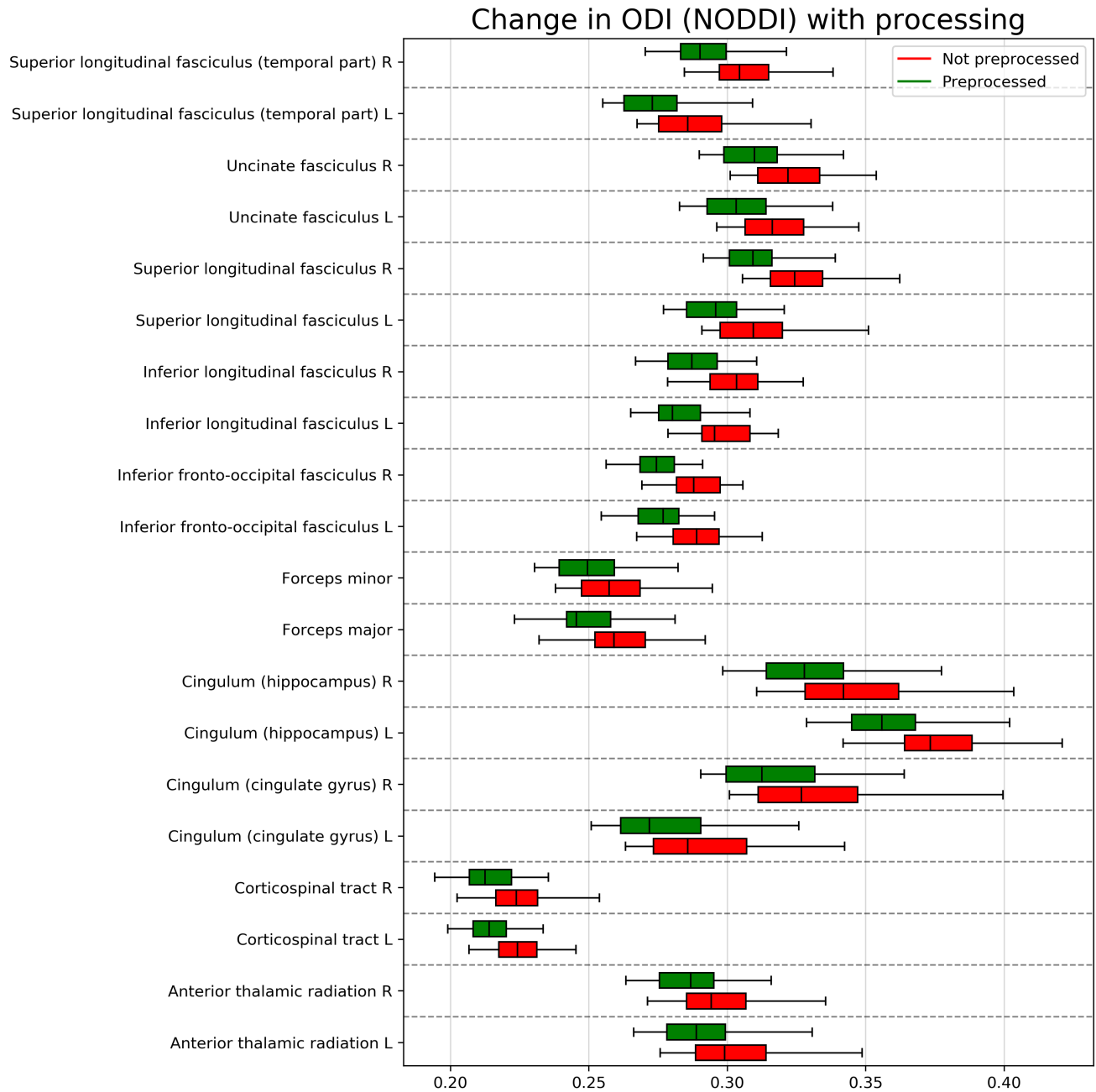


Figure 2.36: Boxplots of the white matter Orientation Dispersion Index for the processed (green) and unprocessed (red) data. The values are reported along the regions of the JHU DTI-based probabilist white matter atlas.

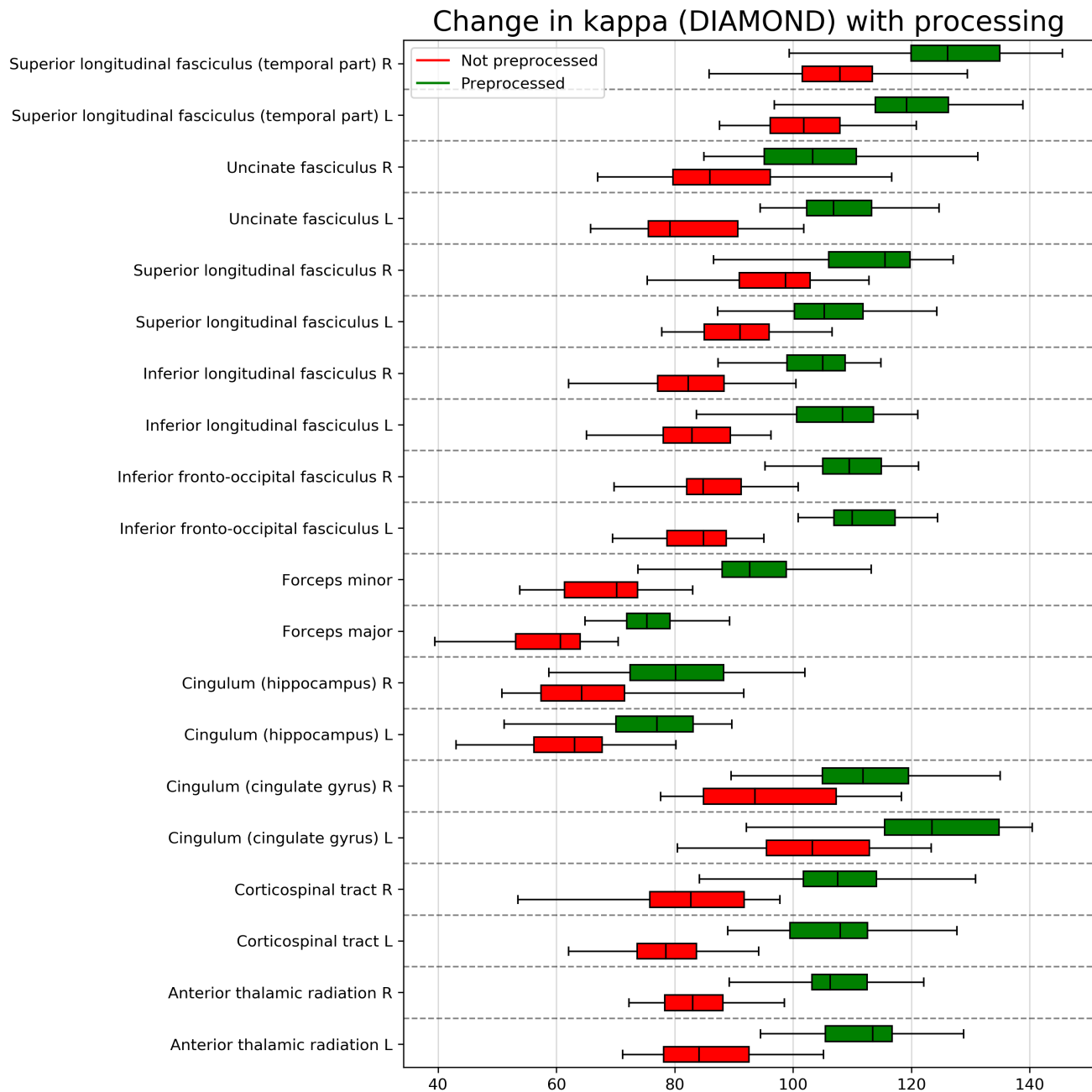


Figure 2.37: Boxplots of the white matter kappa value for the processed (green) and unprocessed (red) data. The values are reported along the regions of the JHU DTI-based probabilist white matter atlas.

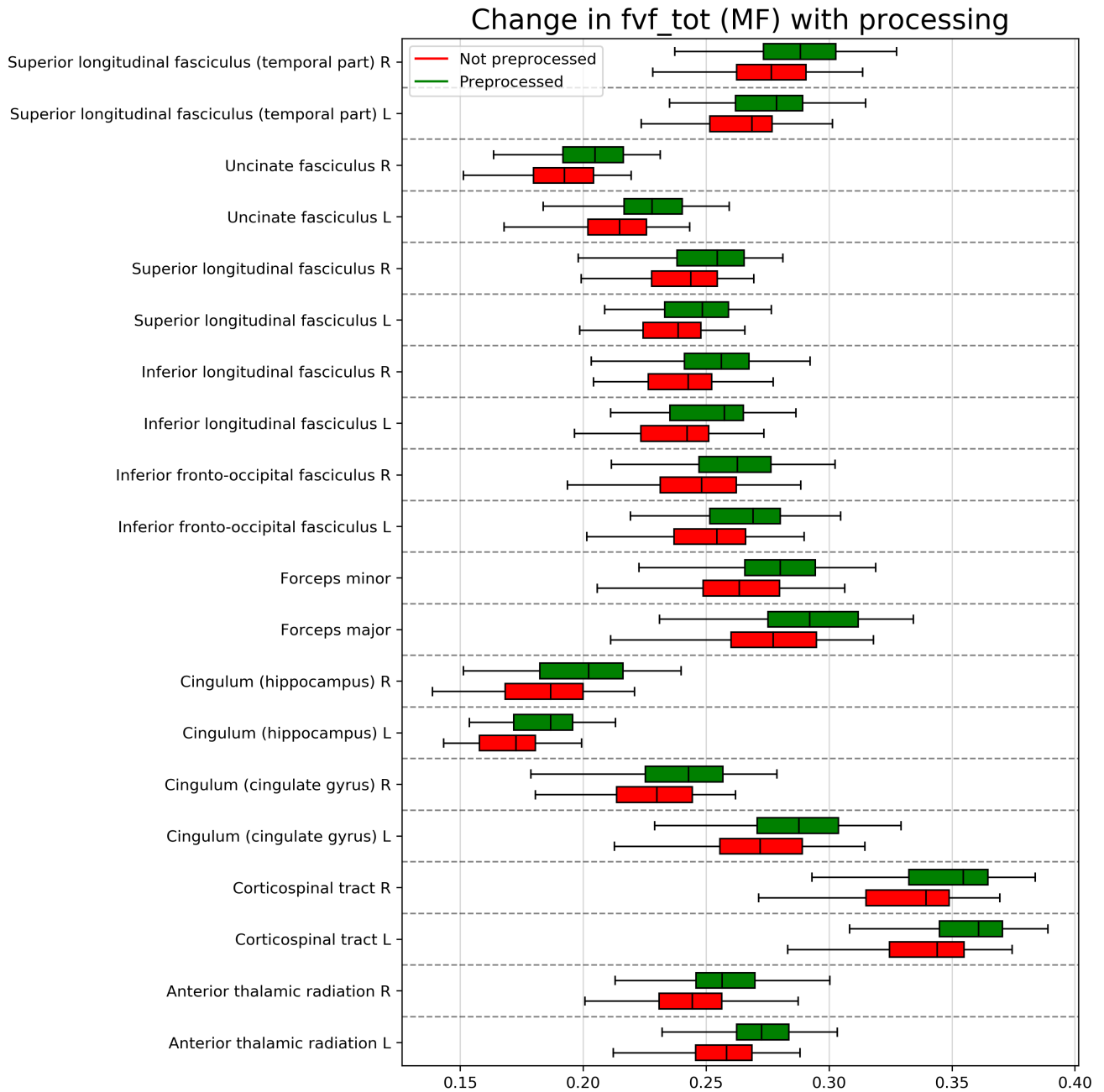


Figure 2.38: Boxplots of the white matter total fiber volume fraction for the processed (green) and unprocessed (red) data. The values are reported along the regions of the JHU DTI-based probabilistic white matter atlas.

Region wise t-values	FA	ODI	kappa	fvf_tot
Anterior thalamic radiation L	2.51	-2.29	5.91	2.33
Anterior thalamic radiation R	2.31	-1.94	5.74	2.18
Corticospinal tract L	2.45	-3.21	7.93	2.47
Corticospinal tract R	2.29	-3.18	7.28	2.27
Cingulum (cingulate gyrus) L	2.13	-2.11	5.66	2.07
Cingulum (cingulate gyrus) R	2.12	-2.06	4.62	1.82
Cingulum (hippocampus) L	3.27	-2.83	4.1	2.52
Cingulum (hippocampus) R	3.17	-2.86	4.22	2.58
Forceps major	2.31	-2.88	5.81	2.26
Forceps minor	2.16	-2.05	7.81	2.08
Inferior fronto-occipital fasciculus L	2.71	-2.96	7.59	2.29
Inferior fronto-occipital fasciculus R	2.71	-3.05	6.58	2.22
Inferior longitudinal fasciculus L	2.88	-3.48	7.31	2.29
Inferior longitudinal fasciculus R	2.82	-3.43	5.85	2.25
Superior longitudinal fasciculus L	2.52	-3.04	5.39	1.82
Superior longitudinal fasciculus R	2.36	-3.23	4.54	1.74
Uncinate fasciculus L	2.53	-2.8	6.69	2.33
Uncinate fasciculus R	2.33	-2.38	4.23	2.22
Superior longitudinal fasciculus (temporal part) L	2.56	-2.93	5.71	1.9
Superior longitudinal fasciculus (temporal part) R	2.35	-2.98	5.	1.78

Table 2.3: Welch’s t-test comparison results between the processed and unprocessed data for the FA, ODI, kappa and fvf_tot in the regions of the JHU DTI-based probabilist white matter atlas.

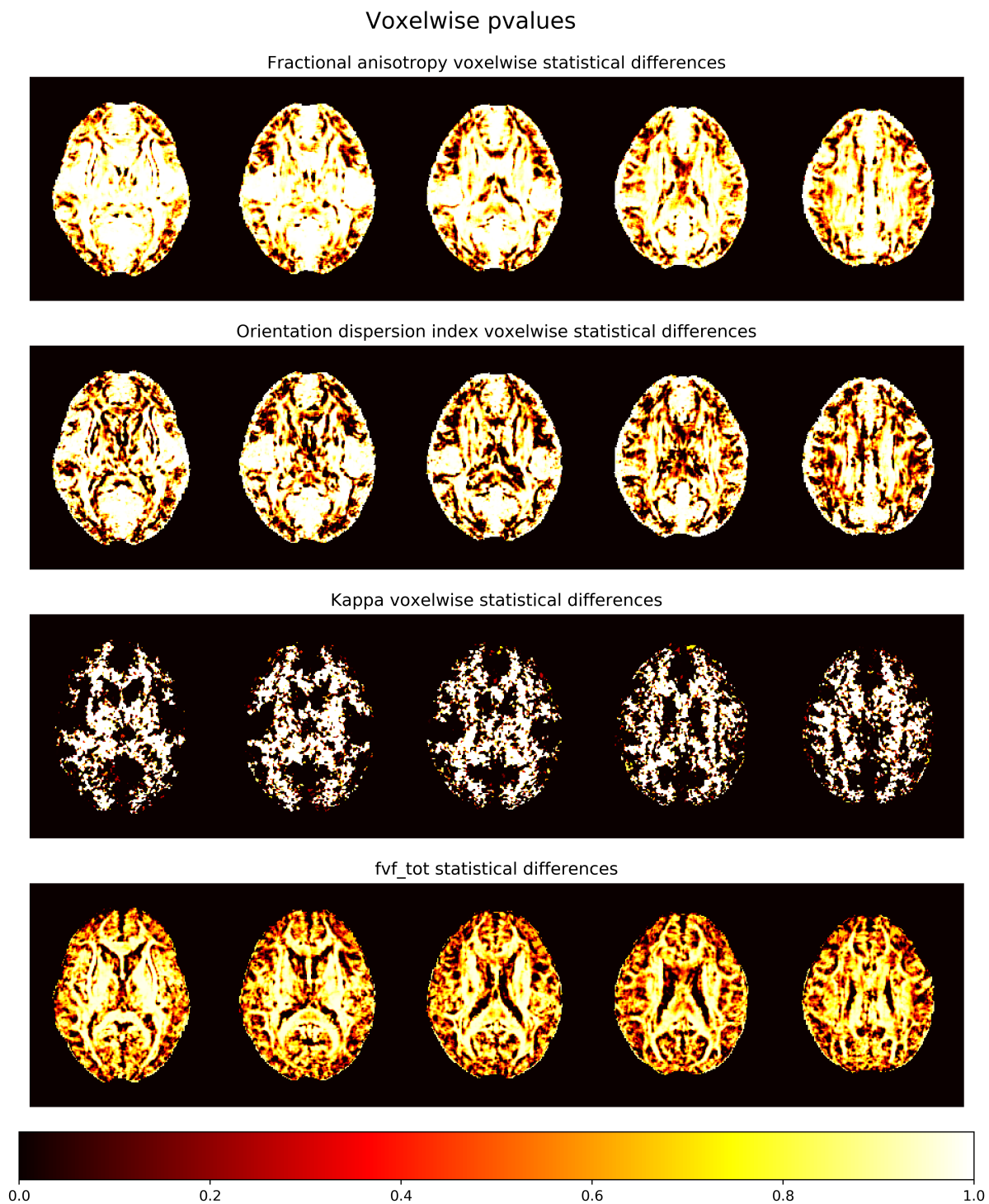


Figure 2.39: Voxel wise FWE-corrected $1-(p \text{ value})$ maps of the statistical differences between the processed and unprocessed datasets for the Fractional Anisotropy, Orientation Dispersion Index , kappa and total fiber volume fraction.

Discussion : The Fractional anisotropy characterizes the diffusion anisotropy in each voxel (see section 1.2.2). It is widely considered to be an indirect measure of the density of packing of fibers in a voxel, the amount of myelin wrapping the axons and the axonal integrity. The analysis revealed a general increase in FA in all regions of the white matter. The comparison is performed between the same subjects processed and non processed, the differences cannot therefore be attributed to biological features. Through processing, the non biological variations along the volumes which can be attributed to noise, bias field and Gibbs ringing are attenuated. These variations are independant of the diffusion gradient directions and will tend to make the signal artificially more isotropic. Thus, removing them cause the FA to increase. Moreover, the distortion and motion corrections allow to obtain better aligned voxels along the diffusion dimension of each data. This results in a better fitting of the data in the white matter by reducing the variations created by including adjacents voxels in the estimation of the metric.

Similarly these changes affect also the fiber Orientation Dispersion Index (ODI). Only this metric is a measure of the orientation dispersion of the fibers around their mean direction (see section 1.3.1). Therefore its value decreases instead of increasing with processing. The evolution of the ODI confirms the results obtained with the DTI for the NODDI model.

The kappa value is the distribution shape parameter and can be seen as a concentration/homogeneity index. The density of fibers becomes more concentrated around the mean as κ increases. The more homogeneous the compartments, the higher the κ parameter is (see section 1.3.2). This metric sees larger changes than the FA or ODI. This is due to the fact that DIAMOND is a multicompartment model that considers multiple fascicles in the voxels contrary to DTI and NODDI. Therefore, in addition to the benefits discussed above, the processing possibly allows to better differentiate crossing fascicles in the voxels. This highlights the interest of processing especially for more advanced diffusion models. It greatly increases the sensitivity to small variations and allow to better discriminate the fiber populations inside each voxel.

This explains also the evolution of the total fiber volume fraction, fvf_{tot} , which represents the total fraction of water inside the axons of each voxel (computed using the fiber volume fraction and fraction of occupancy of the estimated fascicles, see section 1.3.3).

The voxel wise analysis revealed significant increases for the diffusion metrics along the white matter confirming the results obtained for the ROI-analysis. The results are less significant in the gray matter due to the fact that the diffusion metrics do not have much sense in this region. The diffusion models have been designed to assess changes of microstructure in the white matter and gray matter regions see little changes in the quality of the diffusion metrics estimated due to the absence of meaning for the metrics there. This is particularly visible for the kappa κ map as DIAMOND is not designed to estimate outside the white matter.

Overall, the changes in diffusion metrics are significant along the whole white matter for each diffusion model. These results highlights the benefits of processing on the diffusion metrics. The non biological effects are reduced suggesting a potential gain in sensitivity to infer biological changes from the diffusion models.

2.4.4 Ground truth diffusion metric comparisons

From the simulated *GT data*, the ground truth DTI diffusion metrics have been computed and compared with the diffusion metrics obtained from the unprocessed data as well as the data processed with DESIGNER, PreQual and ElikoPy. The FA, MD and RD (see section 1.2.2) metrics are displayed in Fig.2.40 and the errors made on the metric compared to the ground truth are shown in Fig.2.41. The other diffusion models (i.e. NODDI, DIAMOND and MF) have been omitted during this analysis due to their unaffordable processing time when writing this report.

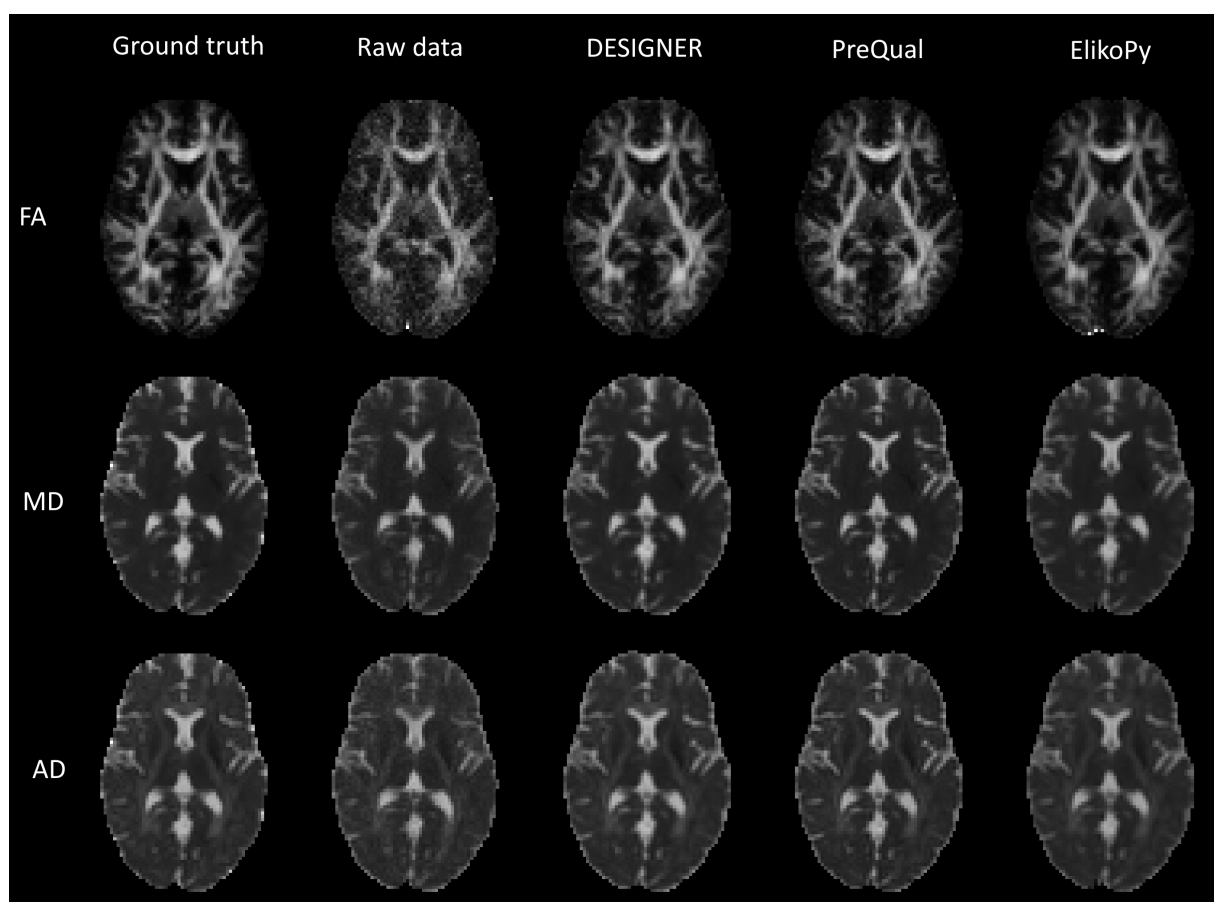


Figure 2.40: Fractional Anisotropy, Mean Diffusivity and Axial Diffusivity (DTI) computed for the *GT data*, the unprocessed image and the data processed using DESIGNER, PreQual and ElikoPy.

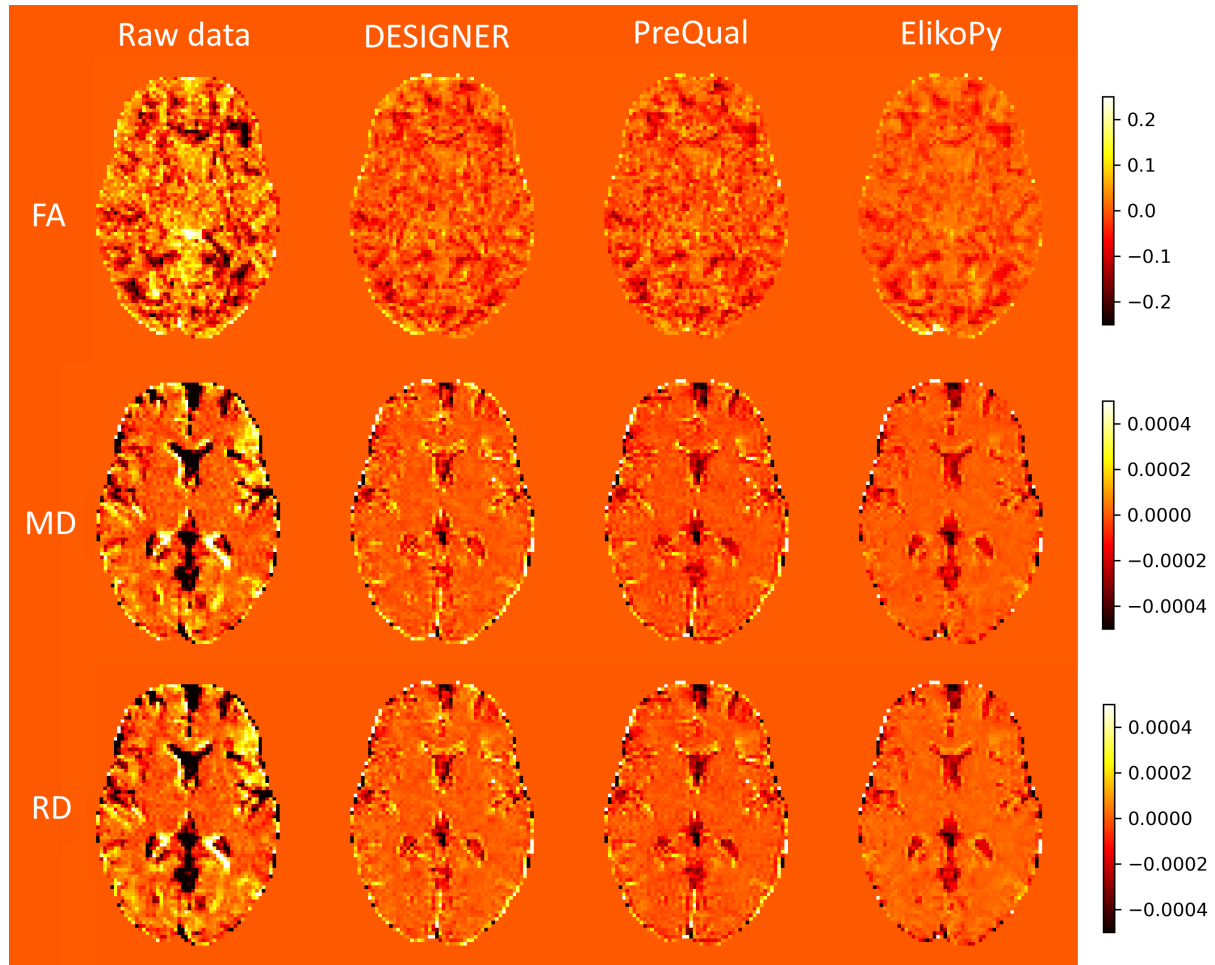


Figure 2.41: Errors on the diffusion metrics : Fractional Anisotropy, Mean Diffusivity and Radial Diffusivity (DTI). The results are given for the unprocessed image and the data processed using DESIGNER, PreQual and ElikoPy.

Discussion : Overall, all pipelines manage to correct for the large errors affecting the raw data. However, it can be noted that the ElikoPy library seems to provide slightly better Gibbs ringing and noise removal with results more uniform across the white and gray matter. The CSF regions are also more marked suggesting that the distortions and motion artifacts are better corrected since the CSF signal does not interfere with the brain tissues. For all pipelines, the diffusivity is underestimated in the CSF. Further investigations would be required to see if the observed trends confirm for the other diffusion models.

2.4.5 Temporal Signal-to-Noise Ratio

Temporal Signal-to-Noise Ratio (tSNR) maps have been computed for each in vivo data and the values have been averaged over the white and gray matter. The results are reported in Fig.2.42 and 2.43. One can observe a general increase in tSNR in both cases for the processed data compared to the unprocessed subjects. This difference gets more predominant at larger b-values. The Welch's t-test t and p-values are reported in Tab.2.4 showing significant differences in all cases. In Fig.2.44 and 2.45 are shown the tSNR changes of one subject for the $1000[s/mm^2]$ and the $3000[s/mm^2]$ shells respectively.

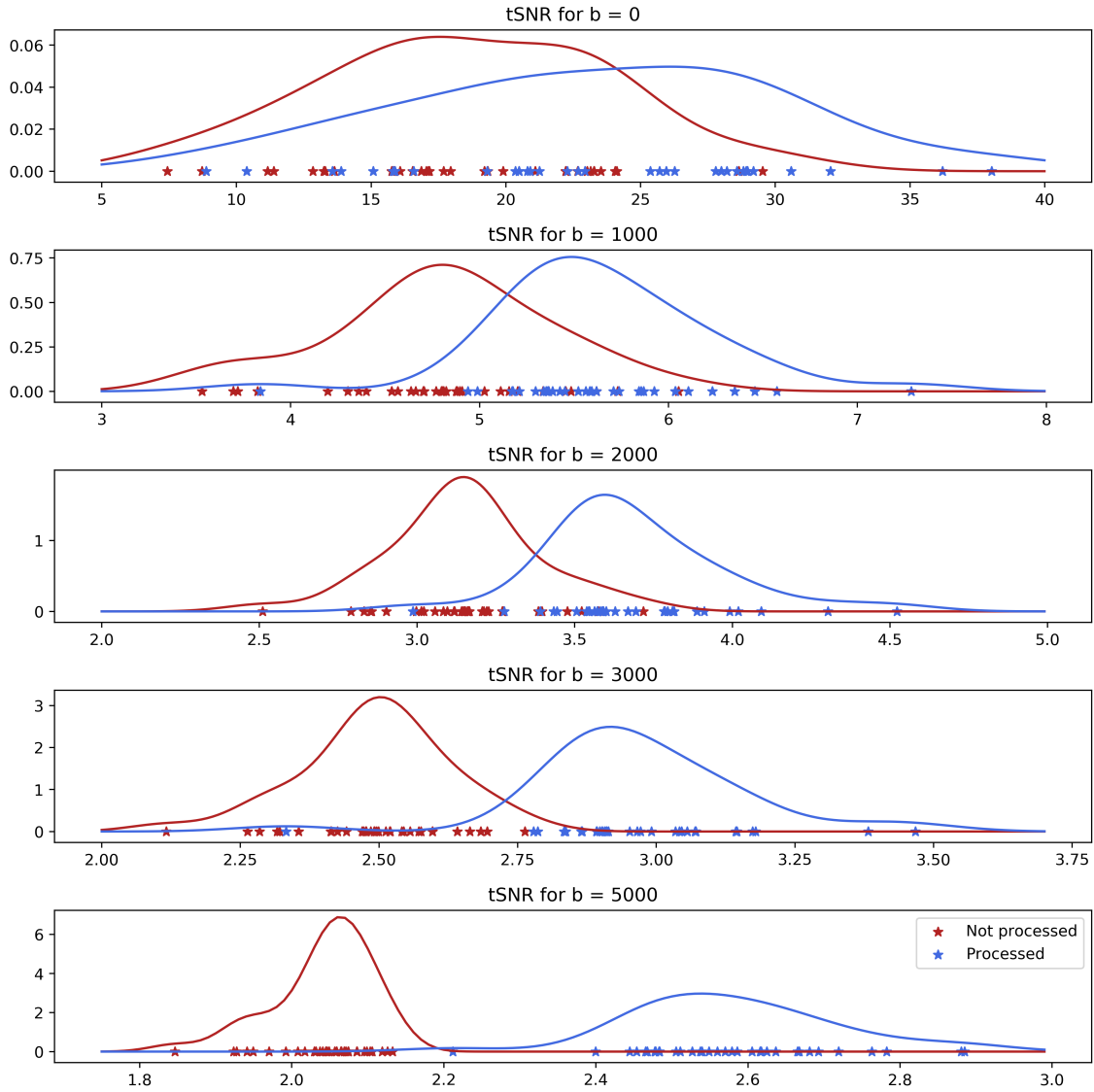


Figure 2.42: Gray matter changes in tSNR between the processed data in blue and the unprocessed one in red. The exact values are represented using stars and probability density function estimates are represented with solid lines. The PDF estimates have been computed for visual purposes using Kernel-density estimation from Gaussian kernels (Scott, 2015).

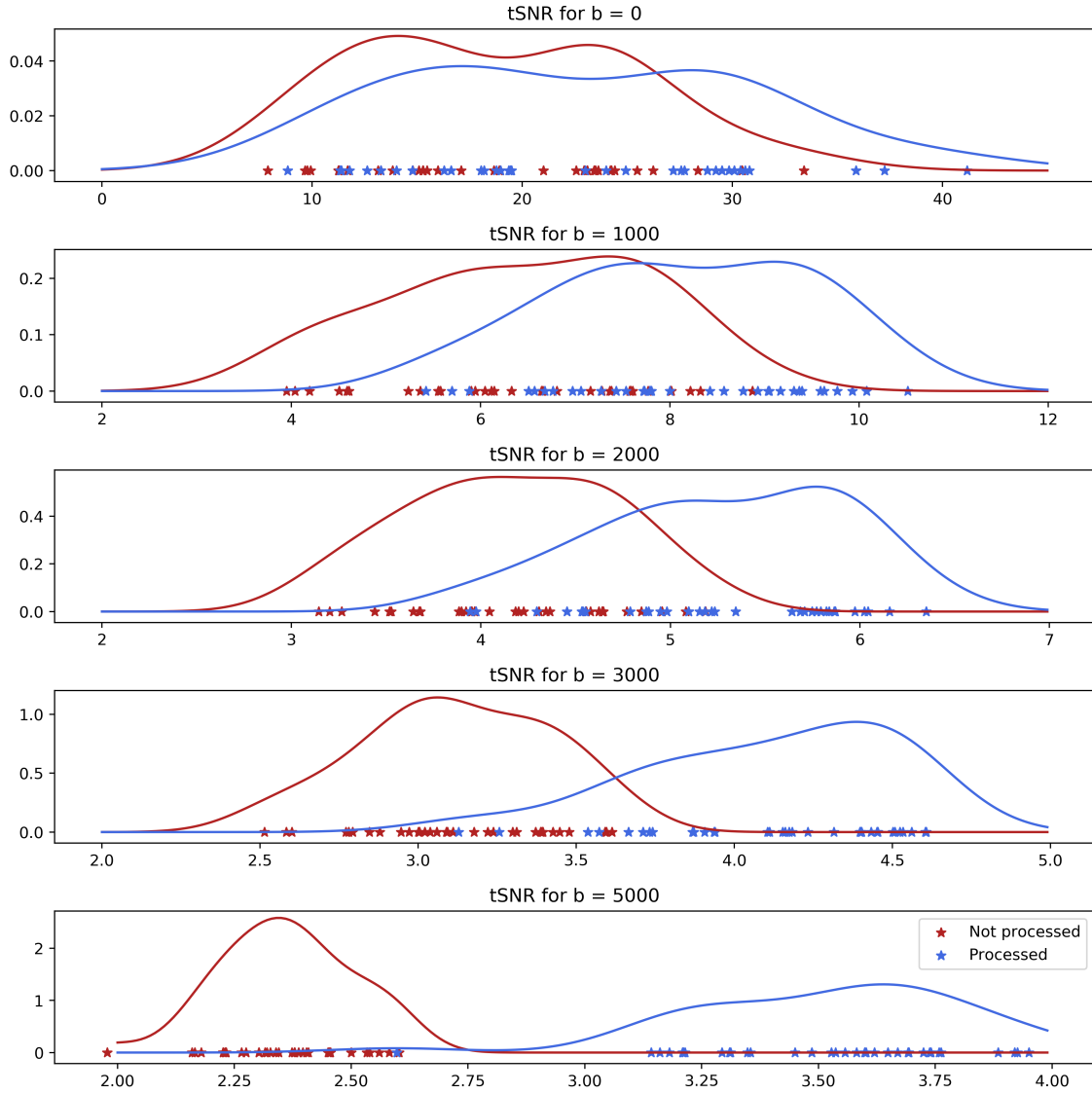


Figure 2.43: White matter changes in tSNR between the processed data in blue and the unprocessed one in red. The exact values are represented using stars and probability density function estimates are represented with solid lines. The estimation have been realized for visual purposes using Kernel-density estimation from Gaussian kernels (Scott, 2015).

Gray Matter, b	$0[s/mm^2]$	$1000[s/mm^2]$	$2000[s/mm^2]$	$3000[s/mm^2]$	$5000[s/mm^2]$
t-values	2.19	5.15	7.65	11.63	20.89
p-values	3.22e-02	2.53e-06	1.26e-10	3.91e-17	7.91e-26
White Matter, b	$0[s/mm^2]$	$1000[s/mm^2]$	$2000[s/mm^2]$	$3000[s/mm^2]$	$5000[s/mm^2]$
t-values	3.16	5.93	8.43	11.90	20.81
p-values	2.43e-03	1.20e-07	5.51e-12	2.54e-17	1.91e-25

Table 2.4: Welch's t-test comparison results between the processed and unprocessed data in the gray and white matter for each shell.

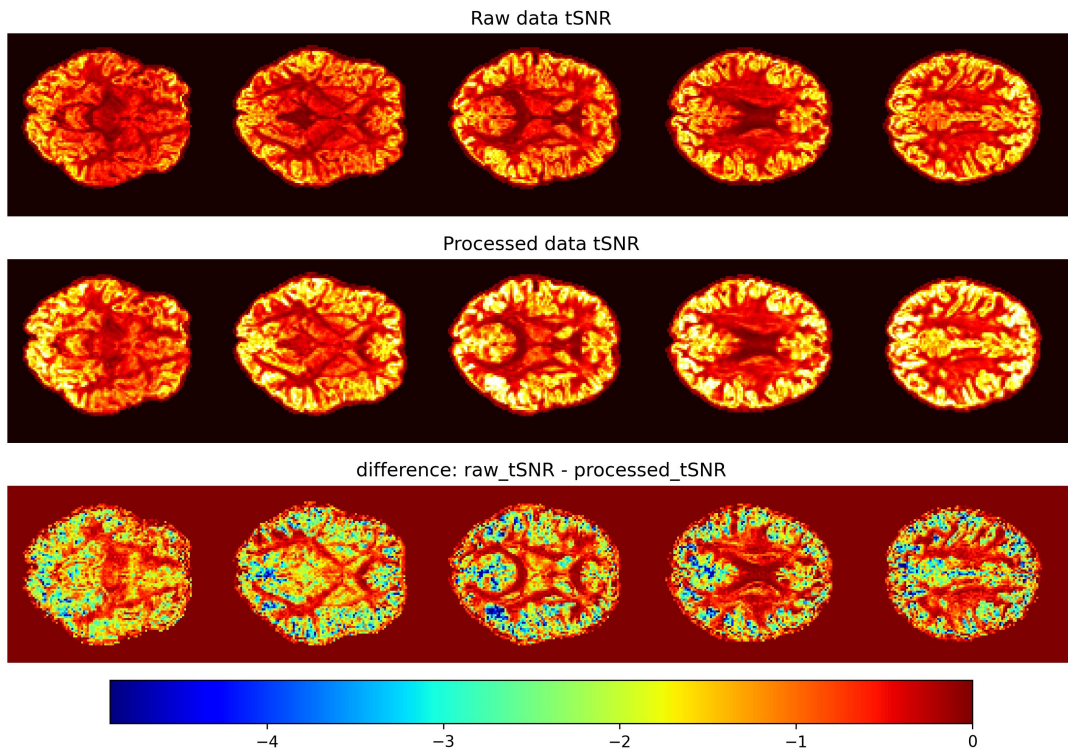


Figure 2.44: Evolution of the temporal SNR with processing for the $b = 1000[s/mm^2]$ shell.

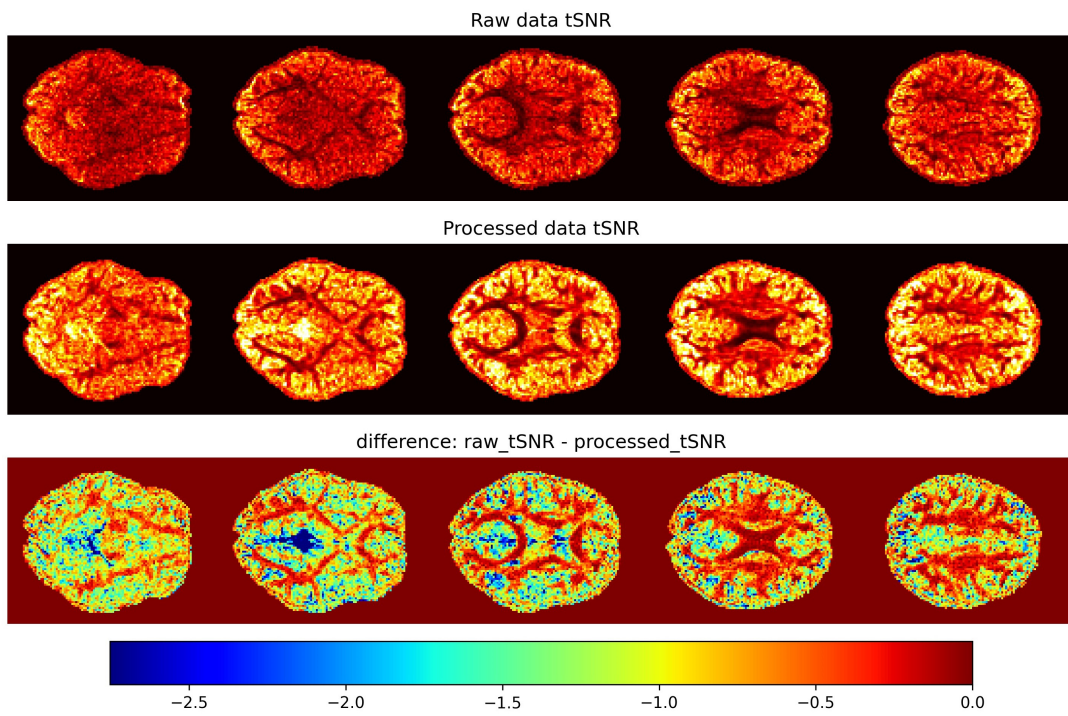


Figure 2.45: Evolution of the temporal SNR with processing for the $b = 3000[s/mm^2]$ shell.

Discussion : Overall the tSNR showed significant improvements on the dataset through processing and the comparison demonstrated that the proposed ElikoPy pipeline yield substantially higher tSNR compared to the original data. For higher b-values the improvement gets more significant. This is due to the fact that, in dMRI, the signal is attenuated by the diffusion weighting. The signal attenuation is larger at higher b-values while the noise level stays constant, effectively decreasing the SNR and making denoising more relevant.

The results shown in Fig.2.44 and 2.45 suggest that this increase in tSNR is limited to the gray matter which is of less importance in dMRI (since this MRI technique usually aims at quantifying axonal changes). However, this apparent decrease of performances in white matter is misleading. Indeed, highly anisotropic regions such as the corpus callosum, or the white matter in general, tends to have low tSNR in all individuals (Roalf et al., 2016). The tSNR is computed by calculating the average and the standard deviation of the signal across the volumes for each shell. Therefore, it makes the assumption that the volumes are repetitions of the same signal. This assumption does not hold in highly anisotropic regions since the signal is different for each volume due to the directional diffusion gradients. Nonetheless, the noise can be assumed constant over the whole brain and the improvements observed on the gray matter can be assumed to generalize also on the white matter.

2.4.6 Signal-to-Noise-Artifact Ratio

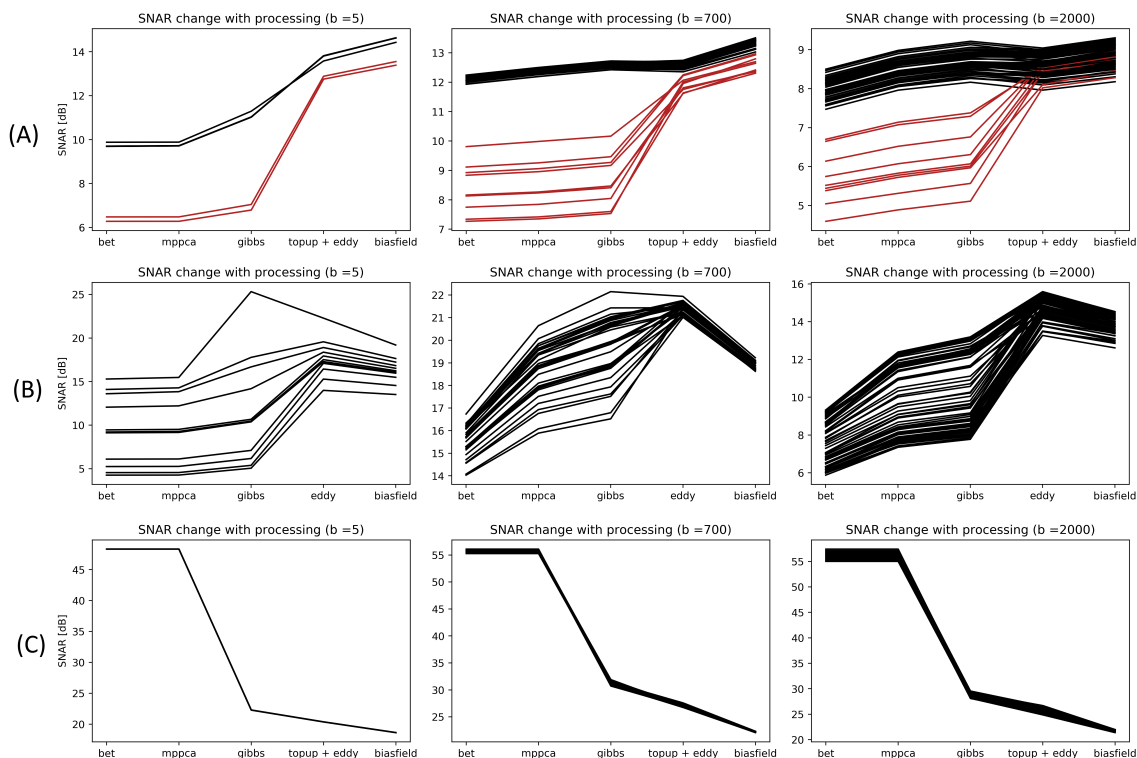


Figure 2.46: SNAR at each processing step of the ElikoPy library for the (A) *Custom data* (B) *Possum data* (C) processed *GT data*. Each solid line represents one volume of the data. The red lines are the volumes affected by slice-to-volume motion. All b-values are given in s/mm^2 .

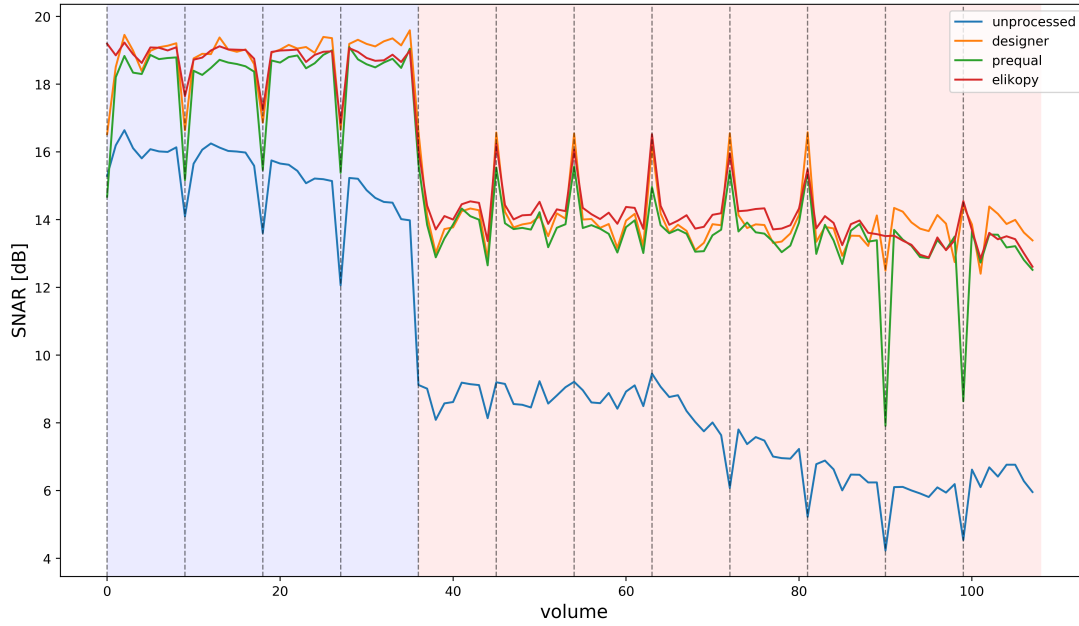


Figure 2.47: SNAR for each volume of the *Possum* data. The vertical dotted lines represent the acquisition of the b_0 volumes. The light blue area corresponds to the $b = 700[s/mm^2]$ images and the light red area corresponds to the $b = 2000[s/mm^2]$ images. The SNAR is shown for the unprocessed data (blue) as well as for the data processed with the pipelines ElikoPy (red), DESIGNER (orange), PreQual (green).

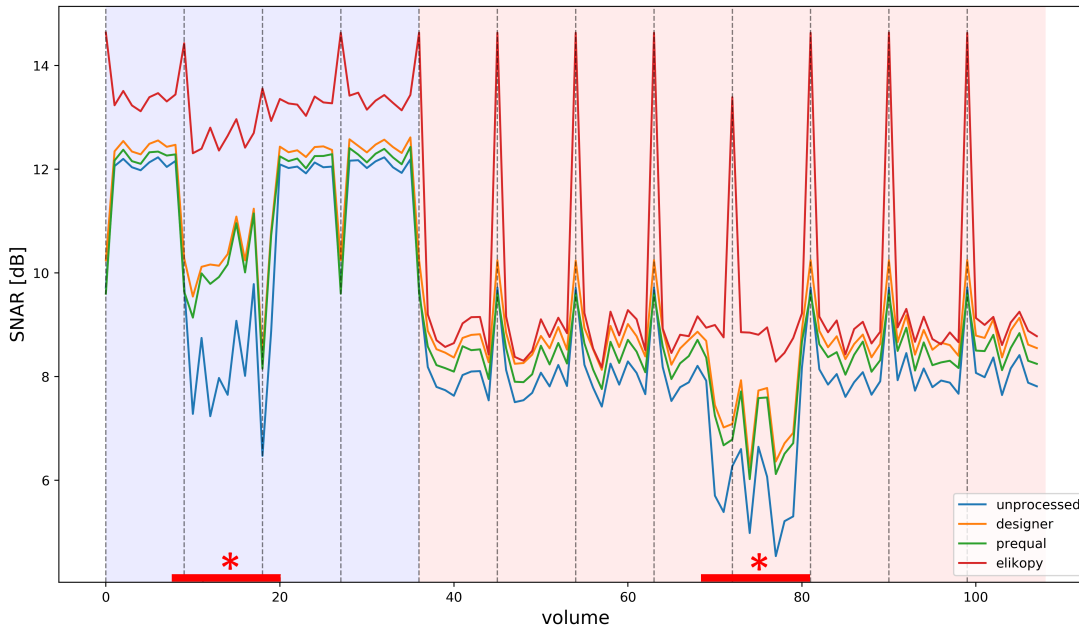


Figure 2.48: SNAR for each volume of the *Custom* data. The vertical dotted lines represent the acquisition of the b_0 volumes. The light blue area corresponds to the $b = 700[s/mm^2]$ images and the light red area corresponds to the $b = 2000[s/mm^2]$ images. The red stars highlight the volumes affected by slice-to-volume motion artifacts. The SNAR is shown for the unprocessed data (blue) as well as for the data processed with the pipelines ElikoPy (red), DESIGNER (orange), PreQual (green).

From the simulated data, the SNAR has been computed at each step for each volume of the *Custom data*, *Possum data* and processed *GT data*. The results are summarized in Fig.2.46.

The final result of ElikoPy is compared with the outputs of the DESIGNER and PreQual pipelines. For each processing pipeline the SNAR of the *Possum* and *Custom data* is reported along each volume. This is represented on Fig.2.47 and 2.48 respectively.

Discussion : From Fig.2.46 several observations and conclusions can be made on the ElikoPy processing steps.

1. the MPPCA denoising allows to increase the SNAR of the data especially at higher b-values (where the SNR is lower) without being detrimental on low noise data. In fact MPPCA does not affect the quality of the *GT data*. The MPPCA denoising especially improves the *Custom data* which is affected by large noise compared to the *Possum data*.
2. The Gibbs ringing suppression increases the SNAR on data which suffers from Gibbs ringing artifacts. However it proves to be highly harmful on data unaffected by this artifact. This is due to the fact that the interpolation procedure of the Gibbs ringing suppression induces smoothing that could blur out small details in the image. Gibbs ringing correction is therefore usually only enabled for data which suffers from this artifact.
3. Eddy and Topup from FSL perform motion correction and correct for the Eddy current and susceptibility induced distortions. This processing step has a large influence on the SNAR. It results overall in an increase of the SNAR (especially for the *Possum data*). This could suggest better aligned volumes. The images that suffer from slice-to-volume motion artifacts are almost perfectly recovered and brought back to level comparable to the other images. However, it can be observed that some volumes see their SNAR decrease during this processing step (e.g. the *Custom data* volumes unaffected by slice-to-volume motion see a SNAR decrease, despite the correction for the susceptibility induced distortions). This can be explained by the fact that Eddy FSL results in volumes aligned in a common space without distortions, but nothing guarantees that this space is strictly equivalent to the ground truth volumes, effectively decreasing the SNAR slightly. A more relevant feature to observe is that the variation of SNAR across volumes is greatly reduced and after this step, all volumes are aggregated on the same SNAR value showing a better alignment of the data along the diffusion dimension.
4. The bias field correction proves to increase the SNAR when the data suffer from this artifact. However, for the data unaffected, this step shows to be detrimental and is disabled by default.

The comparison of the ElikoPy library with the other pipelines in Fig.2.47 and 2.48 shows that ElikoPy performs slightly better than the other two. Moreover, it succeeds in recovering from slice-to-volume motion artifacts. The difference from unprocessed images is less pronounced for the *Custom data* processing due to the low noise level and motion affecting this dataset.

2.4.7 Comparison with ground truth parameters

Using the simulation ground truth parameters and the estimated parameters outputted by the pipeline, several figures have been realized. The estimated motion is compared to the

real applied motion in Fig.2.49 and 2.50 for the slice-to-volume motion artifacts of the *Custom data* while in Fig.2.51 is represented the volume-to-volume motion artifacts with and without processing for the *Possum data*.

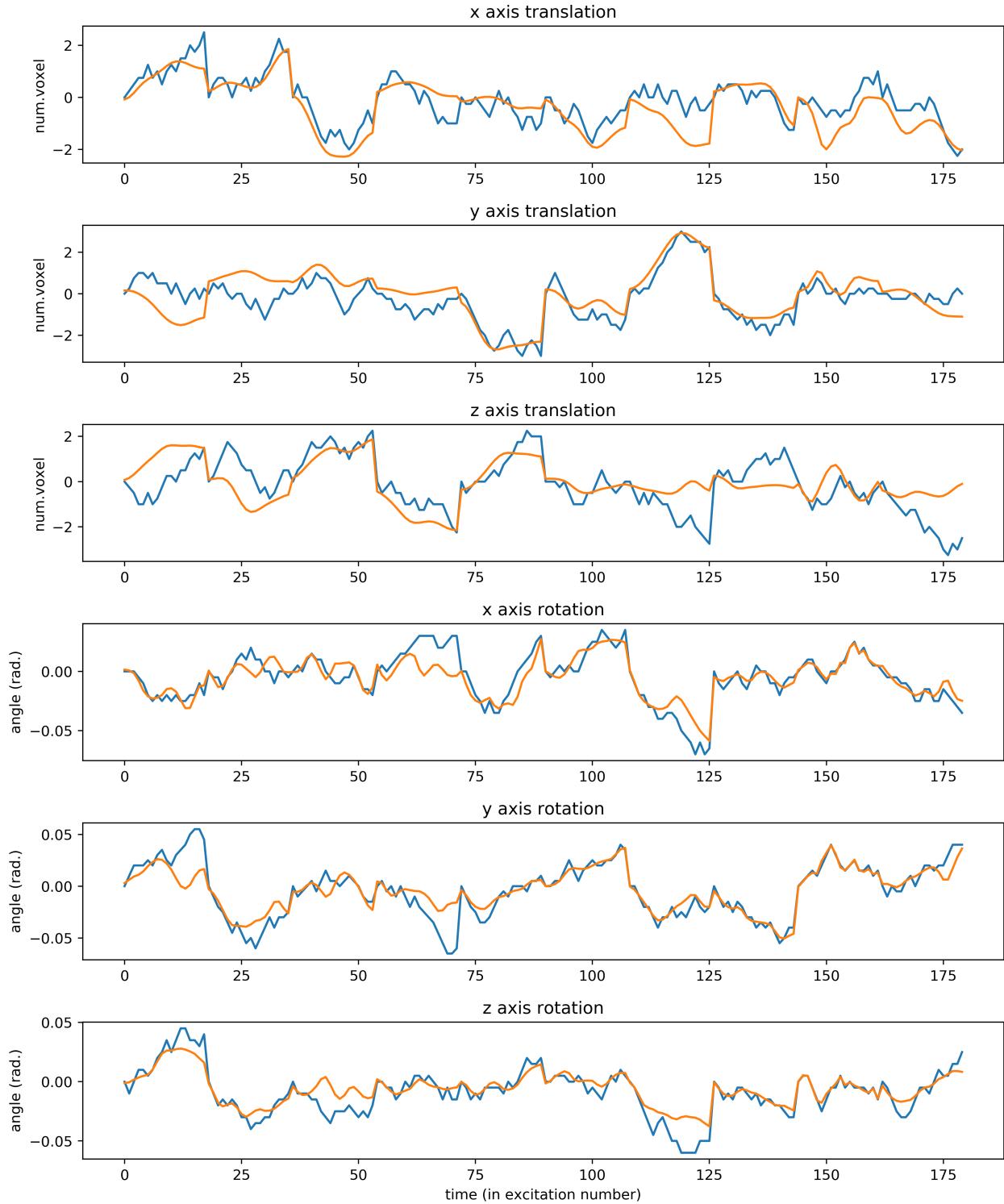


Figure 2.49: Rigid body motion parameters for the volumes $10 \rightarrow 20$ of the *Custom data* (with $b = 700[s/mm^2]$). The real motion and the estimated one are represented in blue and orange respectively in function of time (given in excitation number). The translations are measured in number of voxel and the rotations in radiant.

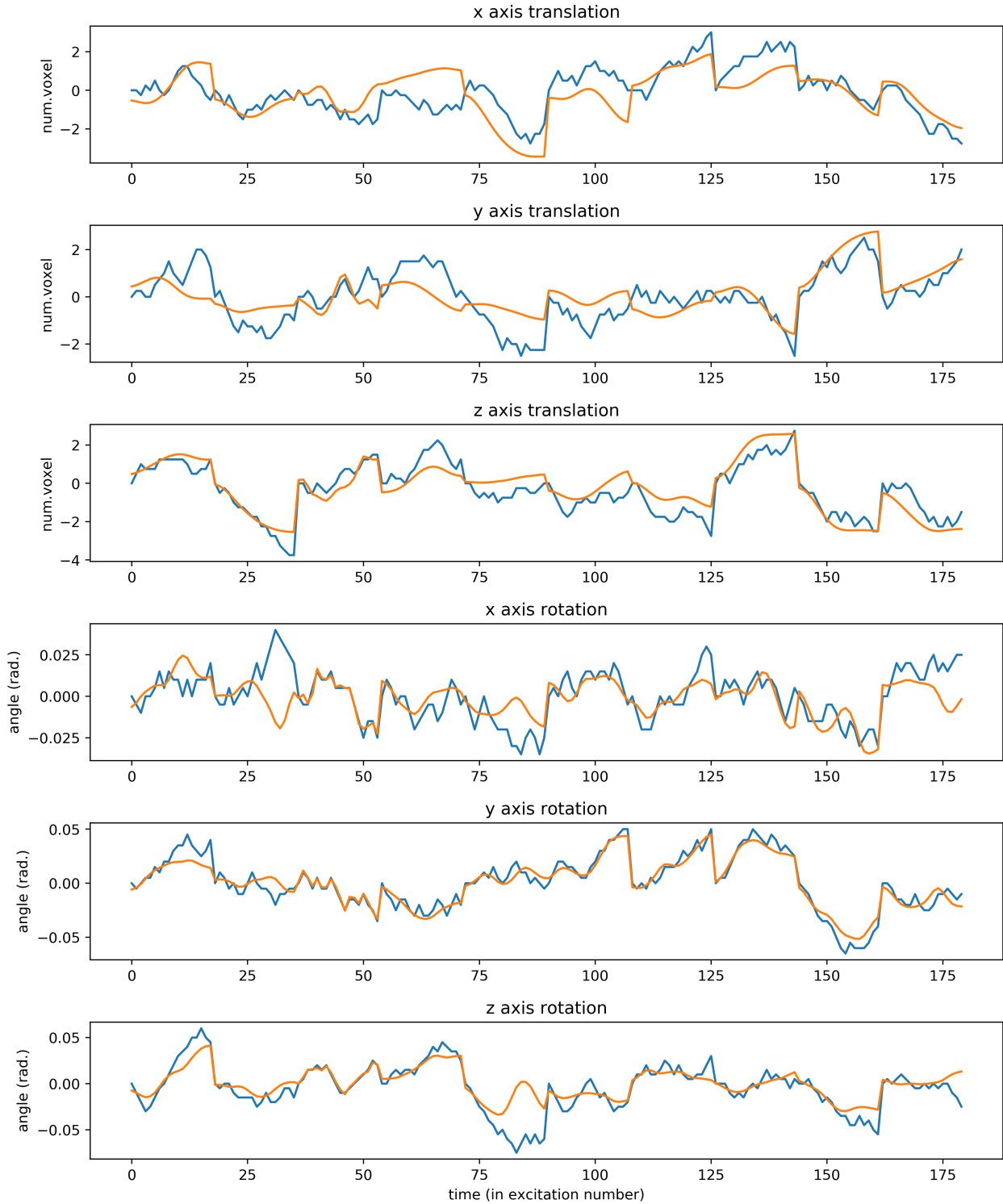


Figure 2.50: Rigid body motion parameters for the volumes $71 \rightarrow 81$ of the *Custom data* (with $b = 2000[s/mm^2]$). The real motion and the estimated one are represented in blue and orange respectively in function of time (given in excitation number). The translations are measured in number of voxel and the rotations in radian.

The Eddy currents have been simulated using a quadratic model with parameters β $[x, y, z, x^2, y^2, z^2, xy, xz, yz]$ representing the linear gradients (Hz/mm) and quadratic gradients (Hz/mm^2) in the x, y and z directions. The applied Eddy currents parameters

are compared to the estimated ones for the linear and quadratic terms in Fig.2.54.

In Fig.2.52, the estimated susceptibility off resonance field is compared to the ground truth field. The values of the fields go from -20 Hz to 80 Hz showing a mean error of 23% for the estimation with possible errors of more than 100% in certain regions of the brain.

Similarly, in Fig.2.53, the estimated bias field is compared to the ground truth field. The values of the fields range from 0.7 to 1.3 showing a mean error of 5% for the estimation with possible errors of 13% at the edges of the volumes.

The MPPCA denoising algorithm provides an estimated map of the noise and it outputted an estimated level of $\sigma = 0.71$, which compares to the noise applied during the simulation with a standard deviation of $\sigma = 1$.

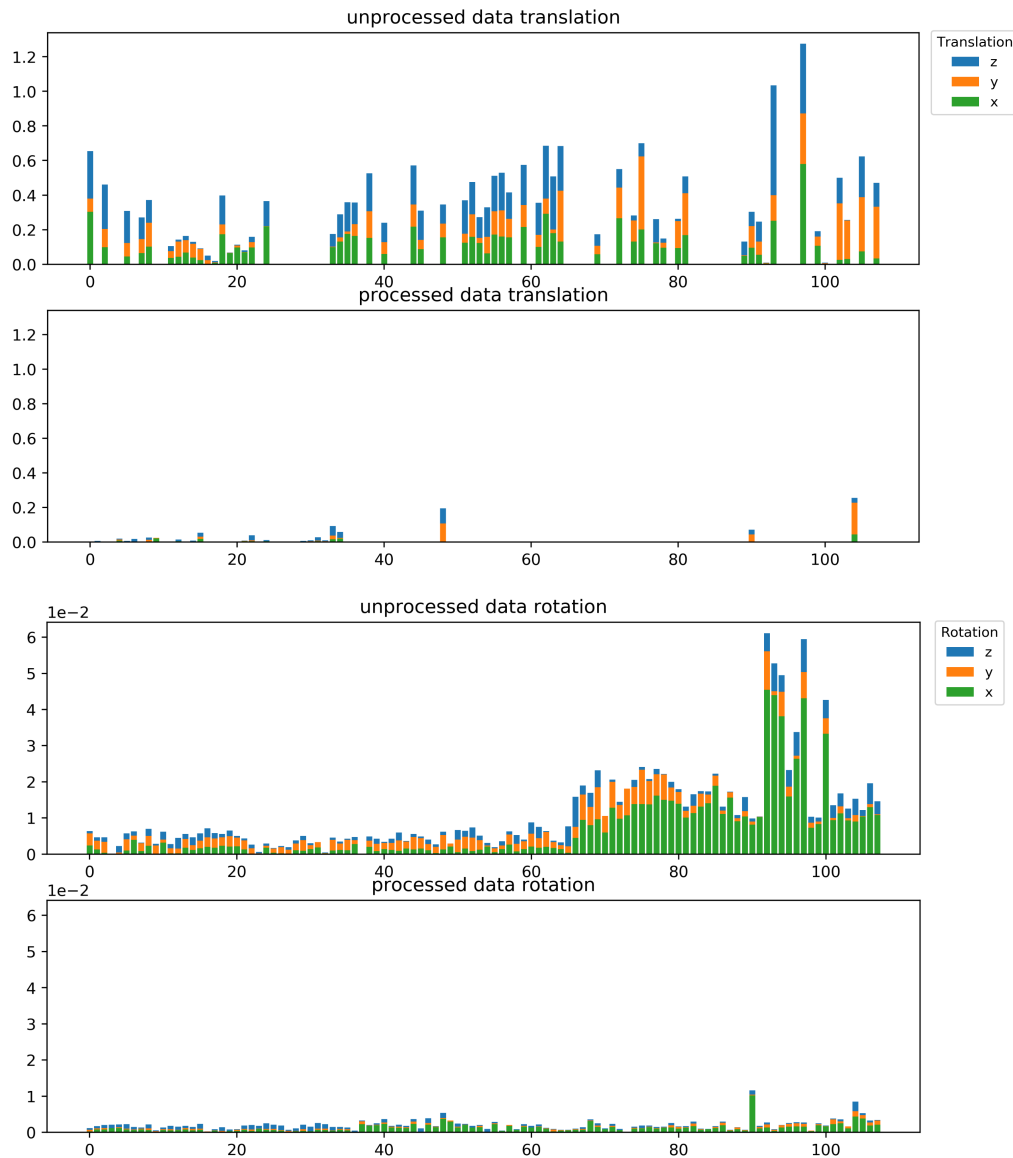


Figure 2.51: Rigid body motion for the *Possum* data before and after preprocessing for each volume considering only volume-to-volume motion. The translations are measured in millimeter and the rotations in radiant. The translation and rotation around the x, y and z axis are shown in green, orange and blue respectively. The motion is expressed relative to the b_0 to evaluate the alignment of the volumes.

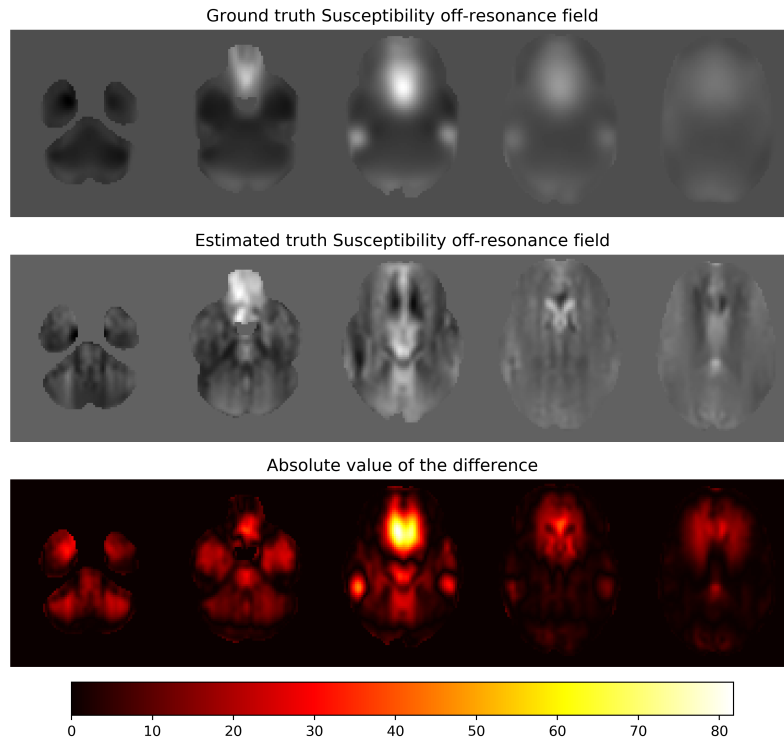


Figure 2.52: Comparison between the ground truth susceptibility off resonance field applied during the simulation and the estimated field computed by the ElikoPy pipeline. The values are given in Hz .

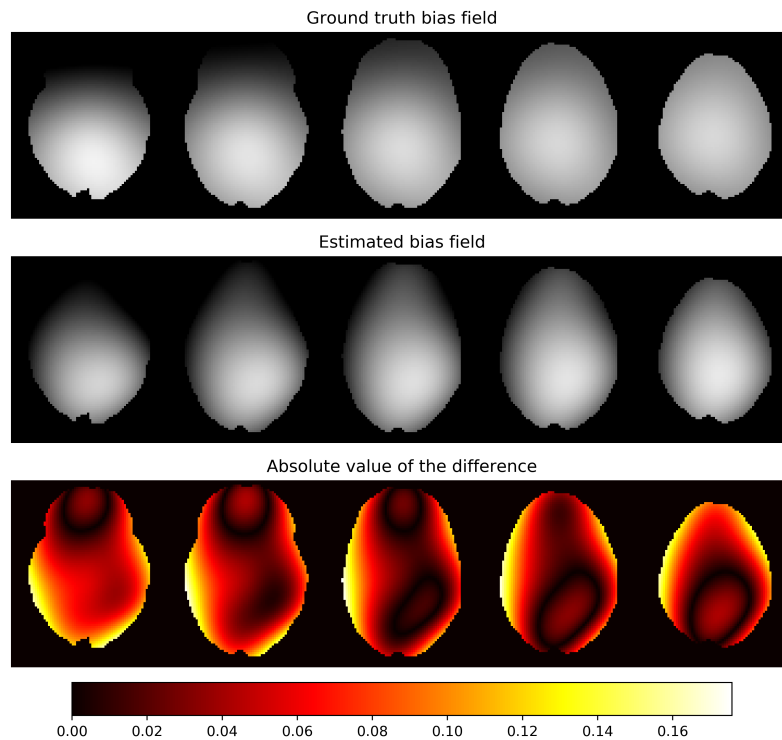


Figure 2.53: Comparison between the ground truth bias field applied during the simulation and the estimated field computed by the ElikoPy pipeline.

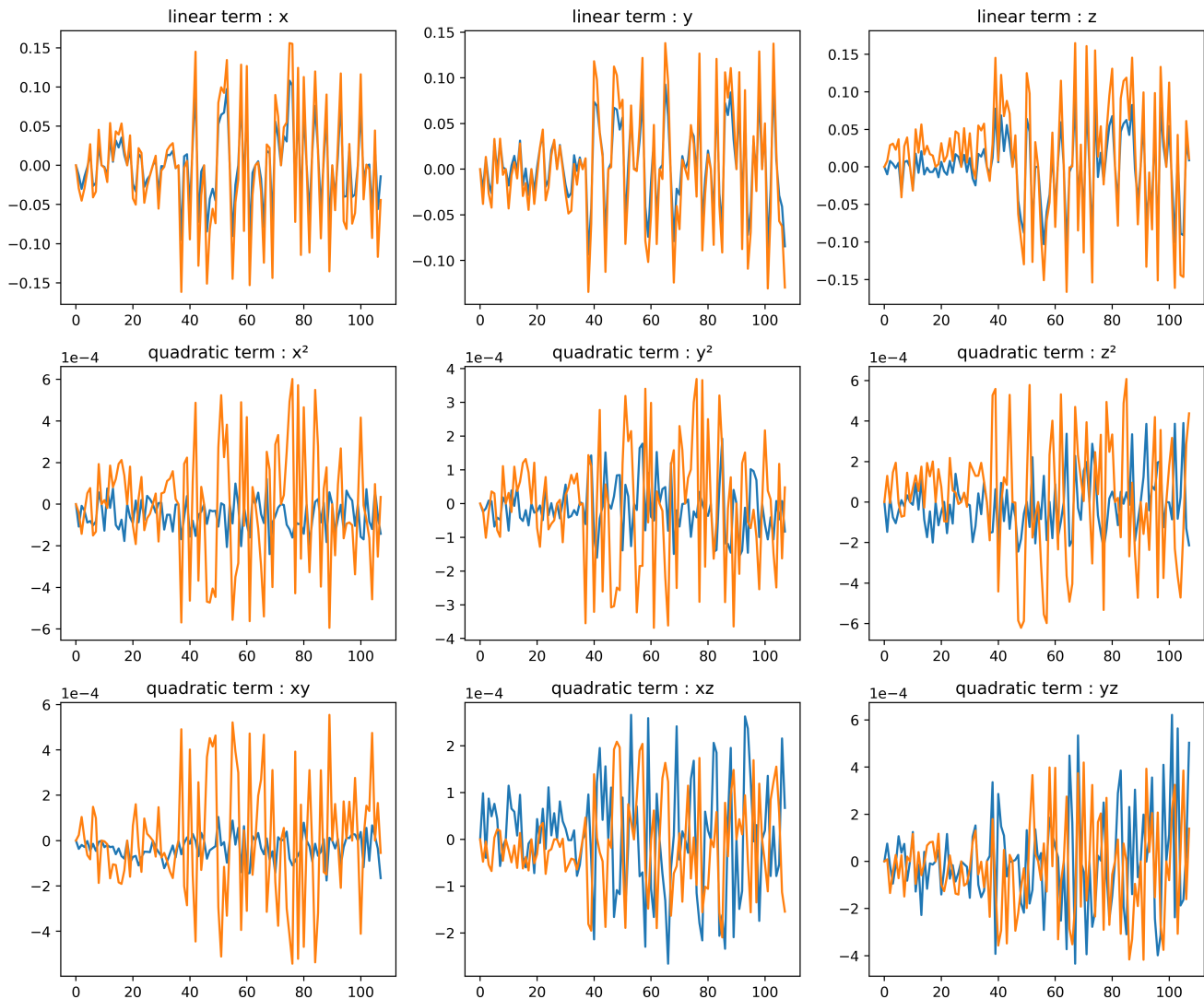


Figure 2.54: Eddy current induced distortion parameters in function of the volume index. The Eddy current model considered is quadratic. The linear terms of the model are given in $H\text{z}/\text{mm}$ and the quadratic terms are given in $H\text{z}/\text{mm}^2$. The real and the estimated parameters are represented in blue and orange respectively.

Discussion : In Fig.2.49 and 2.50, the ElikoPy library proves to be able to track the head motion with time even in harsh conditions and large movement. It can be observed that due to the Discrete Cosine Transform and the temporal regularization, the estimated motion tends to be smoother than the real movement. In practice, the ground truth head motion has rarely such sharp transitions allowing the estimation to better fit the real motion. This is shown in Fig.2.51 where the motion is effectively corrected with only residual motion remaining. The errors observed in the estimation can also be explained by the processing steps that precede Eddy FSL which tends to slightly smooth the data (especially Gibbs ringing correction).

In addition to successfully track the motion, Eddy FSL also recovers correctly the linear terms of the Eddy current parameters. The error that can be observed on the quadratic terms is due to the fact that these terms are two orders of magnitude below the linear parameters and the algorithm fails to capture such small variations that almost do not

affect the data.

The bias field correction operated by N4 recovers an estimation of the bias close to the ground truth field but tends to underestimate the values close to the edges of the volumes where less information is available.

The MPPCA denoising algorithm tends to underestimate the noise variance. The term denoising might be misleading and in practice the noise is not completely suppressed from the image but it is rather reduced since it cannot be completely separated from the signal. The MPPCA algorithm reduce the noise by nullifying the lowest principal components. However, the noise corrupts all principal components and not only the ones that are nullified during the procedure. Consequently, the standard deviation of the noise is systematically underestimated using MPPCA to avoid interfering with the underlying signal (Veraart et al., 2016).

The results observed for the estimation of the susceptibility field shows large differences that could not be explained. Visual results (section 2.4.1) and tests (section 2.4.8) have shown the susceptibility field induced distortions to be properly corrected and the differences in the estimated field could not be explained. Possible reasons for this discrepancy include misconceptions on the outputs of Topup FSL or a wrong application of the susceptibility off resonance field during the simulation.

2.4.8 Susceptibility off resonance field estimation

Using a subject with b0 acquired using two different encoding direction and a T1 structural image, the efficiency of the susceptibility correction based on Topup (Andersson et al., 2003) and Synb0-DisCo (Schilling et al., 2019) is visually demonstrated in Fig.2.55.

Discussion : The two corrected b0 based on the reverse phase encoding directions are highly similar. Since the susceptibility induced distortions affect both b0 with the same intensity in opposing direction and since Topup unwrap both images using the same estimated field, having highly similar images indicates an efficient correction, considering that the groundtruth images is theoretically located exactly between the two images in term of distortion.

The corrected b0 volumes is highly similar to the synthetic undistorted b0, which is desirable since Topup warp the real b0 into the synthesized one when using Synb0-DisCo. At last, Synb0-DisCo needs at least 16Gb of RAM when running on a CPU, requiring powerful computer for processing. Alternatively, Synb0-DisCo can utilize a Cuda capable GPU to lower the RAM requirement and considerably speed up the processing.

The smoothing visible in the volumes corrected by Synb0-DisCo is not problematic since no smoothing will be induced when the estimated field is used to correct the whole DW-MRI dataset.

The synthetized dataset (section 2.4.8) has shown the susceptibility field induced distortions to be properly corrected.

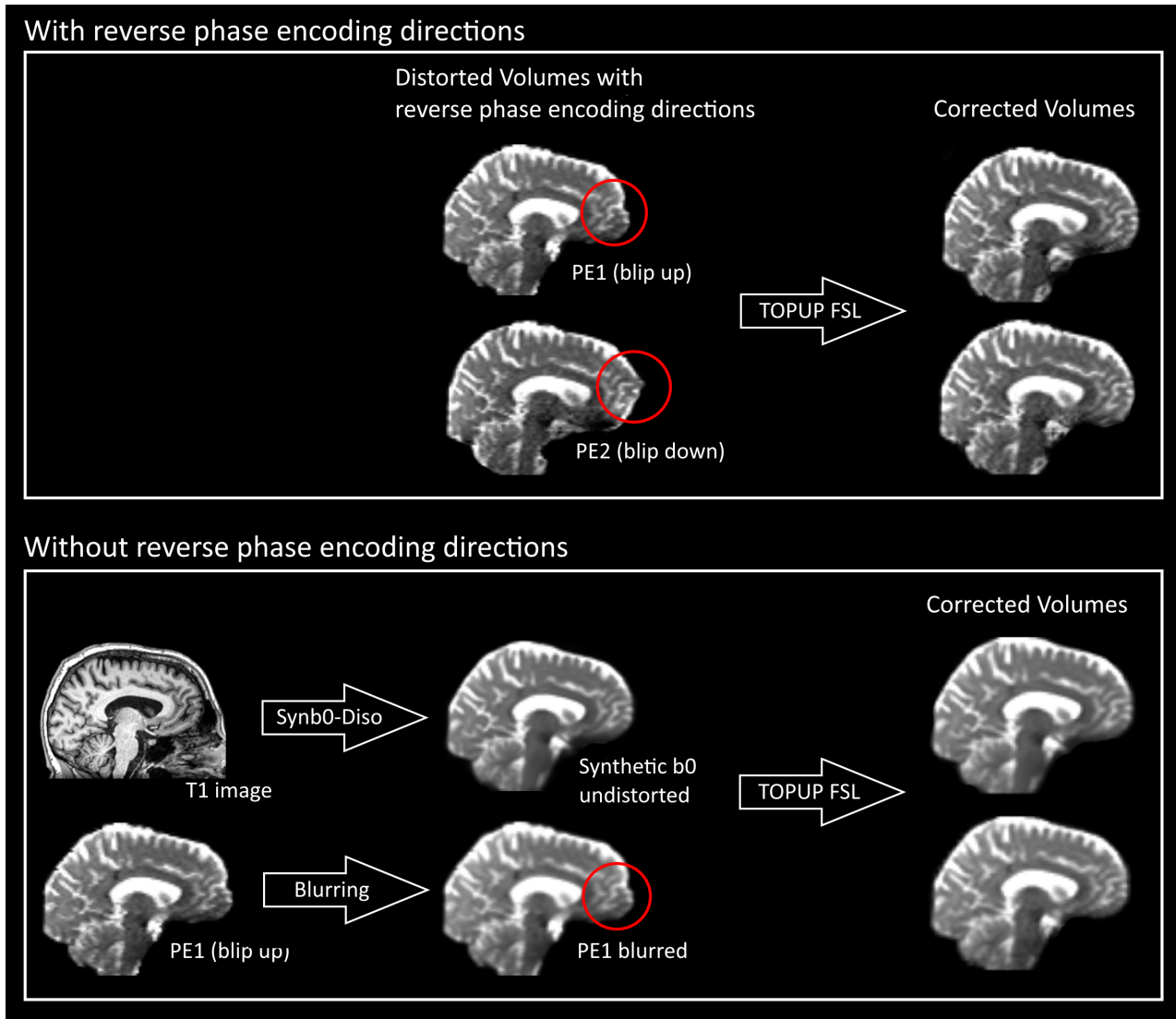


Figure 2.55: Comparison between the correction of the susceptibility resonance field applied using two opposed gradient encoding direction and a structural T1 image. The middle column represent distorted b0 volumes while the right column shows them after susceptibility distortions correction. The red circles highlight zones more affected by susceptibility induced distortions.

2.4.9 Evaluation of the White matter mask

Using the subject 21TD of the dyslexia datasets, the results of the ElikoPy white matter mask computation can be visualized. The different intermediates results of the white matter mask computation using a structural T1 image based on the technique described in 2.2.3 is displayed in Fig.2.56. In Fig.2.58, the preprocessed T1 structural in DWI space is compared with an Anisotropic Power (AP) map. The ability of the whitewater mask function to provide pertinent and adapted mask for DW-MRI data processing with T1 structural image as well as with AP map is demonstrated in Fig.2.57.

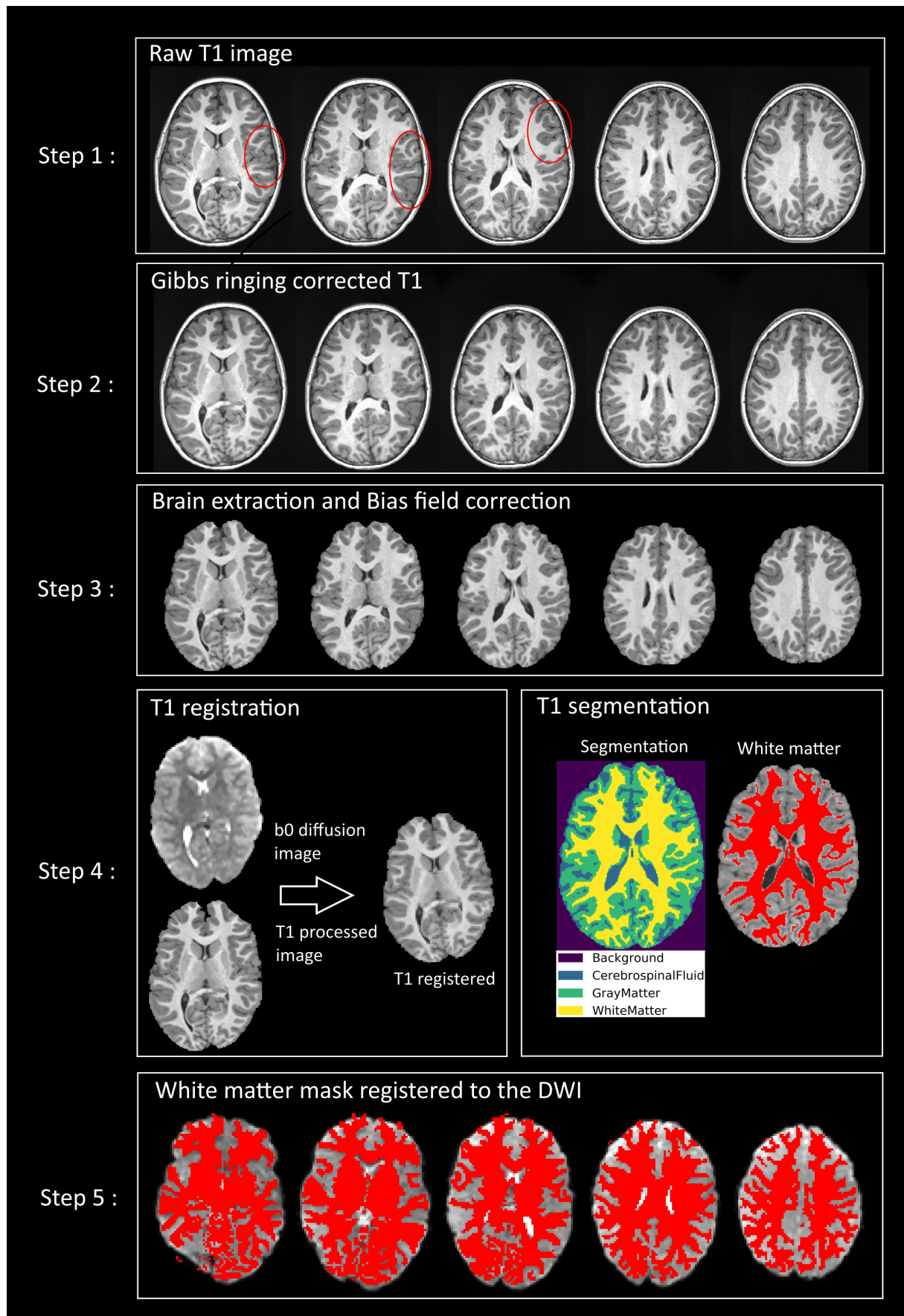


Figure 2.56: Illustration of the different steps performed on the T1 structural image to compute the white matter mask in the space of the preprocessed DWI. The red circles in step 1 highlight zones more affected by Gibbs ringing artifact.

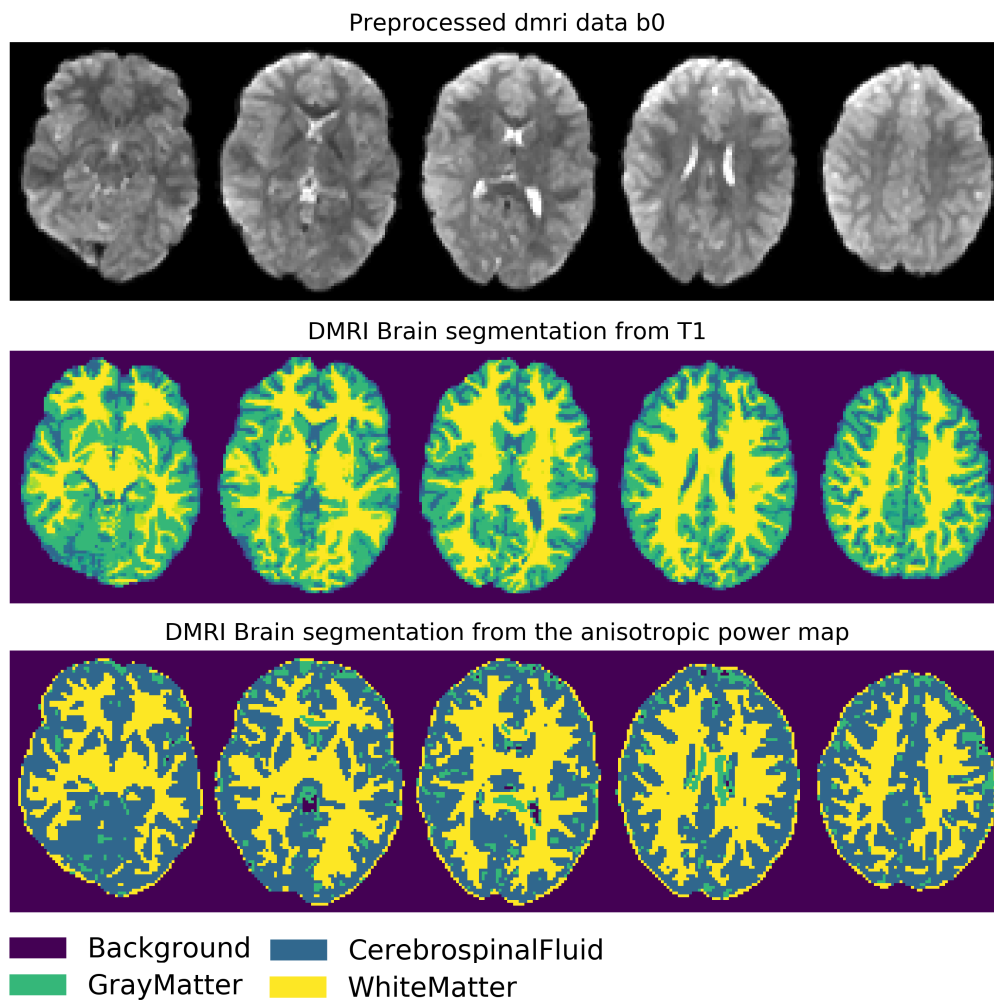


Figure 2.57: Comparison of the first b0 of the preprocessed dmri data, the segmented T1 structural image and the segmented Anisotropic Power map all in the space of the preprocessed dmri data.

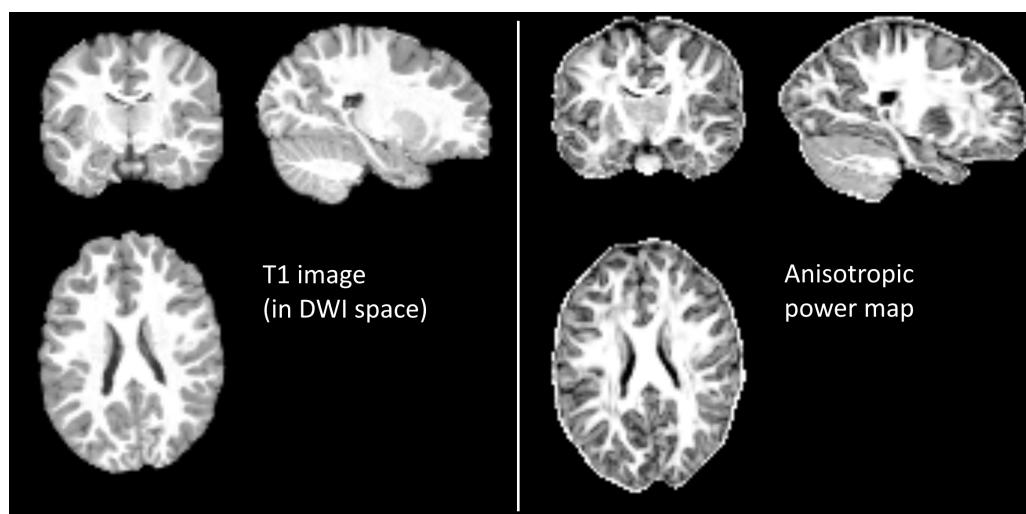


Figure 2.58: Comparison of the preprocessed T1 structural image and the Anisotropic Power map both in the space of the preprocessed DWI.

Discussion : The preprocessing step successfully corrects for Gibbs ringing artifact as visible in step 1 and 2. Moreover, the brain extraction is neat and keeps the vast majority of useful brain tissues. The negative effect of the bias field is successfully removed since the white matter has an almost uniform intensity after bias field correction. While the registration and the segmentation performed in step 4 are visually accurate, the white matter mask registered to the DWI in step 5 seems to be inaccurate. However, this is intended and a design decision. Since the white matter mask is only used to speedup the processing, including a little bit of gray matter is only barely detrimental to the processing speed but increases the odd of encompassing all the white matter. The white matter mask is thus slightly dilated after registration.

As visible in Fig.2.58, both images are able to separate white matter structures from others brain tissues similarly. The T1 structural images has a more constant pixel intensity within white matter and better separate fine path of white matter. Moreover, the T1 has different pixel intensity for the gray matter and cerebrospinal fluid while the AP map has pixel intensity similar to the T1 near the skull and a wide range of different pixel intensity for the gray matter. The impact of these two variation is visible in Fig.2.57. The AP map is unable to separate the gray matter from the cerebrospinalfluid and fail to identify fine white matter tracks. However, both brain mask successfully identify the majority of the white matter, allowing the use of the WM mask instead of the brain mask for a processing time speedup (after mask dilatation to be on the safe side).

2.5 Final discussion and perspectives

The ElikoPy library is a pipeline that aims at facilitating the processing of diffusion weighted images by embedding the different processing steps and correcting the artifacts before fitting the diffusion models.

The validation has demonstrated the ability of ElikoPy to successfully correct for the artifacts. The noise, Gibbs ringing and bias field are effectively reduced and the various susceptibility and Eddy current induced distortions are rectified. Moreover, the motion is corrected and the volumes and slices are properly realigned. These corrections have shown to significantly improve the fitting of the diffusion models especially in the white matter which contains the microstructure information of interest in dMRI. The diffusion models also exhibited increased sensitivity and have shown to better discriminate fascicles inside the voxels.

The preprocessing tends to uniformize the data across the subjects of the studies and to alleviate the non biological differences. However, the validation has shown that care must be taken when applying the Gibbs ringing and bias field correction as their effect might be detrimental on data unaffected by these artifacts. On the contrary, the MPPCA denoising proved to be a robust method applicable in both low and high noise conditions reducing the thermal noise without affecting the signal.

Based on the Synb0-DisCo framework, the pipeline adequately corrects for the susceptibility field induced distortions and provides results comparable to corrections realized with reverse phase encoding directions highlighting the adaptability of ElikoPy to different acquisition schemes.

Due to the registration errors (in presence of a T1) or the suboptimality approach on the use of the AP map for segmentation (in absence of T1), the white matter mask do

not properly segment the white matter with precision for the diffusion weighted images. This mask is however computed mainly to accelerate the processing speed of the diffusion models fit and is dilated such that all the white matter is included. Only the white matter segmentation in the T1 image space (before registration) is used to compute results.

	Typical Duration (min)
Preprocessing	
Reslice	0.5
Brain extraction	1
Denoising using MPPCA	80
Gibbs ringing correction	40
Estimation of the susceptibility field with two opposed phase encoding direction	15
Estimation of the susceptibility field with a T1 structural image	40
Motion correction with Eddy Cuda V2V	20
Motion correction with Eddy Cuda S2V	45
Biasfield correction	2
Metrics computation	
DTI	6
NODDI	110
DIAMOND	190
Microstructure Fingerprinting	70

Table 2.5: Typical runtimes of ElikoPy on the in vivo dyslexia dataset on a workstation running 64-bit CentOS 7 with a 8-core 3.2GHz AMD-FX8320 processor with 3 allocated cores, 24GB of RAM and a dedicated GTX1060 6GB GPU.

In term of effectiveness, ElikoPy offers similar performances than DESIGNER and PreQual when dealing with data with minor motion but outperforms them when dealing with data strongly deteriorated by motion. Moreover, ElikoPy is easy to install thanks to its containerization and supports an easy and seamless processing of bundles of subjects. Besides, ElikoPy preprocessing can be easily restarted from intermediate step for the whole study or for specific subjects. On the contrary, DESIGNER is difficult to install due to its dependence with no longer supported version of Mrtrix and the absence of containerized image. Furthermore, while useful for single subject processing, PreQual and DESIGNER cannot process multiple subjects and are even less adapted when dealing with studies containing data with different parameters of acquisition. By contrast, the file structure of ElikoPy allows to account for data with disparate acquisition parameters making this pipeline applicable to data from multi-site, multi-scanner studies with non uniform data formats and acquisition protocols.

To the contrary of PreQual and DESIGNER, ElikoPy computes various microstructural metrics and performs voxel wise, atlas-based region wise and volumetric statistics between different groups. ElikoPy can adapt to the eventual absence of structural T1 or reverse phase encoding direction, generate complete and easy to interpret quality report and can support, as input, studies that are in the BIDS format. Thanks to its reliance on python and its API, ElikoPy is easy to use with minimal knowledge in DW-MRI processing. The library and its container is also compatible with the scheduler present in high-performance computing centers and has been tested on the CECI cluster. Finally, ElikoPy is currently used for several studies (Delinte et al., 2021) (Kupers et al., 2021) and by other researchers.

Although facilitating the processing of diffusion-weighted images, the utilization of ElikoPy is hindered by the demanding resources requirement of the pipeline and the lengthy processing time (see Tab.2.5). Besides, ElikoPy currently offers a limited while effective

support for longitudinal studies and has still potentiality for further improvements. This includes :

- A graphical interface to ease the use of ElikoPy for users unfamiliar with programming and command line interface.
- The support of data exportation to BIDS folder tree, increasing the integration potential for multi-modal studies.
- A further extension of the documentation with the creation of tutorials in addition to the API.
- The creation of new multi-modal metrics combining the different models of microstructural metrics.
- Addition of a preprocessing pre-alignment to speed up Eddy FSL.
- Pipeline validation on brains with lesions.
- Assessment of the processing reproducibility.

Chapter 3

Motion correction with gradient cycling

In Chapter 2, the results of ElikoPy are illustrated on in vivo data acquired at St-Luc University Hospital on a 3T SIGNATM Premier GE MRI scanner using a brand new gradient cycling acquisition scheme. This scheme allows to reduce thermal load on the coils by interleaving the acquisition of volumes with different b-values. However, this interleaving comes at the cost of intense motion distortions that cannot be corrected using traditional algorithms. This issue is still unsolved by the MRI manufacturers. In this chapter are proposed potential solutions to fill that gap and a novel motion correction framework based on Eddy FSL is introduced to allow for more flexible slice realignments accommodating any slice acquisition order.

3.1 Gradient cycling

Multi-shell acquisition with high gradient volumes are required when using complex microstructural models to uncover the subtle changes in the white matter. Unfortunately, acquiring sequentially high b-values with the same encoding for every slice leads to power and heating peaks (Hutter et al., 2018b). Even though only few volumes are acquired with high b-values, these volumes define the gradient cooling time and therefore increase the repetition time for the whole scan.

In order to reduce the cooling time, a first simple approach is to interleave volumes of high and low b-values to let hardware cool down. Indeed, ohmic heating is less important when using low b-values. However, slice interleaving can lead to an even lower thermal load as illustrated in Fig.3.1. The technique of interleaving slices of different b-values is also known as gradient cycling.

In Spin-Echo single-shot Echo-Planar Imaging, each slice is acquired independently within an EPI sequence block. The total number of acquired slices N_t in an EPI acquisition sequence is given by

$$N_t = d \cdot N_s,$$

where $d = 1, \dots, N_d$ is the defined number of diffusion encodings and N_s is the number of slices $s = 1, \dots, N_s$ by diffusion encoding direction. Every slice can be labeled by its temporal index t with $t = 1, \dots, N_t$, within the sequence. The acquisition is split in volumes v with $v = 1, \dots, N_d$ containing N_s subsequent slice excitations, each acquiring all slices

specified within one geometric volume.

In sequential EPI dMRI sequences, all slices within a volume are acquired with the same diffusion encoding. Thus, N_s repetitions of the same sequence block lead to a full volume acquisition and define the repetition time (TR) of the sequence. To perform a complete acquisition, N_d TR sequence are performed to acquire all N_d diffusion encodings. To reduce thermal loads, volumes of high and low diffusion encodings are interleaved.

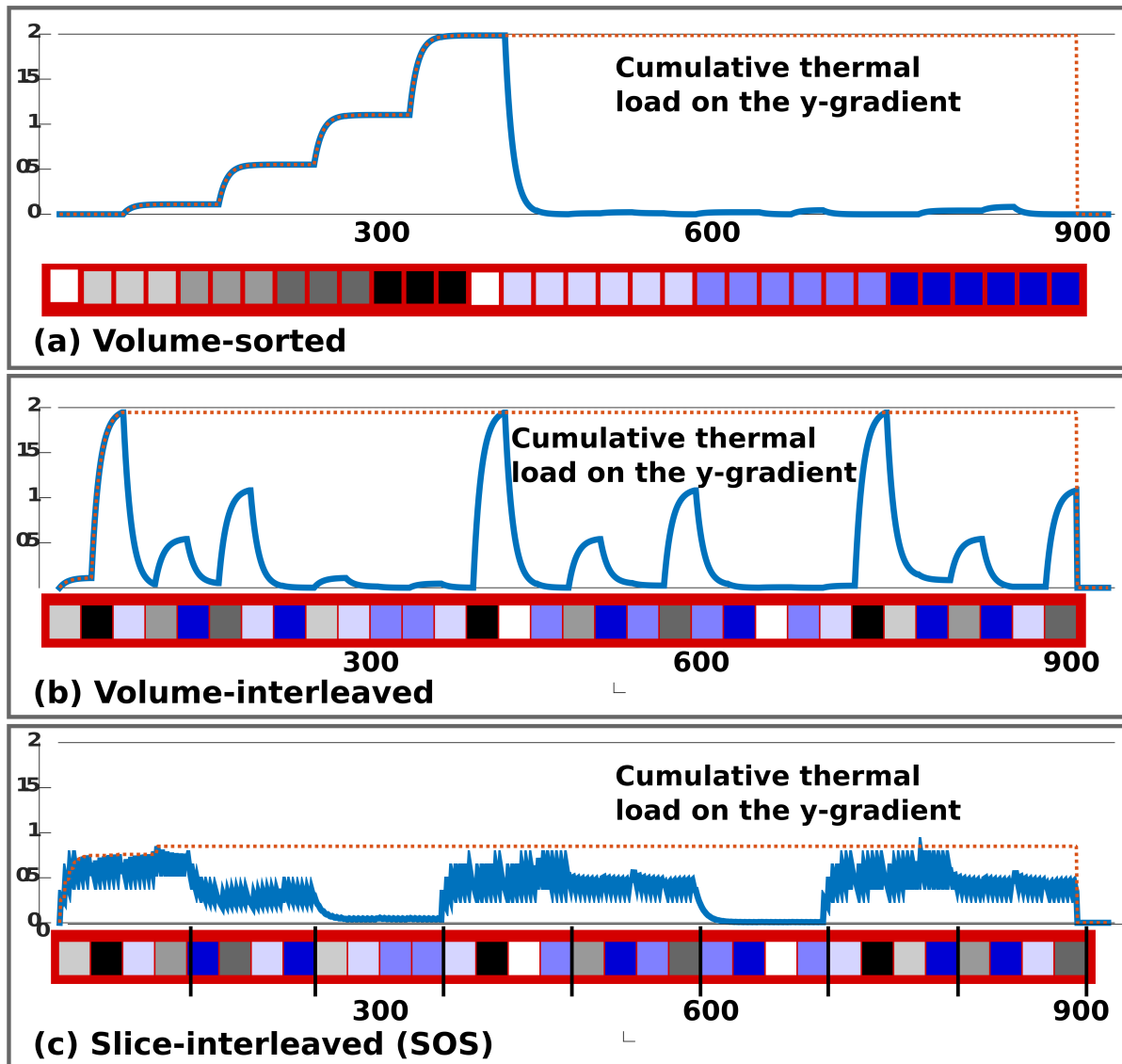


Figure 3.1: Thermal load evolution over slices for volume-sorted, volume-interleaved and slice-interleaved cases on a clinical Philips 3T-Achieva. The slice-interleaved case uses slice-interleaved free-waveform imaging (SIFI) acquisition scheme. From (Hutter et al., 2018b).

In gradient cycling dMRI sequences, different diffusion encodings are used within a single EPI sequence block, resulting in the interleaving of slices with different encodings in a single volume during acquisition (Hutter et al., 2018a). In Fig.3.2 is shown the main differences between the two acquisition schemes.

After the acquisition, a conventional volume-view data set is reconstructed by the scanner

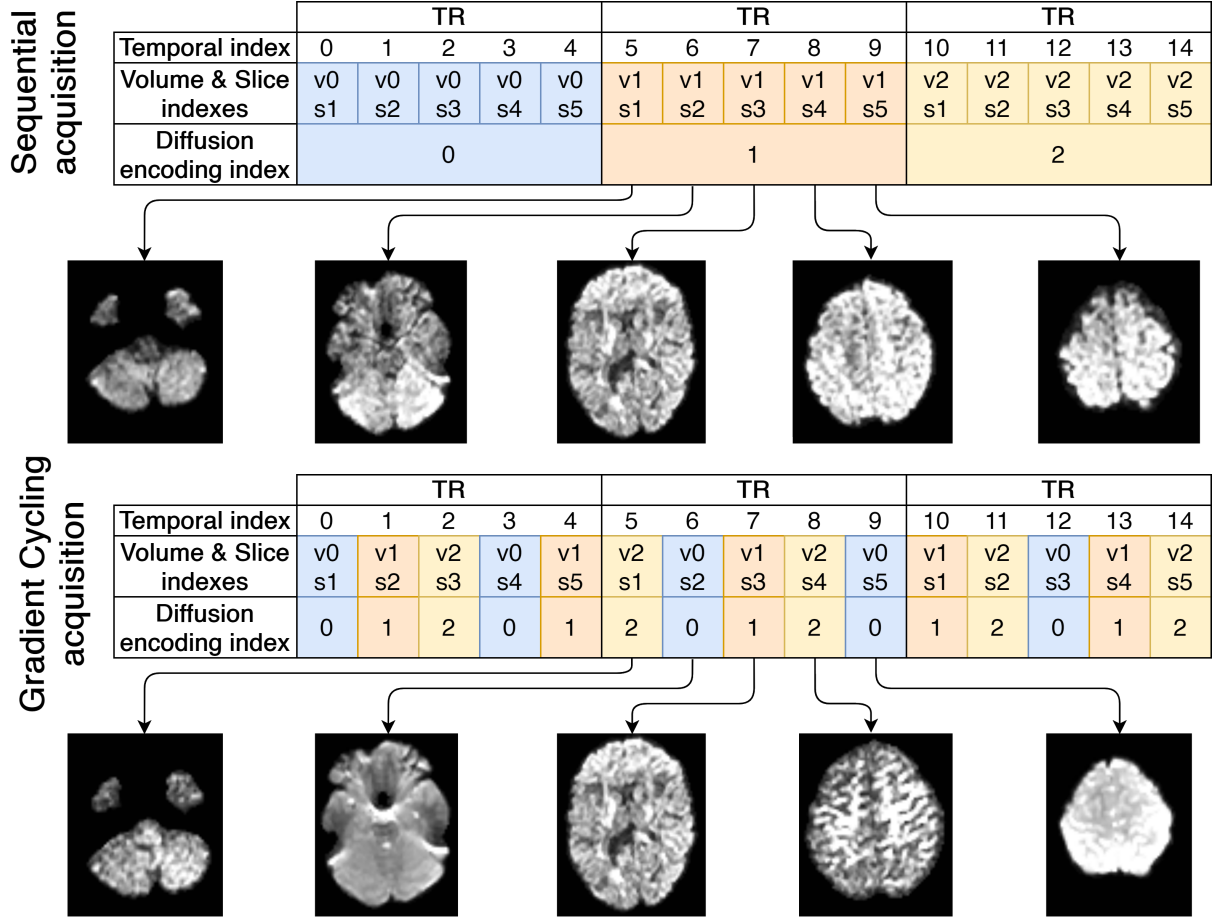


Figure 3.2: The first table depicts a conventional sequential acquisition while the second table depicts an acquisition realized with gradient cycling. The selected parameters are $N_s = 5$, $N_d = 3$ and $v = 3$. Each diffusion encoding is done with different b-values and b-vectors.

by sorting the acquired volumes into new volumes with uniform internal diffusion encoding. This sorting results in the introduction of a wide amount of slice-to-volume motion artifacts. Motion that would have been initially inter-volume motion is converted into intra S2V motion. The significant temporal decorrelation of slices within a reconstructed volume not only creates inter-slice shifts but also rotations between slices. This breaks the intra-volume consistency and renders interpolation between slices almost impossible. Indeed, small inter-volume rotations are converted into inter-slice rotations after reconstruction.

Slice-to-volume (S2V) motion correction performed by Eddy FSL and which is used in the ElikoPy library requires the knowledge of the (relative) acquisition timing of each slice for a given volume. Using gradient cycling prevents the use of S2V motion correction since Eddy only accepts a slice acquisition specification that is common to all volume. Moreover, the hypothesis of a high temporal correlation in the displacement of slices within a volume used by Eddy no longer holds. Indeed, the S2V model of Eddy described in section 2.2.2.5 relies on discrete cosine transform (DCT) for the modeling of motion. The hypothesis of temporality is broken since slices belonging to different multi-band groups in the same reconstructed volume will be acquired with large time intervals.

3.2 Eddy slice-to-volume correction

As explained in the previous section, while gradient cycling helps to reduce hardware heating, its use introduces several challenges related to motion correction with Eddy FSL. In order to tackle these challenges, modifications of Eddy have been introduced using a combination of carefully chosen Eddy parameters and a new way of defining the slice specifications.

Adjustment of Eddy parameters : The motion regularization parameter (λ) and the order of the DCT expansion (M) are the two tweaked Eddy parameters (see section 2.2.2.5). λ defines the strength of the temporal regularization on the model of the slice-to-volume movement. The temporal regularization is used to prevent slices acquired sequentially from having a completely different estimated movement, enforcing the continuity of movement. Values in the ranges 1–10 provide generally good results (Andersson and Sotiropoulos, 2019) for S2V correction without gradient cycling. The M parameter changes the temporal order of movement by fixing the number of degrees of freedom in the discrete cosine transform DCT basis-set used for modeling movement. If the order of the DCT expansion is set to $N - 1$, where N is the number of excitation in a volume, the location of each slice (or group of slices in the case of multi-band) is individually estimated. A value ranging from $N/4$ to $N/2$ is generally recommended for s2v correction without gradient cycling (Andersson and Sotiropoulos, 2019).

Since slices in the same volume are not acquired sequentially and are thus much more temporally uncorrelated, the default temporal regularization of slice-to-volume movement becomes inadequate. To solve this issue, the value of motion regularization term is greatly reduced. The default value of one is reduced to a value in the range of 0.01 to 0.2 depending on the quantity of S2V movement present in the dataset. For the same reason, the temporal order of movement is augmented to a maximum in order to individually estimate the slices locations, allowing the handling of volumes with slice-to-volume movement present on every slice.

Definition of the slice acquisitions : Without gradient cycling, when performing slice-to-volume motion correction, a single slice specification file describing which slices were acquired together (in the case of multi-band) and the order in which these acquisitions occurred is given to Eddy using the command line option `-SLSPECS` or `-JSON`. This corresponds to the $Z(t_n)$ function described in section 2.2.2.5. The file information is then stored as an array in the `MultiBandsGroups` classes of Eddy.

In the case of gradient cycling, the slices of each volume are acquired in different orders and volume specific slice specifications are required : $Z_i(t_n)$ with i the volume index. In order to allow the `MultiBandsGroups` classes to store the slice acquisition specification as a volume specific array, a new command line option `-GC_SLSPECS` has been added to Eddy, enabling the support of slice-to-volume motion correction with gradient cycling. `-GC_SLSPECS` provides the path where the volume specific slice specifications are stored following the naming convention `GC_<volume_id>_slspec.txt` with the `<volume_id>` starting from zero. The source code of the edited Eddy is available on Github (Simon and Dessain, 2021b).

3.3 Methods

In this section, are presented the techniques implemented in order to validate the custom Eddy S2V motion correction. The data used and the way they have been processed are also described.

In-vivo DWI data : To assess the quality of the custom Eddy S2V motion correction, the same dataset of 35 children aged from 8 to 11 described in section 2.3.1 is used. The acquisition was done on a 3T GE SIGNA Premier scanner (GE Healthcare, Chicago, IL) with the following parameters : $TR = 4842ms$, $TE = 77ms$, $2mm$ isotropic voxels (in-plane FOV : $220 \times 220mm^2$), $\Delta = 35.7ms$, $\delta = 22.9ms$, 64 gradients at $b = 1000$, 32 at $b = 2000, 3000, 5000s/mm^2$, corresponding to diffusion gradients up to $68.9mT/m$, and seven interspersed b_0 images. Gradient cycling was performed in one block containing all the volumes except the first b_0 volume as detailed in Tab.3.1.

TR 0																						
v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0	v0
s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15	s16	s17	s18	s19	s20	s21	s22	s23
TR 1																						
v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15	v16	v17	v18	v19	v20	v21	v22	v23
s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1
TR 2																						
v24	v25	v26	v27	v28	v29	v30	v31	v32	v33	v34	v35	v36	v37	v38	v39	v40	v41	v42	v43	v44	v45	v46
s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1	s1
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
TR 9																						
v161	v162	v163	v164	v165	v166	v167	v168	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15
s1	s1	s1	s1	s1	s1	s1	s1	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2
TR 10																						
v16	v17	v18	v19	v20	v21	v22	v23	v24	v25	v26	v27	v28	v29	v30	v31	v32	v33	v34	v35	v36	v37	v38
s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2	s2
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
TR 170																						
v146	v147	v148	v149	v150	v151	v152	v153	v154	v155	v156	v157	v158	v159	v160	v161	v162	v163	v164	v165	v166	v167	v168
s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23	s23

Table 3.1: Diffusion gradient acquisition scheme of the dyslexia dataset. The first volume is a b_0 acquired sequentially, the others 168s volumes contains b_0 , low and high diffusion gradient acquired with gradient cycling. Each volume v contains 69 slices acquired with a multi-band factor of three resulting in 23 slices group's s acquisition.

Processing : The dataset has been preprocessed using the ElikoPy pipeline. The processing was performed twice, once with the volume-to-volume Eddy motion correction and the other time with the novel custom slice-to-volume Eddy motion correction. The S2V motion correction was performed with a temporal order of movement (M) of 22 and a temporal regularization term (λ) of 0.1.

Validation through evaluation of the FA (DTI) diffusion metric : Using ElikoPy library functions (section 2.2.5), the S2V motion corrected data and V2V motion corrected data are projected into a common space. Region wise comparison is performed based on the JHU DTI-based white matter atlas available with FSL (see section 2.2.5). The goal is to assess the differences between the S2V and V2V motion corrected data for the FA (DTI) diffusion metric.

Validation through visual assessment of performances using b0 volumes : The first and 100th volumes of S2V motion corrected data and V2V motion corrected data for three subjects are visually compared. The three selected subjects are the subject 21TD (almost not affected by S2V motion artifacts), the subject 30DYS (slightly affected by S2V motion artifacts) and the subject 8DYS (severely affected by S2V motion artifacts).

Validation through visual assessment of performances on tractography : The S2V motion corrected data and V2V motion corrected data are tracked using deterministic and probabilistic whole brain tractography algorithms for the subject 21TD (almost not affected by S2V motion artifacts), the subject 30DYS (slightly affected by S2V motion artifacts) and the subject 8DYS (severely affected by S2V motion artifacts). Visual and numeric comparison of three isolated tracts are performed. The isolated tracts are the posterior corpus callosum, the right superior longitudinal fasciculus 2 and the right corticospinal tract. These three tracts have been chosen since they offer a good overview of the possible tract orientations. The results for the subject 21TD and tracts found using deterministic tractography are available in appendix C.2. The definition used to isolate the tracts is available in appendix C.3.

3.4 Results

3.4.1 Region wise mean FA

Using the complete dataset, the results of the custom S2V motion correction processing can be visualized and compared to the results of the V2V motion correction processing through a region wise comparison of the mean FA for all subjects of the dyslexia study. The comparison is available in Fig.3.3.

Discussion : A slight decrease in fractional anisotropy is observed after S2V motion correction. One explanation could be that the slice realignment breaks the intra-volume consistency creating volume misalignments between different volumes which leads to more isotropy. These could be due to inconsistent slice realignments between volumes.

Moreover, since Eddy performs outlier replacement, another explanation could be that excluded slices will be re-interpolated using the information contained in adjacent slices. Since there is a lot of S2V movement, the interpolation will likely occur between slices that undergo different rotation movement artifacts, further breaking the consistency with the diffusion vector associated with the corrected volume.

Further investigation is needed to strengthen these hypotheses.

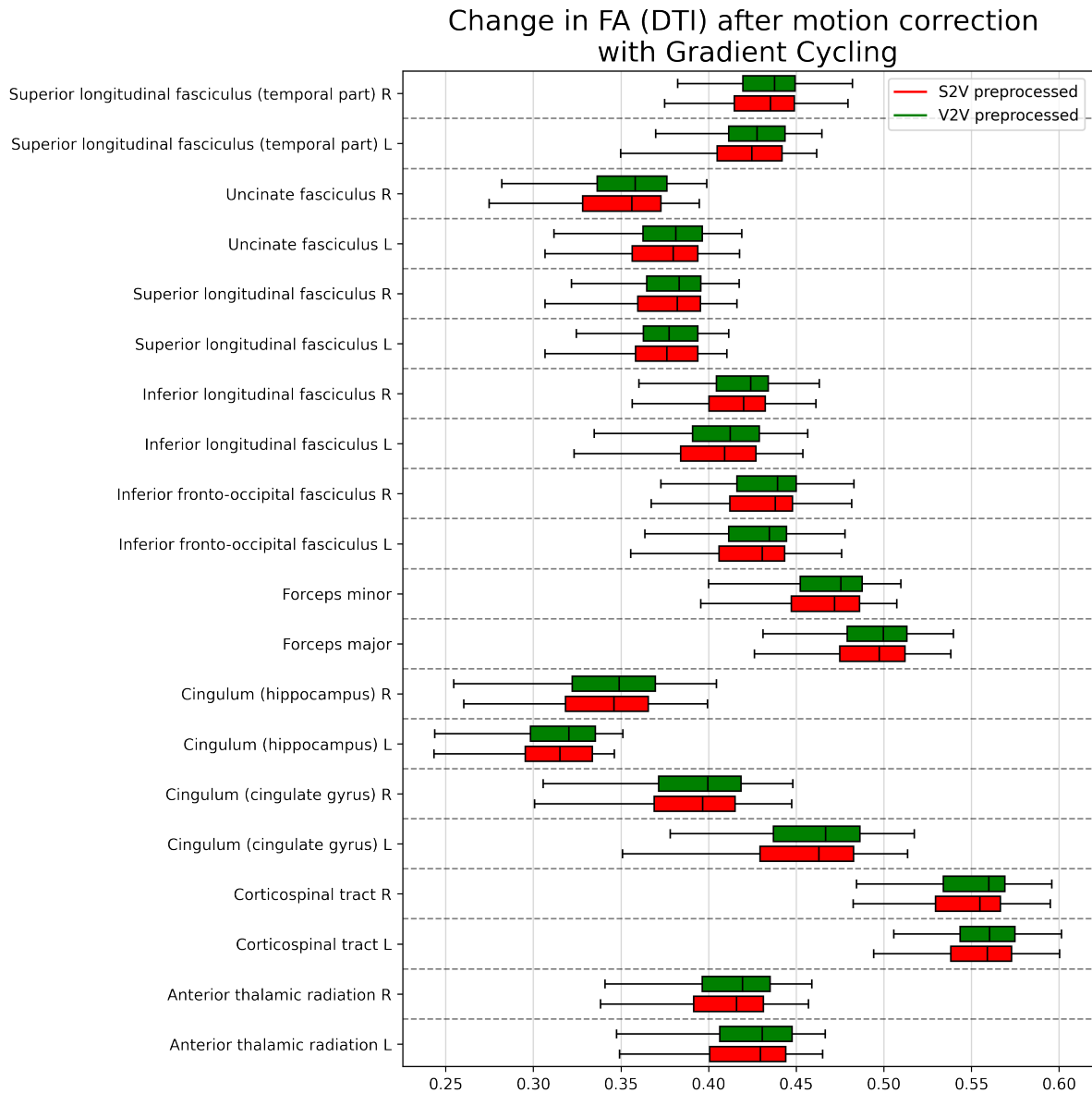


Figure 3.3: The green boxplots represent for each region of the JHU DTI-based white matter atlas the FA values of the subjects of the dyslexia study with volume-to-volume motion correction while the red boxplots represent the FA values of the subjects with slice-to-volume motion correction.

3.4.2 Visual comparison of b0 volumes

The following figures show the 100th volume of the DW-MRI scan with V2V and S2V motion correction. For the subject with mild S2V motion artifact (30DYS), the results are available in Fig.3.4 (V2V correction) and Fig.3.5 (S2V correction). For the subject with severe S2V motion artifact (8DYS) the results are available in Fig.3.6 (V2V correction) and in Fig.3.7 (S2V correction). The figures for the reference subject (21TD) are available in appendix C.1.

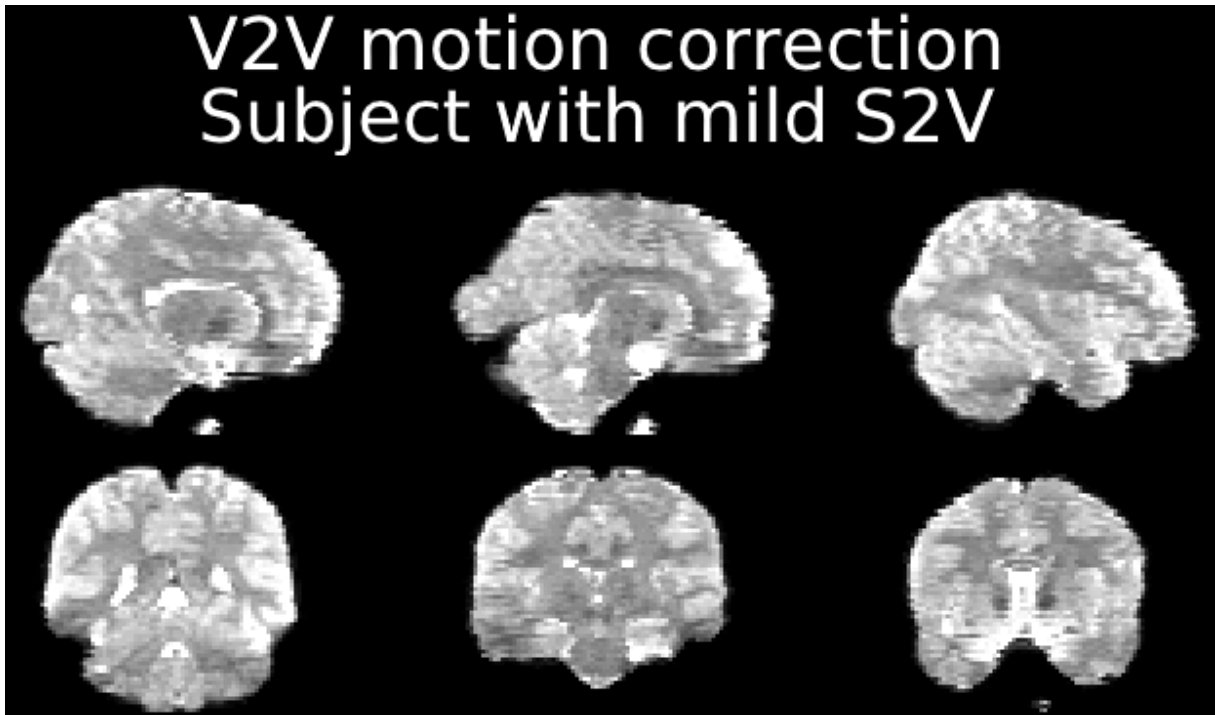


Figure 3.4: 100th volume of subject 30DYS after V2V movement correction. On the first row, multiple artifacts are visible on the different sagittal cuts. A lot of horizontal displacements are visible. The second row features three coronal cuts with fewer artifacts.

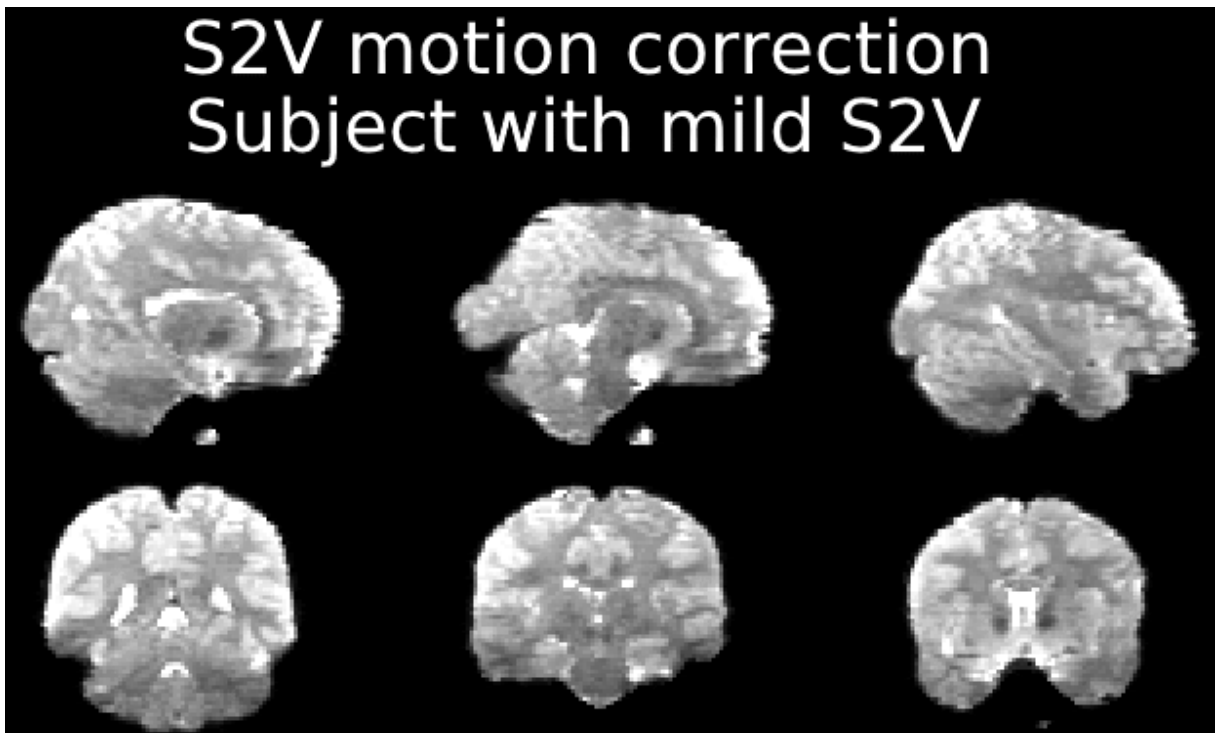


Figure 3.5: 100th volume of subject 30DYS after S2V movement correction. On the first row, fewer artifacts than with the V2V motion correction are visible on the different sagittal cuts. The second row features three coronal cuts with an important decrease of motion artifacts compared to the different V2V motion corrected coronal cuts.

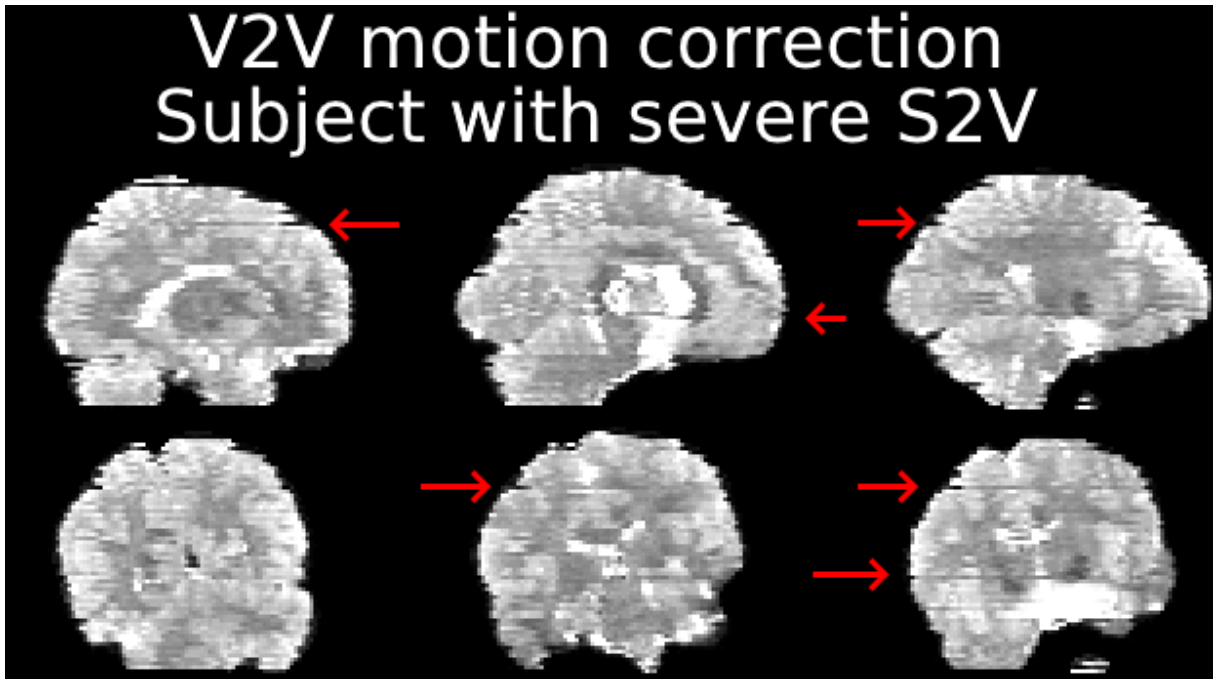


Figure 3.6: 100th volume of subject 8DYS after V2V movement correction. On the first row featuring sagittal cuts and second rows featuring coronal cuts, multiple motion artifacts are visible with a lot of horizontal displacement. Some slices highlighted by the red arrows seem to have experienced rotation since their intensities do not match adjacent slices.

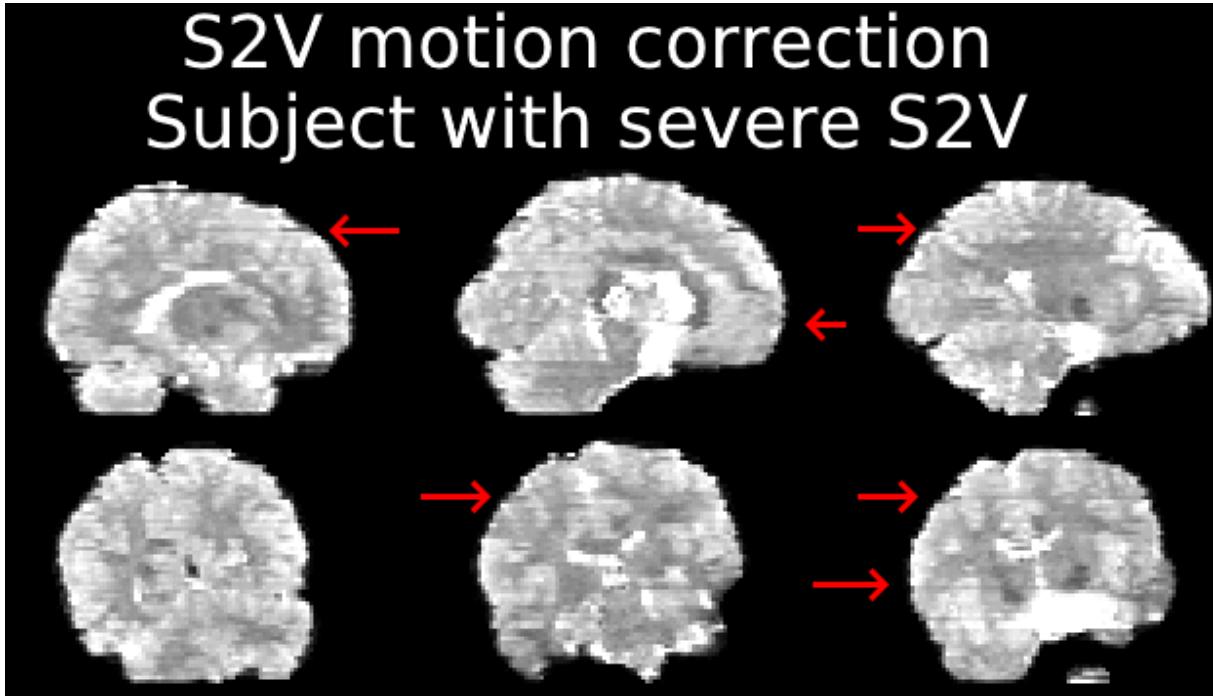


Figure 3.7: 100th volume of subject 8DYS after S2V movement correction. On the first row featuring sagittal cuts and second rows featuring coronal cuts, motion artifacts are visible while less important than without S2V motion correction. The slices highlighted by the red arrow that had intensity issues when performing V2V motion correction seems to also feature important artifacts with the S2V motion correction.

Discussion : S2V motion correction on the subject with mild motion seems to decrease visual artifacts. The correction is the most visible when comparing the coronal view. On the subject with severe motion correction, the visual assessment seems to indicate that fewer artifacts are corrected than with mild motion. Moreover, a little bit of smoothing is induced after correction, probably caused by outlier replacement and interpolation done by Eddy S2V.

Overall, the S2V corrected b0 volumes display fewer artifacts than the V2V corrected b0 volumes, indicating that intra-volume motion correction is occurring.

For both subjects, the axial cut is not display since slices are acquired in single shots and are not affect by S2V motion artifacts.

3.4.3 Visual comparison of bundles

Fig.3.8 and Fig.3.9 visually display the tracking of the corpus callosum along statistical metrics. The corpus callosum is a wide, thick nerve tract, consisting of a flat bundle of commissural fibers, beneath the cerebral cortex in the brain. In Fig.3.10 and Fig.3.11 are displayed the tracking of the right superior longitudinal fasciculus 2 along statistical metrics. The right superior longitudinal fasciculus 2 tract connects the caudal inferior parietal cortex to the dorsolateral prefrontal cortex. The right corticospinal tract, a white matter motor pathway starting at the cerebral cortex that terminates in the spinal cord, is displayed in Fig.3.12 and Fig.3.13 along statistical metrics (Kolb and Whishaw, 2009).

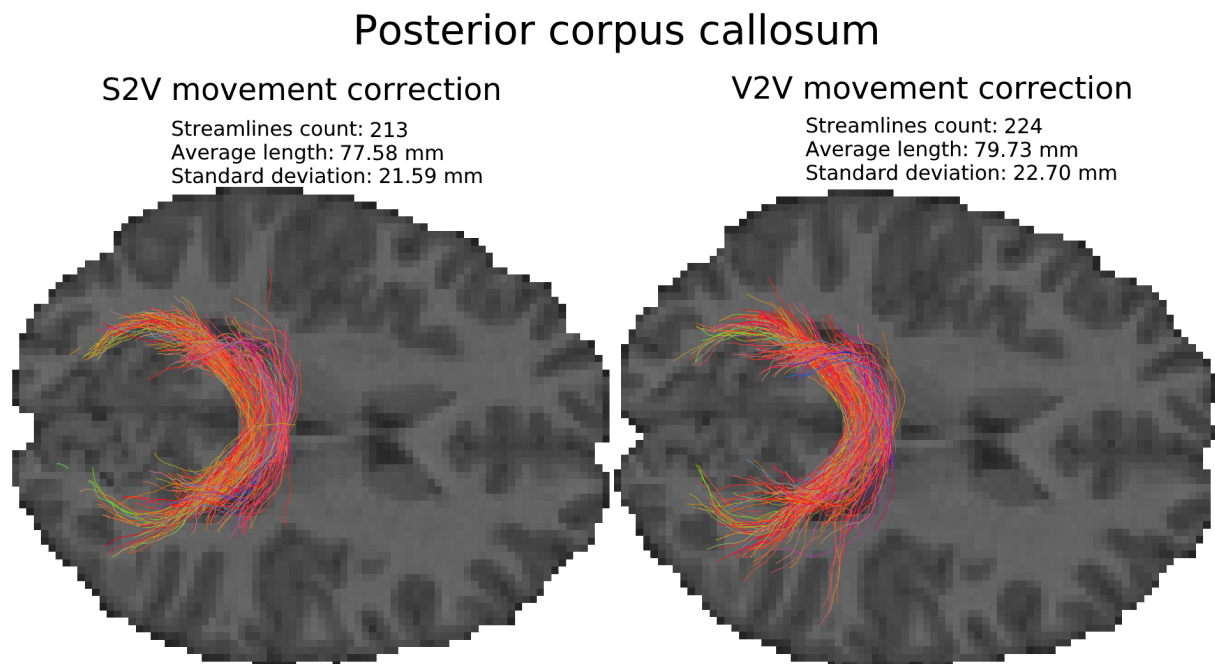


Figure 3.8: Comparison of the effect of movement correction on **probabilistic** tracking of the posterior corpus callosum on the subject with **mild** slice-to-volume movement artifact.

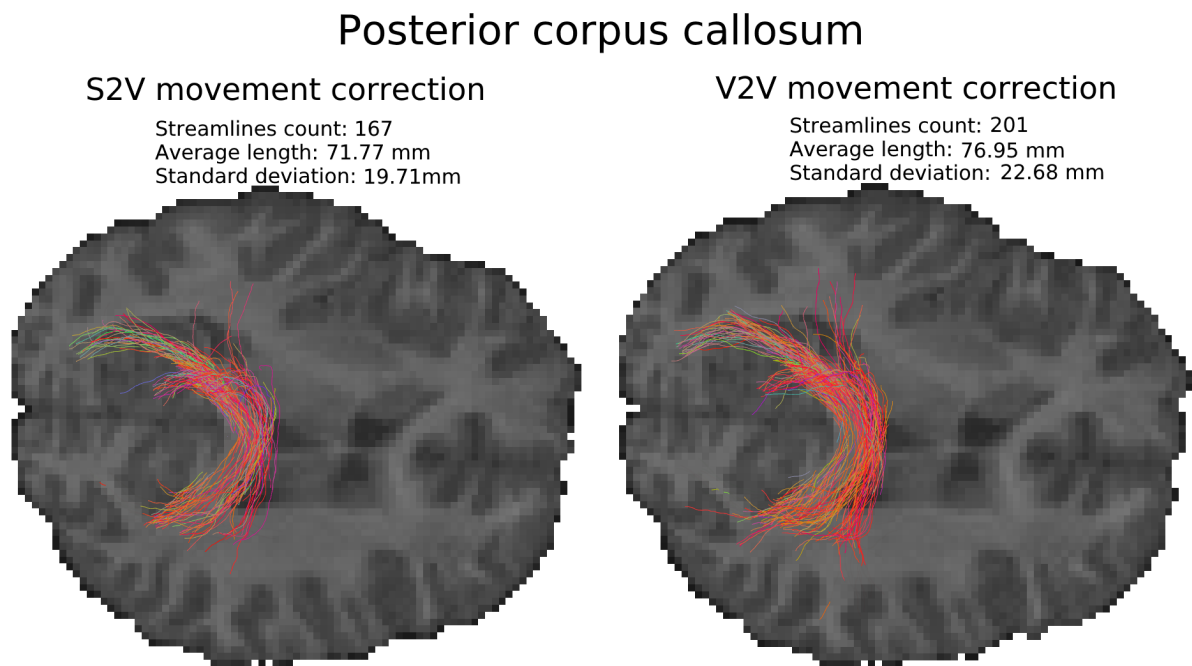


Figure 3.9: Comparison of the effect of movement correction on **probabilistic** tracking of the posterior corpus callosum on the subject with **severe** slice-to-volume movement artifact.

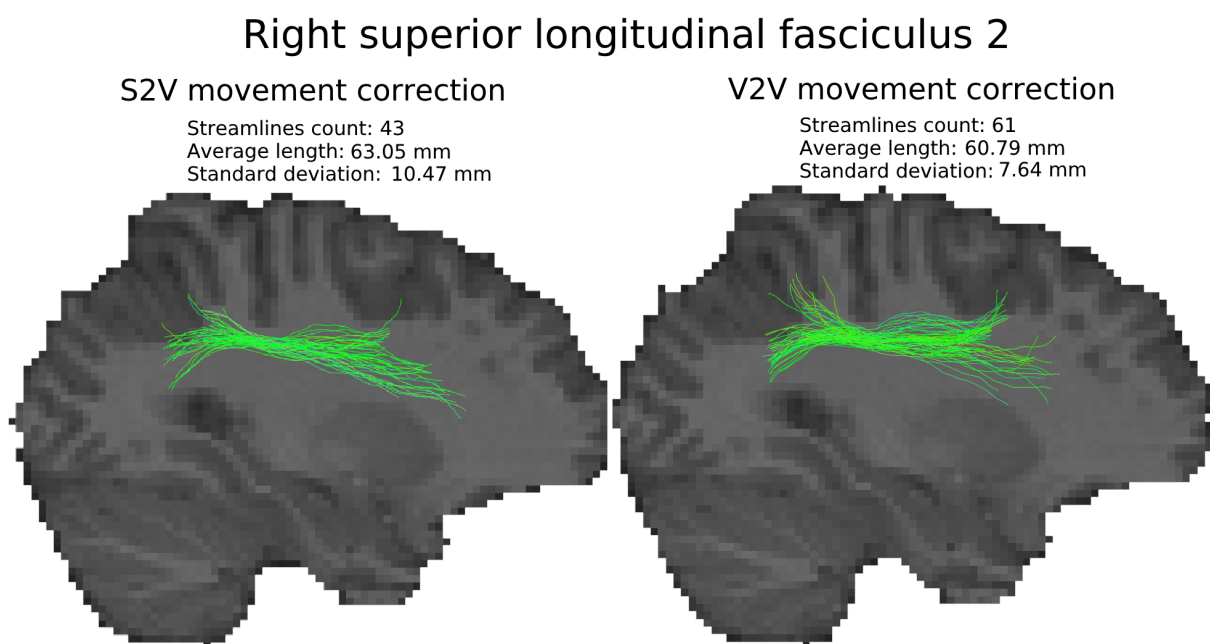


Figure 3.10: Comparison of the effect of movement correction on **probabilistic** tracking of the right superior longitudinal fasciculus on the subject with **mild** slice-to-volume movement artifact.

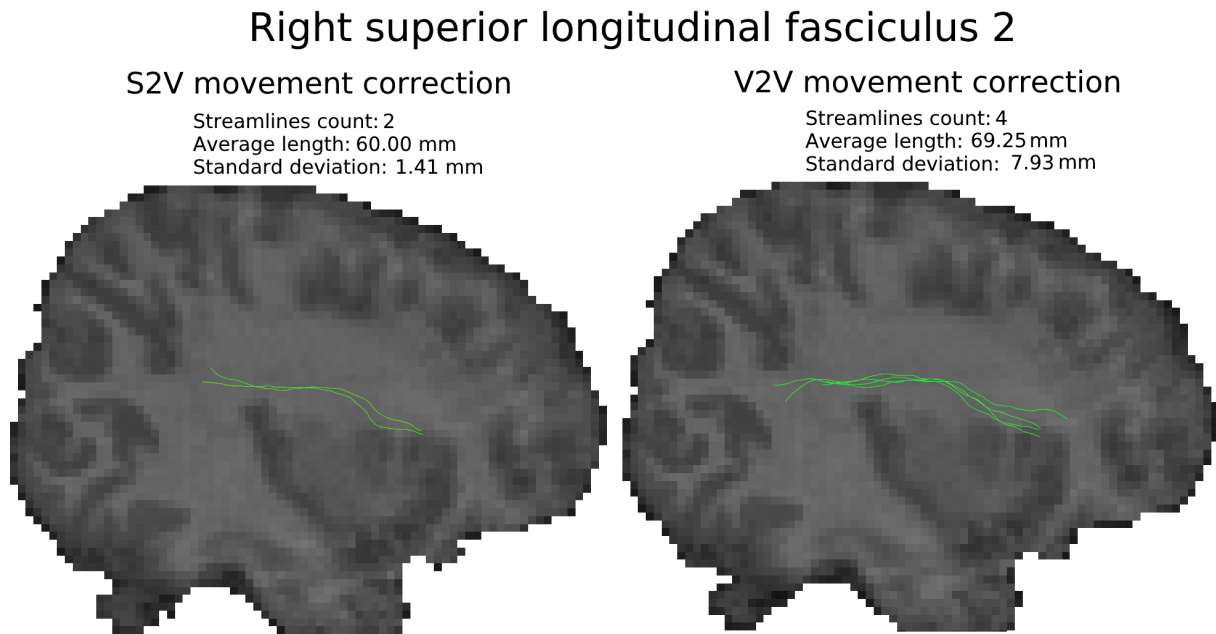


Figure 3.11: Comparison of the effect of movement correction on **probabilistic** tracking of the right superior longitudinal fasciculus on the subject with **severe** slice-to-volume movement artifact.

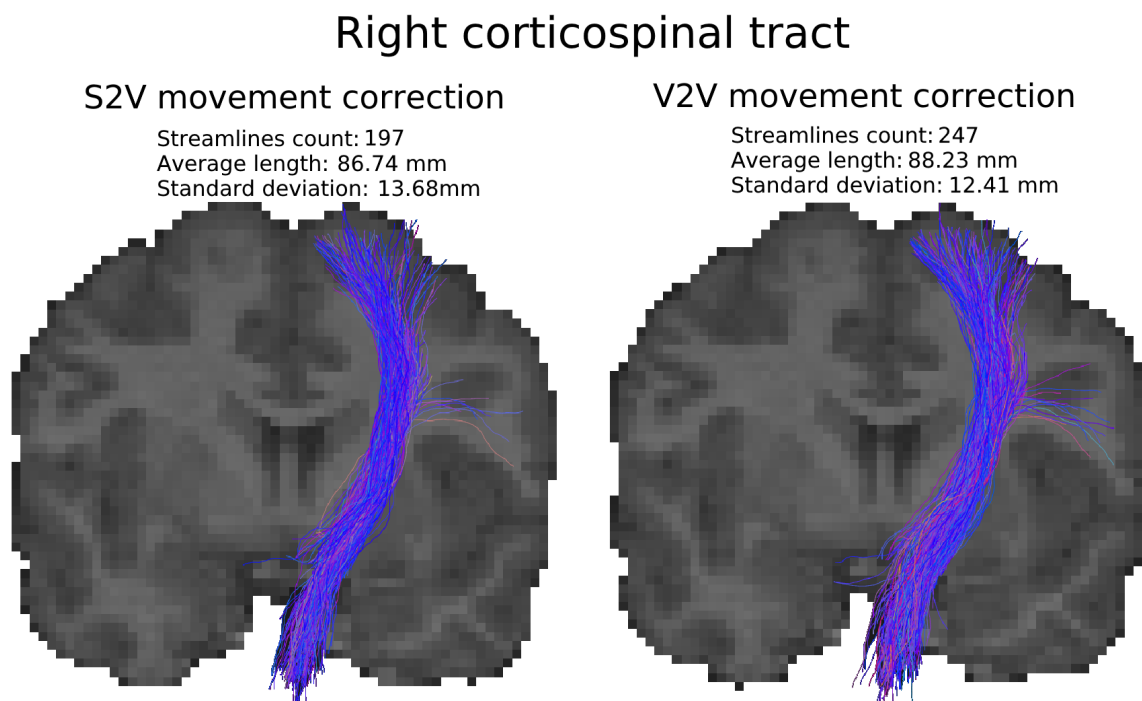


Figure 3.12: Comparison of the effect of movement correction on **probabilistic** tracking of the right corticospinal tract on the subject with **mild** slice-to-volume movement artifact.

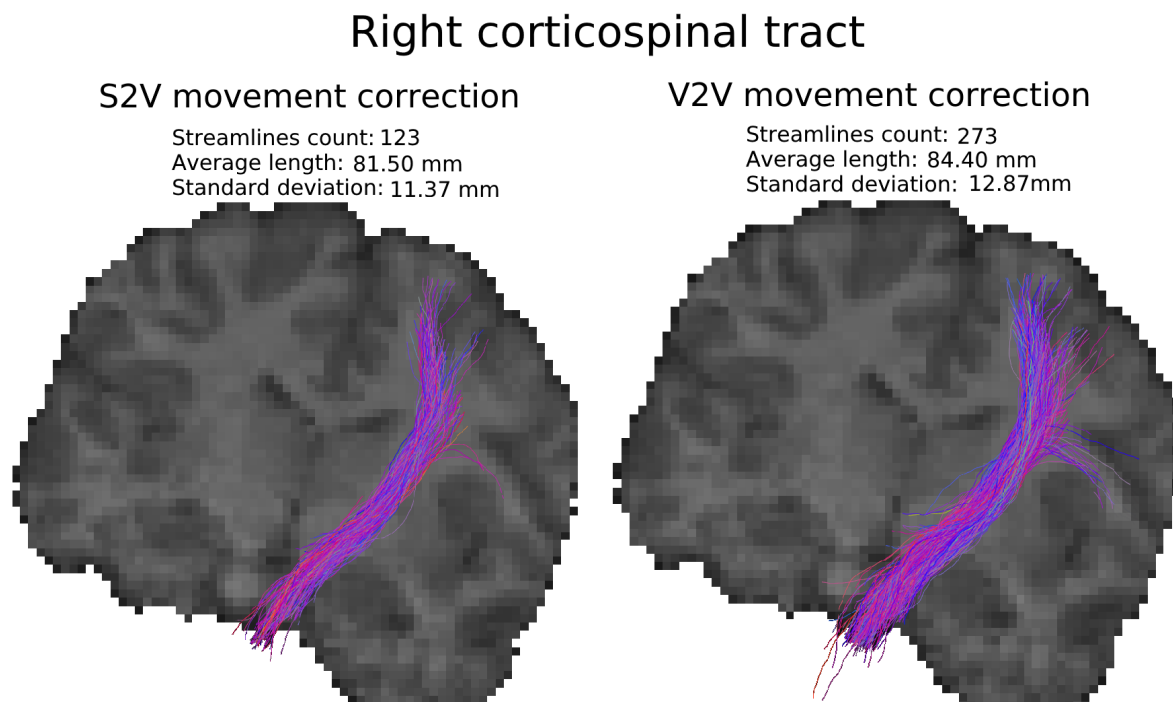


Figure 3.13: Comparison of the effect of movement correction on **probabilistic** tracking of the right corticospinal tract on the subject with **severe** slice-to-volume movement artifact.

Discussion : The number of streamlines found by the S2V motion correction decreases for both subjects for every tracts. This decrease can be explained by the increase of voxel isotropy already highlighted in the group-wise FA comparison.

For the subject with severe movement artifacts, the average length and number of streamlines decrease is more important than with the subject with mild motion artifacts. This could be due to the increased number of interpolations performed by the custom S2V Eddy motion correction algorithm when dealing with more severe motions.

Visual inspection of the different tracts seems to indicate a decrease in the quality of the different tracts found.

3.5 Final discussion and perspectives

Gradient cycling decreases the thermal load by interleaving slices of different diffusion gradients. However, it introduces new challenges regarding the slice-to-volume motion artifact correction compared to sequential EPI acquisitions. Intertwined slices of different diffusion gradients even cause small inter-volume motion to manifest as a slice-to-volume motion in all reconstructed volumes.

The temporal correlation of motion within a reconstructed volume is considerably lowered with gradient cycling. Furthermore, volume wise slice specifications are required, which is not supported by Eddy. As a result, modifications have been applied to restore Eddy slice-to-volume motion correction. The motion regularization parameter has been decreased and the order of the DCT expansion has been maximized. Moreover, the Eddy source code has been edited to accept volume wise slice specifications as input. As a result, S2V motion correction can be performed on datasets acquired with gradient cycling enabled.

The validation showed visual improvements and less artifacts on b0 volumes with S2V motion correction compared to V2V motion correction, suggesting that the slice-to-volume motion is partially corrected. However, neither group-wise FA comparison or tracts analyzes, issued from tractography, validated the consistency of the motion correction, opening the path for future works. The validation of this algorithm is thus not conclusive and no evidence could be found to highlight the benefits of the implemented Eddy modifications although visual improvements do suggest that intra-volume realignments are successfully performed. The decrease of performances in terms of FA and found streamlines indicates that further investigation should be done concerning the volume realignment method.

The results of the custom S2V motion correction opens a path for future improvements. Sets of volumes severely affected by S2V motion artifacts could be removed to prevent error propagation. Unfortunately, this will have limited results when gradient cycling is performed on the full set of volumes in one block. A small movement at any time during the acquisition will result in S2V artifacts affecting all the volumes. In this context, two possible improvements are suggested below. First, a possible more complete modification of Eddy FSL is presented and second an alternative acquisition protocol is proposed.

Additional modifications of Eddy FSL : The approach undertook in section 3.2 is a suboptimal method that considers the slices in each volume to be acquired differently and that circumvents the lack of temporal correlation between the slices by reducing the motion regularization and increasing the degrees of freedom. However, a better and more optimal approach would be to estimate the motion in the correct temporal order across the volumes without changing the regularization or degrees of freedom. The idea is that instead of estimating parameters $\mathbf{r}(t_n)$ for each volume, the algorithm would estimate motion parameters $\mathbf{r}(t_n)$ for each pass. A pass is defined as a measure (=TR) of a complete physical volume across the applied diffusion gradients. A pass is therefore the volume acquired in the temporal order before reconstruction and encompass slices acquired with different b-values.

By going back to the Eddy algorithm explained in section 2.2.2.5, this modification would require 'only' to change the way the motion parameters are applied in the case of slice-to-volume motion correction. The parameters $\mathbf{r}(t_n)$ would estimate the motion for a pass instead of the motion per reconstructed volume. In Fig.3.14 are compared the classical Eddy algorithm used to apply the motion parameters with the suboptimal approach of section 3.2 and the optimal method described here. Once the mapping between the coordinates is obtained, the interpolation and the other Eddy steps can be performed the same way. When no gradient cycling is used, a pass corresponds to a classical volume with a single b-value and the method is equivalent to the classical eddy algorithm.

Alternative acquisition protocol : Even though a better algorithm can be written, some limitations still persist. Slice-to-volume motion correction is inherently difficult due to the low temporal correlation of slices contained within a reconstructed volume. When brain rotation occurs during the acquisition, the correction is extremely difficult since rotations render interpolation between slices almost impossible. Moreover, rotations create inconsistency in the gradient table of the reconstructed volume. Indeed, only one b-vector is provided per reconstructed volume while slices experiencing different rotation artifacts should have slightly different b-vectors.

```

(A) for each volume  $i$  do
    for each  $t_n$  do
         $P = P(r_i(t_n))$ ;
        for each voxel  $x \in \text{slices } s \text{ in } Z(t_n)$  do
             $\begin{bmatrix} x' \\ 1 \end{bmatrix} = P(r_i(t_n)) \begin{bmatrix} x \\ 1 \end{bmatrix}$ 
        end
    end
end

(B) for each volume  $i$  do
    for each  $t_n$  do
         $P = P(r_i(t_n))$ ;
        for each voxel  $x \in \text{slices } s \text{ in } Z_i(t_n)$  do
             $\begin{bmatrix} x' \\ 1 \end{bmatrix} = P(r_i(t_n)) \begin{bmatrix} x \\ 1 \end{bmatrix}$ 
        end
    end
end

(C) for each complete pass  $i$  do
    for each  $t_n$  do
         $P = P(r_i(t_n))$ ;
        for each voxel  $x \in \text{slices } s \text{ in } Z(t_n)$  do
             $\begin{bmatrix} x' \\ 1 \end{bmatrix} = P(r_i(t_n)) \begin{bmatrix} x \\ 1 \end{bmatrix}$ 
        end
    end
end

```

Figure 3.14: Pseudocode of the algorithm used to apply the motion parameters. (A) **Classical Eddy approach**, for each volume and for each discrete time step t_n , the rigid body transform is applied to all slices s acquired at t_n . The motion parameters correspond to specific volumes and the slices in each volume are acquired in the same way i.e single $Z(t_n)$ function (B) **Suboptimal approach**, for each volume and for each discrete time step t_n , the rigid body transform is applied to all slices s acquired at t_n . The motion parameters correspond to specific volumes and the slices in each volume are acquired in different orders due to the gradient cycling i.e. multiple $Z_i(t_n)$ functions. (C) **Optimal approach**, for each pass and for each discrete time step t_n , the rigid body transform is applied to all slices s acquired at t_n . The motion parameters correspond to specific passes and the slices in each pass are acquired in the same way i.e single $Z(t_n)$ function. The modifications compared to the classical Eddy algorithm are highlighted in red.

A novel EPI acquisition sequence featuring gradient cycling is proposed by (Hutter et al., 2018a) to overcome these issues. Instead of applying gradient cycling over all volumes, slice interleaving is performed over superblock containing low and high b-values. Each superblock consists of $L \cdot N_s$ slices with L chosen diffusion encodings. Within a superblock, slices are intertwined as with the conventional gradient cycling approach (see Fig.3.15). This change of strategy permits to considerably increase the temporal correlation between slices of the same reconstructed volumes. Moreover, superblock ensures that complete volumes of encoded data are achieved even if the scan is interrupted or abandoned. Finally, if a superblock display a lot of motion, this superblock could be discarded to prevent error propagation as suggested in the discussion above.

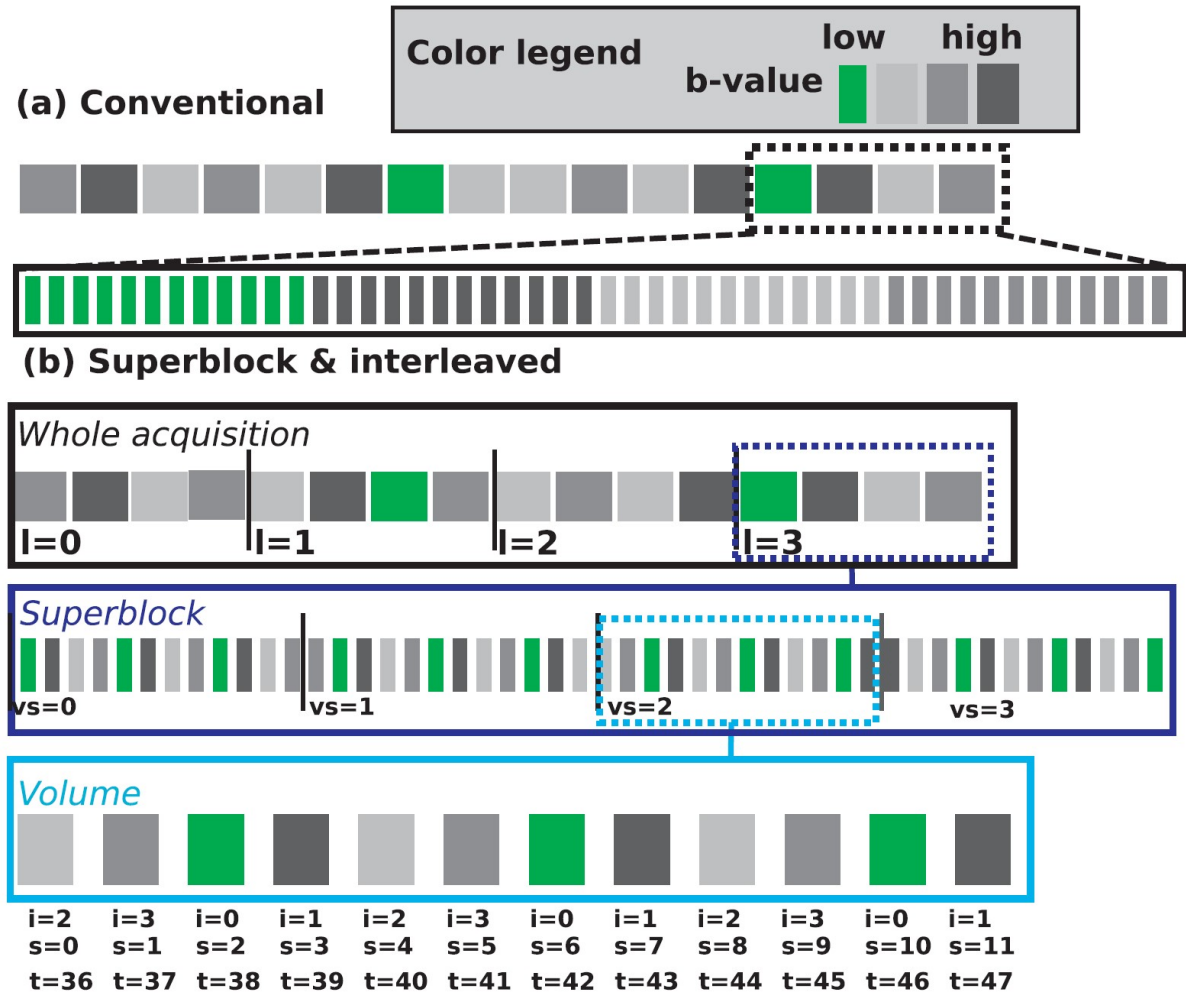


Figure 3.15: The structure of the sequential EPI sequence (in (a)) and the gradient cycling with superblock and interleaving EPI sequence (in (b)) are depicted. In (a) the whole acquisition is shown together with a zoom into the last four volumes in the second row. In (b), a whole acquisition is shown in the first row. The second row shows a zoom into the superblock $l = 3$. Finally, in the third row, volume $vs = 2$ within this superblock is shown together with all relevant parameters where i is the diffusion gradient index, s is the slice index, t is the temporal index. From (Hutter et al., 2018a).

Conclusion

In this Master’s thesis, a complete dMRI pipeline has been developed based on state of the art diffusion weighted processing algorithms. Throughout the report, this pipeline has been validated using in vivo and groundtruth simulated diffusion-weighted images. The library has proven to properly correct the numerous artifacts affecting diffusion data namely noise, gibbs ringing, EC distortions, susceptibility distortions, motion and intensity bias. These corrections have shown to significantly improve the fitting of the diffusion models especially in the white matter which contains the microstructure information of interest in dMRI. Moreover, the diffusion models exhibited increased sensitivity suggesting that the preprocessing helps to alleviate non biological effects.

In addition to these preprocessing advantages, ElikoPy is able to directly compute various microstructural metrics and perform voxel wise, atlas-based region wise and volumetric statistics between different groups. Overall, ElikoPy has demonstrated to be a powerful tool to process large databases by providing a standardized framework that ensures consistency and reproducibility across studies. Significant efforts have been made to provide a pipeline that is easy to install (thanks to its containerization), compatible with the scheduler present in high-performance computing centers and usable with minimal knowledge in DW-MRI processing. ElikoPy allows to account for data with disparate acquisition parameters making this pipeline applicable to DW images from multi-site, multi-scanner studies with non uniform data formats and acquisition protocols. ElikoPy is fully open-source (AGPL3 License) and available publicly on Github with an extensive documentation and API.

Finally, the challenges of gradient cycling for motion artifacts have been explored and a novel motion correction framework has been introduced to allow for more flexible slice realignments that can accommodate any slice acquisition order. The validation of this algorithm has not been conclusive and little evidence could be found to highlight the benefits of the implemented modifications in a gradient cycling scenario. In this context, more complete modifications of Eddy FSL and alternative acquisition protocols have been suggested.

The work presented in this Master’s thesis has already been used for studies involving dyslexic children, meditator and hemiplegic children. The dyslexia study has led to the publication of a conference paper submitted to the 2021 Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM) (Delinte et al., 2021).

Bibliography

- J. Y. e. a. A. Haouimi. Aliasing on mri. <https://radiopaedia.org/articles/aliasing-on-mri>, 2009.
- B. Ades-Aron, J. Veraart, P. Kochunov, S. McGuire, P. Sherman, E. Kellner, D. S. Novikov, and E. Fieremans. Evaluation of the accuracy and precision of the diffusion parameter estimation with gibbs and noise removal pipeline. *NeuroImage*, 183:532–543, 2018.
- S. Aja-Fernández, V. Brion, and A. Tristán-Vega. Effective noise estimation and filtering from correlated multiple-coil mr data. *Magnetic resonance imaging*, 31(2):272–285, 2013.
- J. Andersson and S. Skare. Image distortion and its correction in diffusion mri. *Diffusion MRI: theory, methods, and applications*, pages 285–302, 2010.
- J. L. Andersson and S. N. Sotiropoulos. Non-parametric representation and prediction of single-and multi-shell diffusion-weighted mri data using gaussian processes. *Neuroimage*, 122:166–176, 2015.
- J. L. Andersson and S. N. Sotiropoulos. An integrated approach to correction for off-resonance effects and subject movement in diffusion mr imaging. *Neuroimage*, 125:1063–1078, 2016.
- J. L. Andersson and S. N. Sotiropoulos. Fsl - eddy user guide. https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/eddy/UsersGuide#A--s2v_lambda, 2019.
- J. L. Andersson, S. Skare, and J. Ashburner. How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging. *Neuroimage*, 20(2):870–888, 2003.
- J. L. Andersson, M. S. Graham, E. Zsoldos, and S. N. Sotiropoulos. Incorporating outlier detection and replacement into a non-parametric framework for movement and distortion correction of diffusion mr images. *Neuroimage*, 141:556–572, 2016.
- J. L. Andersson, M. S. Graham, I. Drobnjak, H. Zhang, N. Filippini, and M. Bastiani. Towards a comprehensive framework for movement and distortion correction of diffusion mr images: Within volume movement. *Neuroimage*, 152:450–466, 2017.
- J. Ashburner and K. J. Friston. Voxel-based morphometry—the methods. *Neuroimage*, 11(6):805–821, 2000.
- B. B. Avants, N. Tustison, and G. Song. Advanced normalization tools (ants). *Insight j*, 2(365):1–35, 2009.

- B. B. Avants, N. J. Tustison, J. Wu, P. A. Cook, and J. C. Gee. An open source multivariate framework for n-tissue segmentation with evaluation on public data. *Neuroinformatics*, 9(4):381–400, 2011.
- N. Barnea-Goraly, V. Menon, M. Eckert, L. Tamm, R. Bammer, A. Karchemskiy, C. C. Dant, and A. L. Reiss. White matter development during childhood and adolescence: a cross-sectional diffusion tensor imaging study. *Cerebral cortex*, 15(12):1848–1854, 2005.
- A. W. Barritt, M. C. Gabel, M. Cercignani, and P. N. Leigh. Emerging magnetic resonance imaging techniques and analysis methods in amyotrophic lateral sclerosis. *Frontiers in neurology*, 9:1065, 2018.
- P. J. Basser and C. Pierpaoli. Microstructural and physiological features of tissues elucidated by quantitative-diffusion-tensor mri. *Journal of magnetic resonance*, 213(2): 560–570, 2011.
- P. J. Basser, J. Mattiello, and D. LeBihan. Estimation of the effective self-diffusion tensor from the nmr spin echo. *Journal of Magnetic Resonance, Series B*, 103(3):247–254, 1994.
- P. J. Basser, S. Pajevic, C. Pierpaoli, J. Duda, and A. Aldroubi. In vivo fiber tractography using dt-mri data. *Magnetic resonance in medicine*, 44(4):625–632, 2000.
- M. Bastiani, M. Cottaar, S. P. Fitzgibbon, S. Suri, F. Alfaro-Almagro, S. N. Sotiropoulos, S. Jbabdi, and J. L. Andersson. Automated quality control for within and between studies diffusion mri data using a non-parametric framework for movement and distortion correction. *Neuroimage*, 184:801–812, 2019.
- Y. A. T. Bhagat. *Diffusion Tensor Imaging of the Human Brain*. ProQuest, 2008.
- K. T. Block, M. Uecker, and J. Frahm. Suppression of mri truncation artifacts using total variation constrained data extrapolation. *International journal of biomedical imaging*, 2008, 2008.
- F. L. Bookstein. “voxel-based morphometry” should not be used with imperfectly registered images. *Neuroimage*, 14(6):1454–1462, 2001.
- L. Y. Cai, Q. Yang, C. B. Hansen, V. Nath, K. Ramadass, G. W. Johnson, B. N. Conrad, B. D. Boyd, J. P. Begnoche, L. L. Beason-Held, et al. Prequal: An automated pipeline for integrated preprocessing and quality assurance of diffusion weighted mri images. *Magnetic Resonance in Medicine*, 86(1):456–470, 2021.
- D. L. Collins, C. J. Holmes, T. M. Peters, and A. C. Evans. Automatic 3-d model-based neuroanatomical segmentation. *Human brain mapping*, 3(3):190–208, 1995.
- A. Daducci, E. J. Canales-Rodríguez, H. Zhang, T. B. Dyrby, D. C. Alexander, and J.-P. Thiran. Accelerated microstructure imaging via convex optimization (amico) from diffusion mri data. *NeuroImage*, 105:32–44, 2015.
- C. Davatzikos. Why voxel-based morphometric analysis should be used with great caution when characterizing group differences. *Neuroimage*, 23(1):17–20, 2004.

- N. Delinte, C. Gosse, L. Dricot, Q. Dessain, M. Simon, B. Macq, M. Van Reybroeck, and G. Rensonnet. Microstructural alterations in the white matter of children with dyslexia assessed by multi-fascicle diffusion compartment imaging. In *29th Annual Meeting of the International Society for Magnetic Resonance in Medicine (ISMRM)*, 2021.
- F. Dell’Acqua, L. Lacerda, M. Catani, and A. Simmons. Anisotropic power maps: a diffusion contrast to reveal low anisotropy tissues from hardi data. In *Proc Intl Soc Mag Reson Med*, volume 22, pages 29960–29967, 2014.
- R. S. Desikan, F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on mri scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.
- I. Drobnjak, D. Gavaghan, E. Süli, J. Pitt-Francis, and M. Jenkinson. Development of a functional magnetic resonance imaging simulator for modeling realistic rigid-body motion artifacts. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 56(2):364–380, 2006.
- B. Efron. Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American statistical association*, 78(382):316–331, 1983.
- R. H. Fick, D. Wassermann, and R. Deriche. The dmipy toolbox: Diffusion mri multi-compartment modeling and microstructure recovery made easy. *Frontiers in neuroinformatics*, 13:64, 2019.
- A. Filler. Spin echo diagram. https://en.wikipedia.org/wiki/File:Spin_Echo_Diagram.jpg, 2009.
- B. Fischl. Freesurfer. *Neuroimage*, 62(2):774–781, 2012.
- F. Peeters. *Magnetic Resonance Imaging course notes*. Moodle, 2017.
- F. Frenet. Sur les courbes à double courbure. *Journal de mathématiques pures et appliquées*, pages 437–447, 1852.
- E. Garyfallidis. *Towards an accurate brain tractography*. PhD thesis, University of Cambridge, 2013.
- E. Garyfallidis, M. Brett, B. Amirbekian, A. Rokem, S. Van Der Walt, M. Descoteaux, and I. Nimmo-Smith. Dipy, a library for the analysis of diffusion mri data. *Frontiers in neuroinformatics*, 8:8, 2014.
- K. J. Gorgolewski, T. Auer, V. D. Calhoun, R. C. Craddock, S. Das, E. P. Duff, G. Flandin, S. S. Ghosh, T. Glatard, Y. O. Halchenko, et al. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific data*, 3(1):1–9, 2016.
- M. S. Graham, I. Drobnjak, and H. Zhang. Realistic simulation of artefacts in diffusion mri for validating post-processing correction techniques. *NeuroImage*, 125:1079–1094, 2016.

- M. S. Graham, I. Drobnjak, M. Jenkinson, and H. Zhang. Quantitative assessment of the susceptibility artefact and its interaction with motion in diffusion mri. *PloS one*, 12(10):e0185647, 2017.
- R. Guillemaud and M. Brady. Estimating the bias field of mr images. *IEEE Transactions on Medical imaging*, 16(3):238–251, 1997.
- D. Hernando, D. Karampinos, K. King, J. Haldar, S. Majumdar, J. Georgiadis, and Z.-P. Liang. Removal of olefinic fat chemical shift artifact in diffusion mri. *Magnetic resonance in medicine*, 65(3):692–701, 2011.
- K. Hua, J. Zhang, S. Wakana, H. Jiang, X. Li, D. S. Reich, P. A. Calabresi, J. J. Pekar, P. C. van Zijl, and S. Mori. Tract probability maps in stereotaxic spaces: analyses of white matter anatomy and tract-specific quantification. *Neuroimage*, 39(1):336–347, 2008.
- T. Huisman. Diffusion-weighted and diffusion tensor imaging of the brain, made easy. *Cancer Imaging*, 10(1A):S163, 2010.
- J. Hutter, D. J. Christiaens, T. Schneider, L. Cordero-Grande, P. J. Slator, M. Deprez, A. N. Price, J.-D. Tournier, M. Rutherford, and J. V. Hajnal. Slice-level diffusion encoding for motion and distortion correction. *Medical image analysis*, 48:214–229, 2018a.
- J. Hutter, M. Nilsson, D. Christiaens, T. Schneider, A. N. Price, J. V. Hajnal, and F. Szczepankiewicz. Highly efficient diffusion mri by slice-interleaved free-waveform imaging (sifi). In *Proceedings of the 26th Annual Meeting of ISMRM, Paris, France*, 2018b.
- S. Jaume, N. Rannou, N. Tustison, and R. Kikinis. Modules:mribiasfieldcorrection-documentation-3.6. <https://www.slicer.org/wiki/Modules:MRIBiasFieldCorrection-Documentation-3.6>, 2011.
- M. Jenkinson, M. Pechaud, S. Smith, et al. Bet2: Mr-based estimation of brain, skull and scalp surfaces. In *Eleventh annual meeting of the organization for human brain mapping*, volume 17, page 167. Toronto., 2005.
- M. Jenkinson, C. F. Beckmann, T. E. Behrens, M. W. Woolrich, and S. M. Smith. Fsl. *Neuroimage*, 62(2):782–790, 2012.
- B. Jeurissen, A. Leemans, J.-D. Tournier, D. K. Jones, and J. Sijbers. Investigating the prevalence of complex fiber configurations in white matter tissue with diffusion magnetic resonance imaging. *Human brain mapping*, 34(11):2747–2766, 2013.
- P. Jezzard and R. S. Balaban. Correction for geometric distortion in echo planar images from b0 field variations. *Magnetic resonance in medicine*, 34(1):65–73, 1995.
- H. Johansen-Berg and T. E. Behrens. *Diffusion MRI*. Elsevier, 2013.
- D. K. Jones, M. R. Symms, M. Cercignani, and R. J. Howard. The effect of filter size on vbm analyses of dt-mri data. *Neuroimage*, 26(2):546–554, 2005.

- J. Juntu, J. Sijbers, D. Van Dyck, and J. Gielen. Bias field correction for mri images. In *Computer Recognition Systems*, pages 543–551. Springer, 2005.
- P. Kalavathi and V. S. Prasath. Methods on skull stripping of mri head scan images—a review. *Journal of digital imaging*, 29(3):365–379, 2016.
- B. Kastler, P. Anstett, B. Kastler, and D. Vetter. *Comprendre l’IRM*. Elsevier, 2011.
- E. Kellner, B. Dhital, V. G. Kiselev, and M. Reiser. Gibbs-ringing artifact removal based on local subvoxel-shifts. *Magnetic resonance in medicine*, 76(5):1574–1581, 2016.
- F. Knoll, K. Bredies, T. Pock, and R. Stollberger. Second order total generalized variation (tgv) for mri. *Magnetic resonance in medicine*, 65(2):480–491, 2011.
- C. G. Koay and P. J. Basser. Analytically exact correction scheme for signal extraction from noisy magnitude mr signals. *Journal of magnetic resonance*, 179(2):317–322, 2006.
- B. Kolb and I. Q. Whishaw. *Fundamentals of human neuropsychology*. Macmillan, 2009.
- R. Kupers, L. Dricot, S. Darkner, S. Laureys, A. De Volder, M. Simon, Q. Dessain, and P. de Timary. A multimodal imaging study of structural and functional brain changes in long-term meditators. 2021.
- S. K. Lam, A. Pitrou, and S. Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6, 2015.
- D. Le Bihan, J.-F. Mangin, C. Poupon, C. A. Clark, S. Pappata, N. Molko, and H. Chabriet. Diffusion tensor imaging: concepts and applications. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 13(4):534–546, 2001.
- D. Le Bihan, C. Poupon, A. Amadon, and F. Lethimonnier. Artifacts and pitfalls in diffusion mri. *Journal of Magnetic Resonance Imaging: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 24(3):478–488, 2006.
- X. Li, P. S. Morgan, J. Ashburner, J. Smith, and C. Rorden. The first step for neuroimaging data analysis: Dicom to nifti conversion. *Journal of neuroscience methods*, 264:47–56, 2016.
- C. Luo, M. Li, R. Qin, H. Chen, D. Yang, L. Huang, R. Liu, Y. Xu, F. Bai, and H. Zhao. White matter microstructural damage as an early sign of subjective cognitive decline. *Frontiers in aging neuroscience*, 11:378, 2020.
- J. V. Manjón, P. Coupé, L. Concha, A. Buades, D. L. Collins, and M. Robles. Diffusion weighted image denoising using overcomplete local pca. *PloS one*, 8(9):e73021, 2013.
- I. I. Maximov, D. Alnæs, and L. T. Westlye. Towards an optimised processing pipeline for diffusion magnetic resonance imaging data: Effects of artefact corrections on diffusion metrics and their age associations in uk biobank. *Human brain mapping*, 40(14):4146–4162, 2019.

- J. Mazziotta, A. Toga, A. Evans, P. Fox, J. Lancaster, K. Zilles, R. Woods, T. Paus, G. Simpson, B. Pike, et al. A probabilistic atlas and reference system for the human brain: International consortium for brain mapping (icbm). *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 356(1412):1293–1322, 2001.
- A. McQuarrie, R. Shumway, and C.-L. Tsai. The model selection criterion aicu. *Statistics & probability letters*, 34(3):285–292, 1997.
- D. McRobbie, E. Moore, M. Graves, and M. Prince. Mri from picture to proton. *Ohmsha, Ltd*, 2004.
- T. V. ME Mercredi. Inferring axon diameter sizes using monte carlo simulations of magnetic resonance oscillating gradient spin echo sequences. <https://slideplayer.com/slide/4663152/>, 2016.
- S. Mori, S. Wakana, P. C. Van Zijl, and L. Nagae-Poetscher. *MRI atlas of human white matter*. Elsevier, 2005.
- I. Oguz, M. Farzinfar, J. Matsui, F. Budin, Z. Liu, G. Gerig, H. J. Johnson, and M. A. Styner. Dtiprep: quality control of diffusion-weighted images. *Frontiers in neuroinformatics*, 8:4, 2014.
- N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- D. Perrone, J. Aelterman, A. Pižurica, B. Jeurissen, W. Philips, and A. Leemans. The effect of gibbs ringing artifacts on measures derived from diffusion mri. *Neuroimage*, 120:441–455, 2015.
- F. Pesarin and L. Salmaso. *Permutation tests for complex data: theory, applications and software*. John Wiley & Sons, 2010.
- S. Plein, J. Greenwood, and J. P. Ridgway. *Cardiovascular MR manual*. Springer, 2011.
- G. Rensonnet. In vivo diffusion magnetic resonance imaging of the white matter microstructure from dictionaries generated by monte carlo simulations: development and validation. Technical report, EPFL, 2019.
- G. Rensonnet, B. Scherrer, G. Girard, A. Jankovski, S. K. Warfield, B. Macq, J.-P. Thiran, and M. Taquet. Towards microstructure fingerprinting: Estimation of tissue properties from a dictionary of monte carlo diffusion mri simulations. *NeuroImage*, 184:964–980, 2019.
- D. R. Roalf, M. Quarmley, M. A. Elliott, T. D. Satterthwaite, S. N. Vandekar, K. Ruparel, E. D. Gennatas, M. E. Calkins, T. M. Moore, R. Hopson, et al. The impact of quality assurance assessment on diffusion tensor imaging outcomes in a large-scale population-based cohort. *Neuroimage*, 125:903–919, 2016.
- G. D. Ruxton. The unequal variance t-test is an underused alternative to student’s t-test and the mann–whitney u test. *Behavioral Ecology*, 17(4):688–690, 2006.
- B. Scherrer, M. Taquet, M. Sahin, S. P. Prabhu, and S. K. Warfield. Quantitative evaluation of biophysical models of the diffusion with in vivo data by assessment of the generalization error. In *Proc. Intl. Soc. Mag. Reson. Med*, volume 22, page 2648, 2014.

- B. Scherrer, A. Schwartzman, M. Taquet, M. Sahin, S. P. Prabhu, and S. K. Warfield. Characterizing brain tissue by assessment of the distribution of anisotropic microstructural environments in diffusion-compartment imaging (diamond). *Magnetic resonance in medicine*, 76(3):963–977, 2016.
- K. G. Schilling, J. Blaber, Y. Huo, A. Newton, C. Hansen, V. Nath, A. T. Shafer, O. Williams, S. M. Resnick, B. Rogers, et al. Synthesized b0 for diffusion distortion correction (synb0-disco). *Magnetic resonance imaging*, 64:62–70, 2019.
- D. W. Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.
- M. Simon and Q. Dessain. Elikopy. <https://github.com/hyedryn/elikopy>, 2021a.
- M. Simon and Q. Dessain. Eddy with gradient cycling. https://github.com/Hyedryn/eddy_ge, 2021b.
- S. M. Smith. Fast robust automated brain extraction. *Human brain mapping*, 17(3):143–155, 2002.
- S. M. Smith, M. Jenkinson, H. Johansen-Berg, D. Rueckert, T. E. Nichols, C. E. Mackay, K. E. Watkins, O. Ciccarelli, M. Z. Cader, P. M. Matthews, et al. Tract-based spatial statistics: voxelwise analysis of multi-subject diffusion data. *Neuroimage*, 31(4):1487–1505, 2006.
- R. Tao, P. T. Fletcher, S. Gerber, and R. T. Whitaker. A variational image-based approach to the correction of susceptibility artifacts in the alignment of diffusion weighted and structural mri. In *International Conference on Information Processing in Medical Imaging*, pages 664–675. Springer, 2009.
- M. Taquet, A. Jankovski, G. Rensonnet, D. Jacobs, A. Des Rieux, B. Macq, S. K. Warfield, and B. Scherrer. Extra-axonal restricted diffusion as an in-vivo marker of reactive microglia. *Scientific reports*, 9(1):1–10, 2019.
- J.-D. Tournier, F. Calamante, and A. Connelly. Robust determination of the fibre orientation distribution in diffusion mri: non-negativity constrained super-resolved spherical deconvolution. *Neuroimage*, 35(4):1459–1472, 2007.
- J.-D. Tournier, R. Smith, D. Raffelt, R. Tabbara, T. Dhollander, M. Pietsch, D. Christiaens, B. Jeurissen, C.-H. Yeh, and A. Connelly. Mrtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *NeuroImage*, 202:116137, 2019.
- N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee. N4itk: improved n3 bias correction. *IEEE transactions on medical imaging*, 29(6):1310–1320, 2010.
- J. Veraart, D. S. Novikov, D. Christiaens, B. Ades-Aron, J. Sijbers, and E. Fieremans. Denoising of diffusion mri using random matrix theory. *Neuroimage*, 142:394–406, 2016.
- P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett,

- J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, Í. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.
- A. M. Winkler, G. R. Ridgway, M. A. Webster, S. M. Smith, and T. E. Nichols. Permutation inference for the general linear model. *Neuroimage*, 92:381–397, 2014.
- M. Wu, L.-C. Chang, L. Walker, H. Lemaitre, A. S. Barnett, S. Marengo, and C. Pierpaoli. Comparison of epi distortion correction methods in diffusion tensor mri using a novel framework. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 321–329. Springer, 2008.
- S. Yoshida, K. Oishi, A. V. Faria, and S. Mori. Diffusion tensor imaging of normal brain development. *Pediatric radiology*, 43(1):15–27, 2013.
- P. A. Yushkevich, Y. Gao, and G. Gerig. Itk-snap: an interactive tool for semi-automatic segmentation of multi-modality biomedical images. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 3342–3345. IEEE, 2016.
- M. Zaitsev, J. Hennig, and O. Speck. Point spread function mapping with parallel imaging techniques and high acceleration factors: Fast, robust, and flexible method for echo-planar imaging distortion correction. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 52(5): 1156–1166, 2004.
- H. Zhang, T. Schneider, C. A. Wheeler-Kingshott, and D. C. Alexander. Noddi: practical in vivo neurite orientation dispersion and density imaging of the human brain. *Neuroimage*, 61(4):1000–1016, 2012.
- Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.

Appendices

Appendix A

Outputed metrics for each microstructural model

A.1 DTI Metrics

- Fractional anisotropy (FA).
- Axial diffusivity (AD).
- Radial diffusivity (RD).
- Mean diffusivity (MD).
- Colored FA (fargb).
- Residual (residual).
- Eigenvectors of the diffusion tensor (evecs).
- Eigenvalues of the diffusion tensor (evals).
- Diffusion Tensor (dtensor).

A.2 DIAMOND Metrics

- Tensor orientations of the fiber population 0 (t0).
- Tensor orientations of the fiber population 1 (t1).
- Residuals (residuals).
- Intermediary step of the t0 output (mtm_t0).
- Intermediary step of the t1 output (mtm_t1).
- Intermediary step of the fractions output (mtm_fractions).
- Volume with a null b-value (b0).
- DTI estimate (dti).
- Automose model selection map. Gray and white matter correspond to positive values and CSF to negative values (aicu).
- Fraction of voxel attributed to each compartment (fractions).
- Shape parameters of the mv- Γ distribution, Homogeneity index (kappa).
- Log of kappa (logkappa).
- The heterogeneity indexes defined as $HEI = \frac{2}{\pi} \arctan\left(\frac{1}{\kappa}\right)$ (hei).
- Number of fascicles by voxel (mosemap).

A.3 NODDI Metrics

- Thresholded intra-cellular volume fraction ν_{ic} (icvf).
- Fiber orientation dispersion index (odi).
- Mean of the watson distribution of the Intra-cellular model (mu).
- Fiber bundles volume fraction (fbundle).
- Extra-cellular volume fraction (fextra).
- Intra-cellular volume fraction (fintra).
- Free water volume fraction (fiso).
- Mean squared error (mse).
- R squared (R2).

A.4 Microstructure Fingerprinting Metrics

- Extra-axonal diffusivity of fascicle 0 (DIFF_ex_f0).
- Extra-axonal diffusivity of fascicle 1 (DIFF_ex_f1).
- Total extra-axonal diffusivity (DIFF_ex_tot).
- Volume fraction of cerebrospinal fluid (frac_csf).
- Volume fractions of fascicle 0 (frac_f0).
- Volume fractions of fascicle 1 (frac_f1).
- Fiber volume fractions of the fascicle 0 (fvf_f0).
- Fiber volume fractions of the fascicle 1 (fvf_f1).
- Total fiber volume fraction of all fascicles (fvf_tot).
- Mask (M0).
- Mean squared error (MSE).
- Peak map fascicle 0 (peak_f0).
- Peak map fascicle 1 (peak_f1).
- R squared (coefficient of determination, square of Pearson) (R2).

Appendix B

Parameters used during data processing

B.1 PreQual and DESIGNER parameters during simulated data processing

The ElikoPy pipeline has been compared to the PreQual and DESIGNER pipeline in the chapter 2.4.

The DESIGNER pipeline was launched using the command :

```
1 $ python3.6 DESIGNER.py <input_file> <output_file> -denoise -  
    extent 5,5,5 -degibbs -rician -eddy -rpe_none -pe_dir AP -  
    fit_constraints 0,1,1 -outliers -akc
```

During the execution, the following steps were enable :

- Thermal noise correction.
- Gibbs ringing correction.
- Eddy current induced spatial distortions and motion correction.
- Rician biases correction using the methods of moments.
- Signal outliers removal.

The PreQual pipeline was launched through a singularity container using the command :

```
1 $ singularity run -e --contain -B <input_dir>:/INPUTS -B <  
    output_dir>:/OUTPUTS /DATA/prequal_sandbox/ j --num_threads  
    8
```

During the execution, the following steps were enable :

- Thresholding of all b-values such that values less than the `-bval_threshold` parameter are 0.
- Checking that all b-vectors are unit normalized and all b-values greater than zero have associated non zero b-vectors.
- Denoising images prior to the preprocessing using Marchenko-Pastur PCA implemented in MRTrix3.
- Images intensity normalization prior to preprocessing by maximizing the intra-mask intensity-histogram intersections between the averaged b0s of the scans.
- Eddy current induced spatial distortions and motion correction.

B.2 Default parameters of ElikoPy’s microstructural metrics

In this section are available the python codes used to preprocess the datasets used throughout this Master’s thesis.

```

1 import elikopy
2
3 f_path="/CECI/proj/pilab/dyslexia_delilab/data_v2/"
4 dic_path="/home/users/q/d/qdessain/Script_python/fixed_rad_dist
   .mat"
5
6 study = elikopy.core.Elikopy(f_path, slurm=True, slurm_email='
   quentin.dessain@student.uclouvain.be', cuda=True)
7
8 study.patient_list()
9
10 study.preproc(cuda_name="eddy_cuda_10.1", cuda=True, eddy=True,
   topup=False, denoising=True, reslice=True, gibbs=True,
   biasfield=True)
11
12 study.white_mask(corr_gibbs=True, cpus=2)
13
14 study.dti()
15 study.noddi(force_brain_mask=True, cpus=4)
16 study.diamond(cpus=8)
17 study.fingerprinting(dic_path, cpus=8)
18
19 grp1=[1]
20 grp2=[2]
21
22 study.regall_FA(grp1=grp1, grp2=grp2, registration_type="-T",
   postreg_type="-S", prestats_treshold=0.2, cpus=8)
23
24 metrics={'_noddi_odi': 'noddi', '_mf_fvf_tot': 'mf', '
   _diamond_kappa': 'diamond'}
25 study.regall(grp1=grp1, grp2=grp2, metrics_dic=metrics)
26
27 study.randomise_all(randomise_numberofpermutation=10000,
   skeletonised=True, metrics_dic=metrics, regionWiseMean=True,
   cpus=4)

```

Listing B.1: Python code used to preprocess the in-vivo dataset for the result of the second chapter

```

1 import elikopy
2
3 f_path="/DATA/simulated/"
4
5 study = elikopy.core.Elikopy(f_path, slurm=True, slurm_email='
    quentin.dessain@student.uclouvain.be', cuda=True)
6
7 study.patient_list()
8
9 study.preproc(cuda_name="eddy_cuda_10.1", cuda=True, eddy=True,
    topup=True, denoising=True, reslice=False, gibbs=True,
    biasfield=True, s2v=[17,5,1,"trilinear"], olrep=[True,6,250,
    "both"])
10
11 study.dti()

```

Listing B.2: Python code used to preprocess the simulated dataset

```

1 import elikopy
2
3 f_path="/DATA/data_custom/"
4
5 dic_path="/home/users/q/d/qdessain/Script_python/fixed_rad_dist
    .mat"
6
7 study = elikopy.core.Elikopy(f_path, slurm=True, slurm_email='
    quentin.dessain@student.uclouvain.be', cuda=True)
8
9 study.patient_list()
10
11 study.preproc(cuda_name="eddy_cuda_10.1", cuda=True, eddy=True,
    topup=True, denoising=True, reslice=True, gibbs=True,
    biasfield=True)
12 study.preproc(cuda_name="eddy_cuda_10.1", cuda=True, eddy=True,
    topup=True, forceSynboDisCo=True, denoising=True, reslice=
    True, gibbs=True, biasfield=True, starting_state="topup")
13
14 study.white_mask(corr_gibbs=True, cpus=2)

```

Listing B.3: Python code used to preprocess the in-vivo subject with two phase encoding direction

```

1 import elikopy
2
3 f_path="/CECI/proj/pilab/dyslexia_delilab/data_S2V_r6/"
4 gc_slspec="/CECI/proj/pilab/dyslexia_delilab/data_S2V_r6/
   slspecs/"
5 dic_path="/home/users/q/d/qdessain/Script_python/fixed_rad_dist
   .mat"
6
7 study = elikopy.core.Elikopy(f_path, slurm=True, slurm_email='
   quentin.dessain@student.uclouvain.be', cuda=True)
8
9 study.preproc(cuda_name="eddy_cuda_10.1_gc10.1", cuda=True,
   eddy=True, topup=False, denoising=True, reslice=True, gibbs=
   True, biasfield=True, s2v=[22,7,0.1,"trilinear"], olrep=[
   True,6,250,"both"])
10
11 study.white_mask(corr_gibbs=True, cpus=2)
12
13 study.dti()
14 study.noddi(force_brain_mask=True, cpus=4)
15 study.diamond(cpus=8)
16 study.fingerprinting(dic_path, cpus=8)
17
18 grp1=[1]
19 grp2=[2]
20
21 study.regall_FA(grp1=grp1,grp2=grp2, registration_type="-T",
   postreg_type="-S", prestats_treshold=0.2, cpus=8)
22
23 metrics={'_noddi_odi':'noddi','_mf_fvf_tot':'mf','
   _diamond_kappa':'diamond'}
24 study.regall(grp1=grp1,grp2=grp2, metrics_dic=metrics)
25
26 study.randomise_all(randomise_numberofpermutation=10000,
   skeletonised=True, metrics_dic=metrics, regionWiseMean=True,
   cpus=4)

```

Listing B.4: Python code used to the in-vivo dataset with the custom S2V motion correction

Appendix C

S2V gradient cycling movement correction : Extra information

C.1 b0 comparison for subject with almost no S2V motion artifact

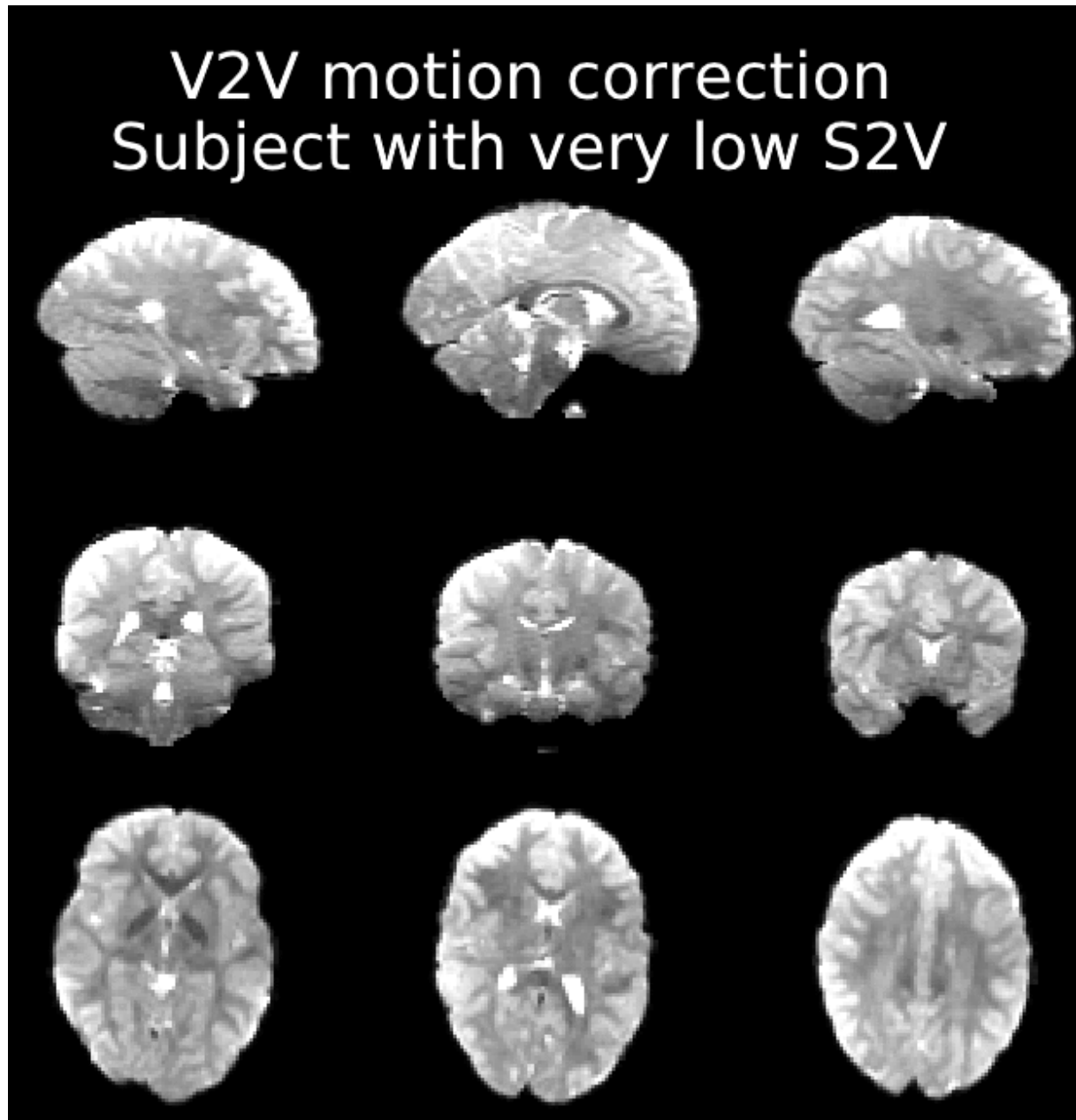


Figure C.1: 100th volume of subject 21TD after V2V movement correction.

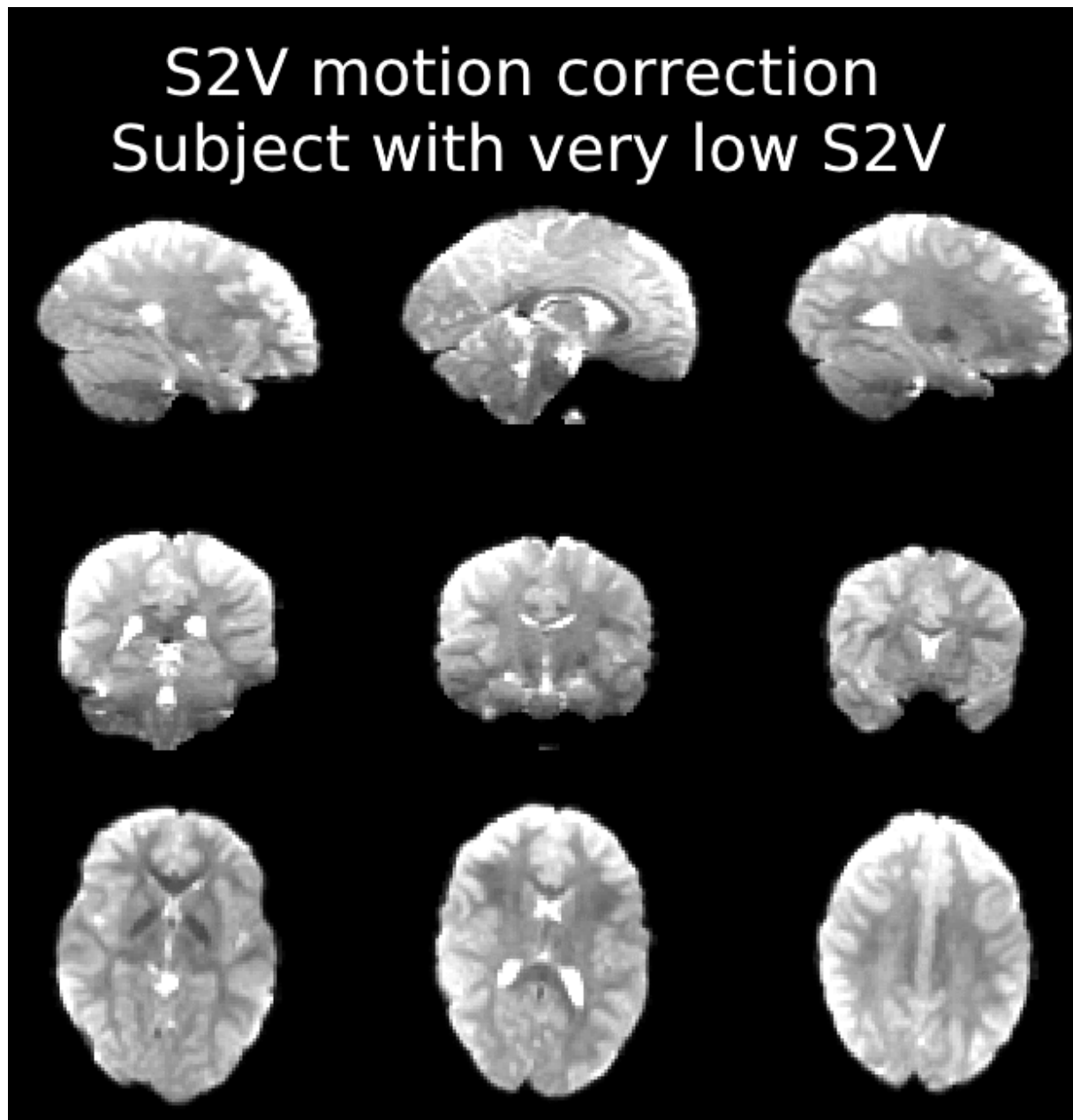


Figure C.2: 100th volume of subject 21TD after S2V movement correction.

C.2 Tractography comparison

C.2.0.1 Posterior corpus callosum

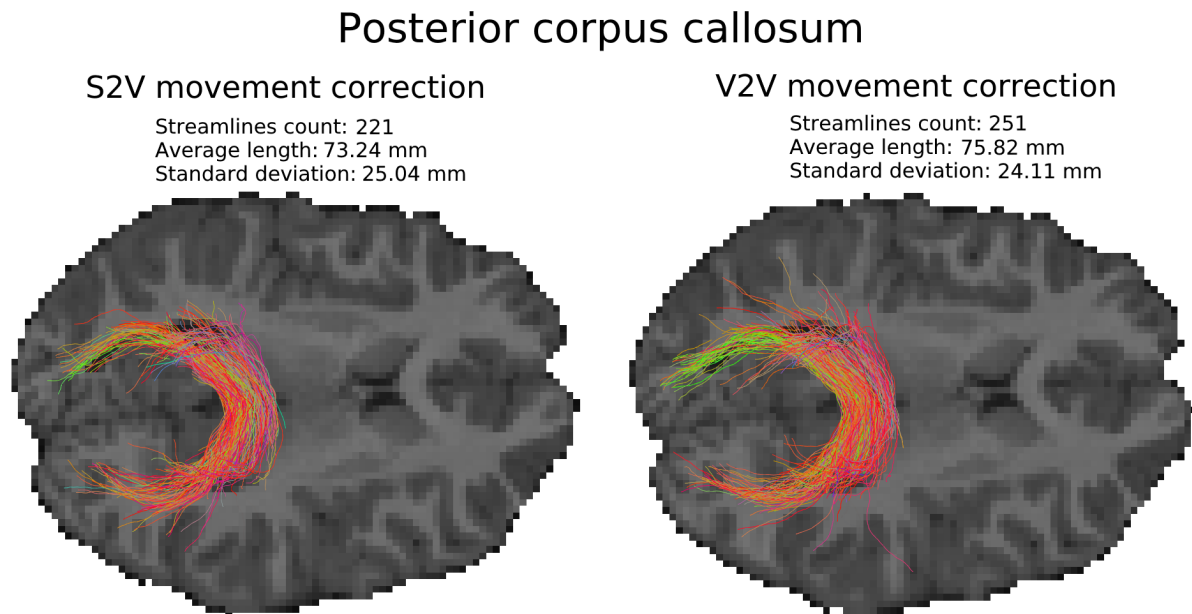


Figure C.3: Comparison of the effect of movement correction on **probabilistic** tracking of the posterior corpus callosum on the subject with **almost no** slice-to-volume movement artifact.

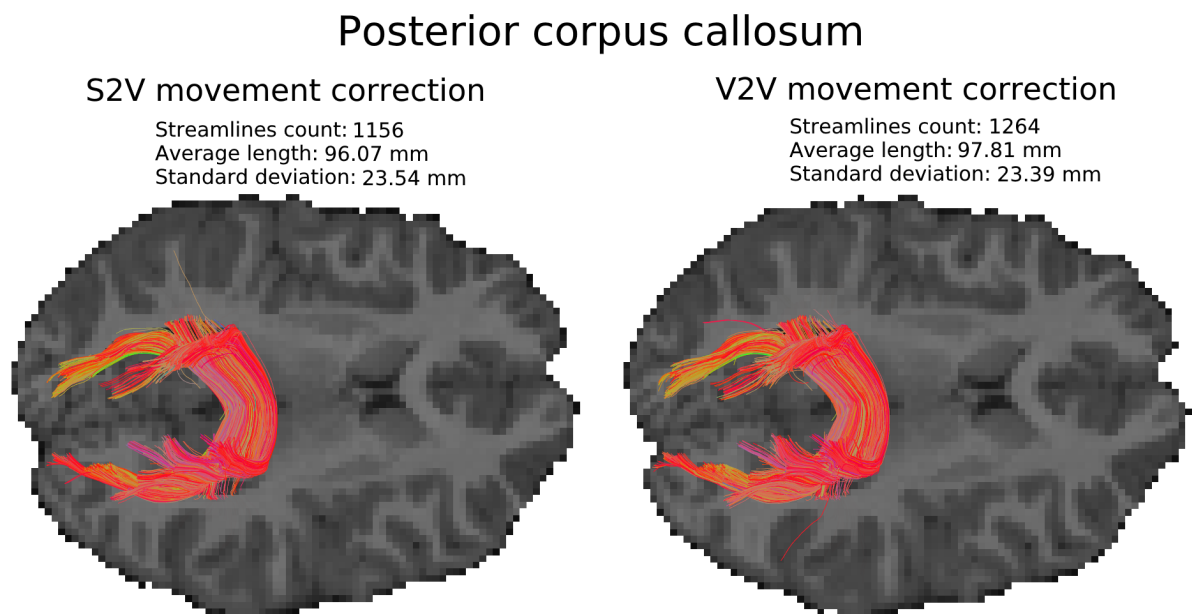


Figure C.4: Comparison of the effect of movement correction on **deterministic** tracking of the posterior corpus callosum on the subject with **almost no** slice-to-volume movement artifact.

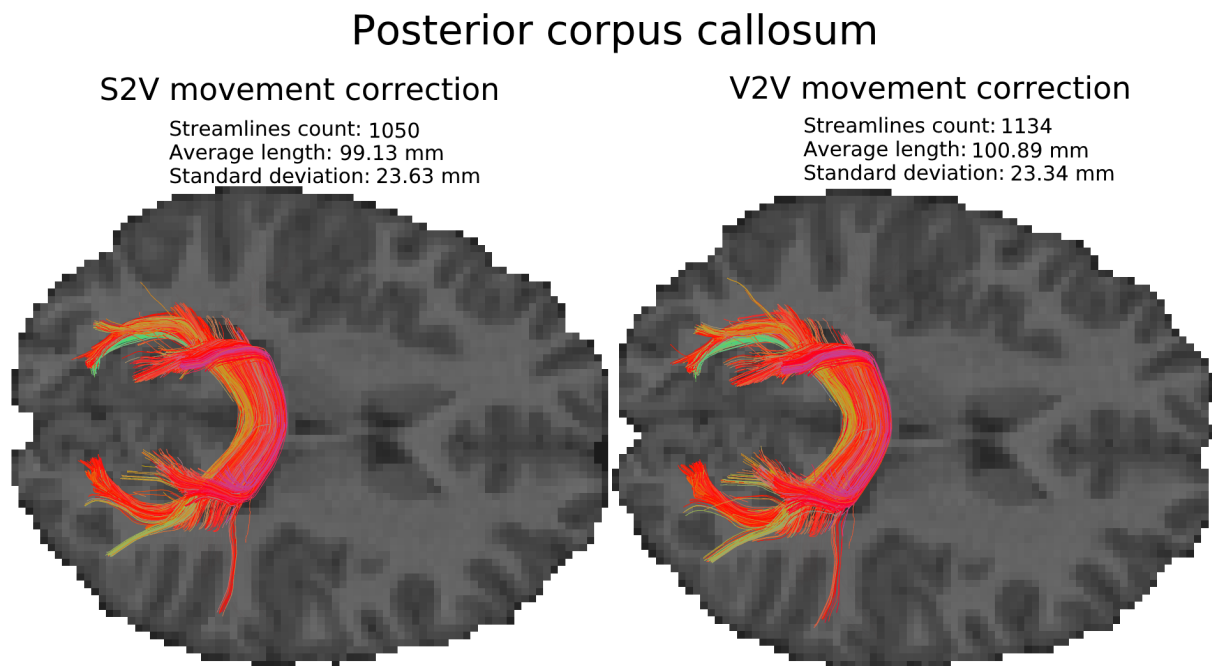


Figure C.5: Comparison of the effect of movement correction on **deterministic** tracking of the posterior corpus callosum on the subject with **mild** slice-to-volume movement artifact.

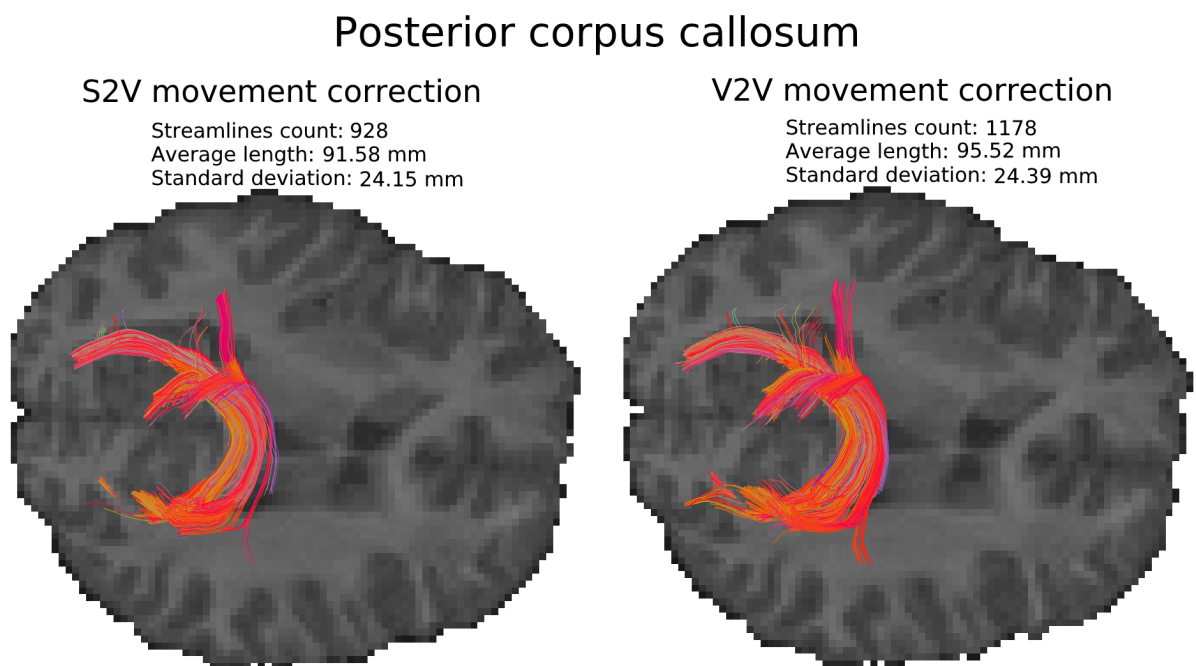


Figure C.6: Comparison of the effect of movement correction on **deterministic** tracking of the posterior corpus callosum on the subject with **severe** slice-to-volume movement artifact.

C.2.0.2 Right superior longitudinal fasciculus

The white matter query language was unable to find tracks when using **deterministic** tracking of the right superior longitudinal fasciculus on the subject with **almost no** slice-to-volume movement artifact.

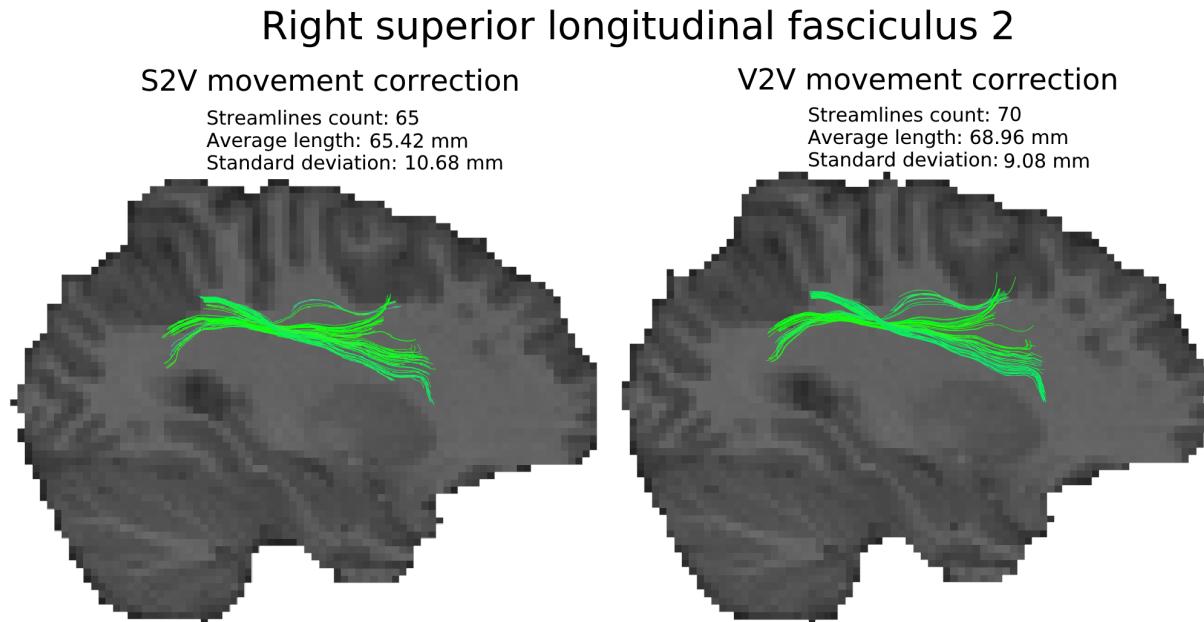


Figure C.7: Comparison of the effect of movement correction on **deterministic** tracking of the right superior longitudinal fasciculus on the subject with **mild** slice-to-volume movement artifact.

The white matter query language was unable to find tracks when using **deterministic** tracking of the right superior longitudinal fasciculus on the subject with **severe** slice-to-volume movement artifact.

C.2.0.3 Right corticospinal tract

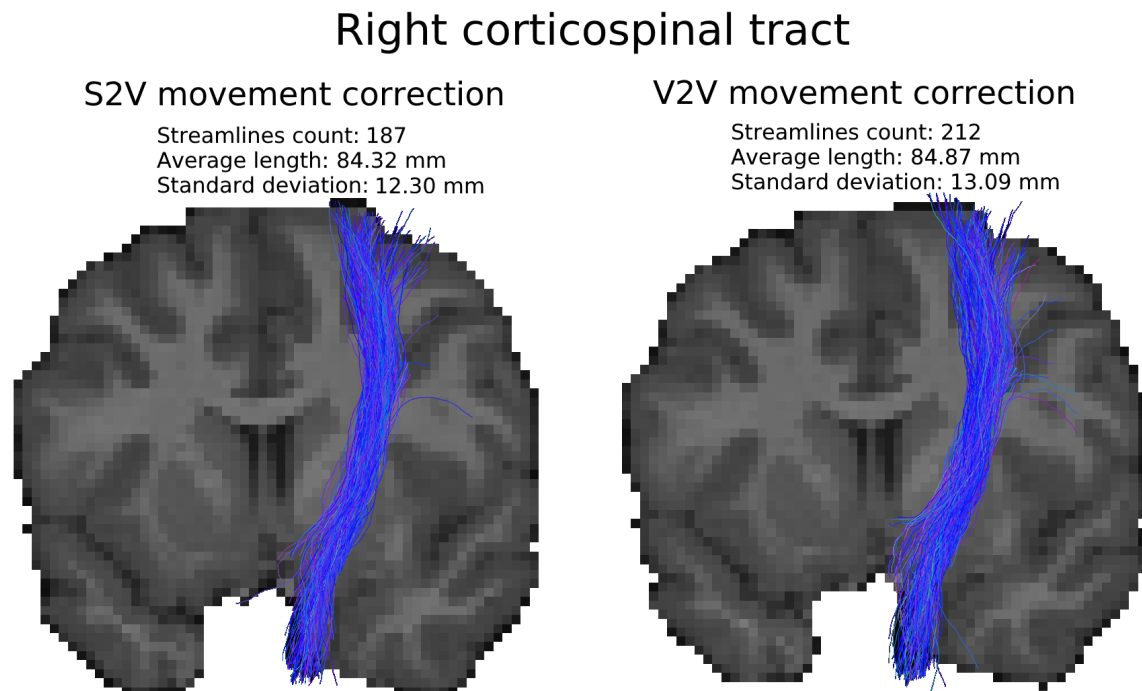


Figure C.8: Comparison of the effect of movement correction on **probabilistic** tracking of the right corticospinal tract on the subject with **almost no** slice-to-volume movement artifact.

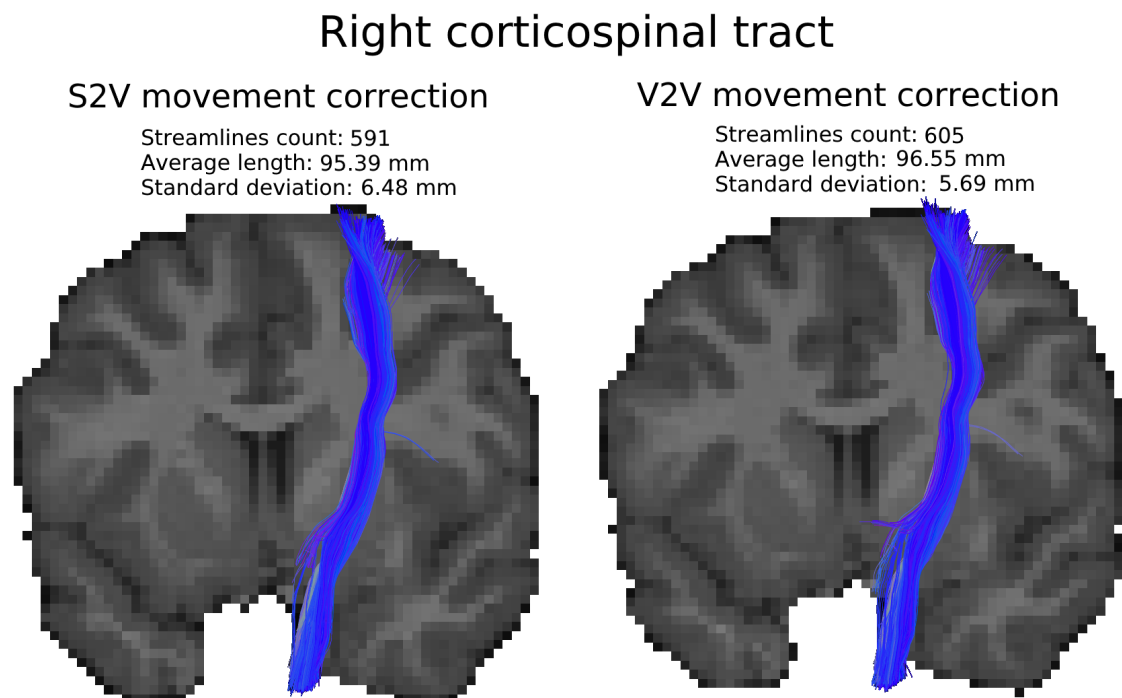


Figure C.9: Comparison of the effect of movement correction on **deterministic** tracking of the right corticospinal tract on the subject with **almost no** slice-to-volume movement artifact.

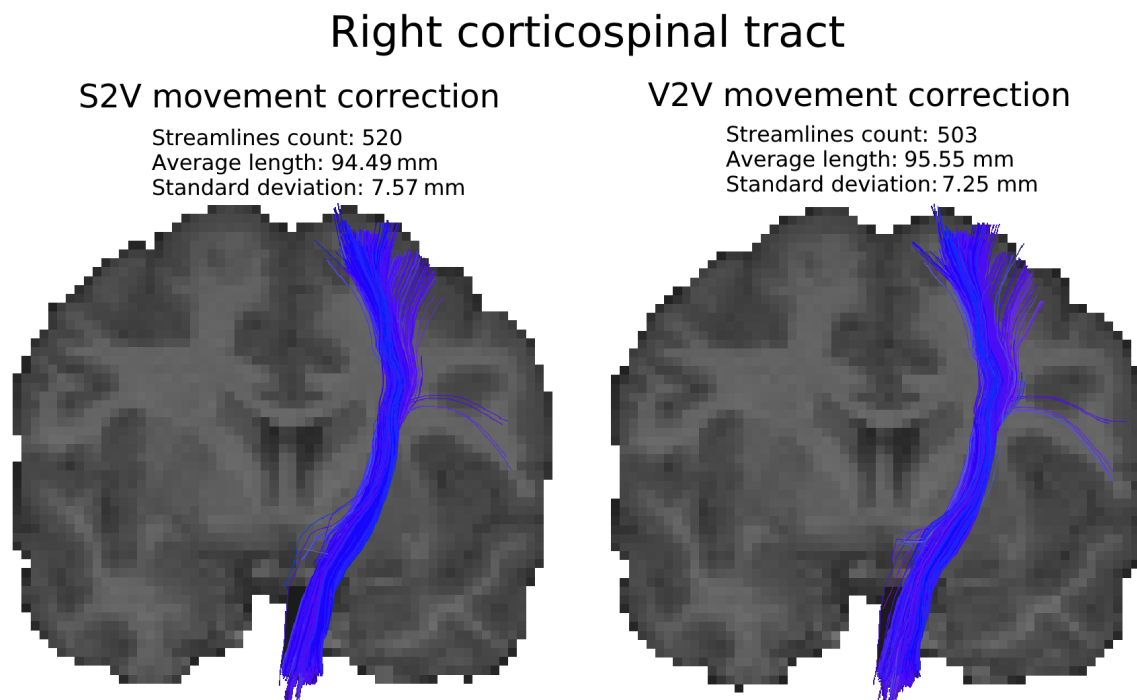


Figure C.10: Comparison of the effect of movement correction on **deterministic** tracking of the right corticospinal tract on the subject with **mild** slice-to-volume movement artifact.

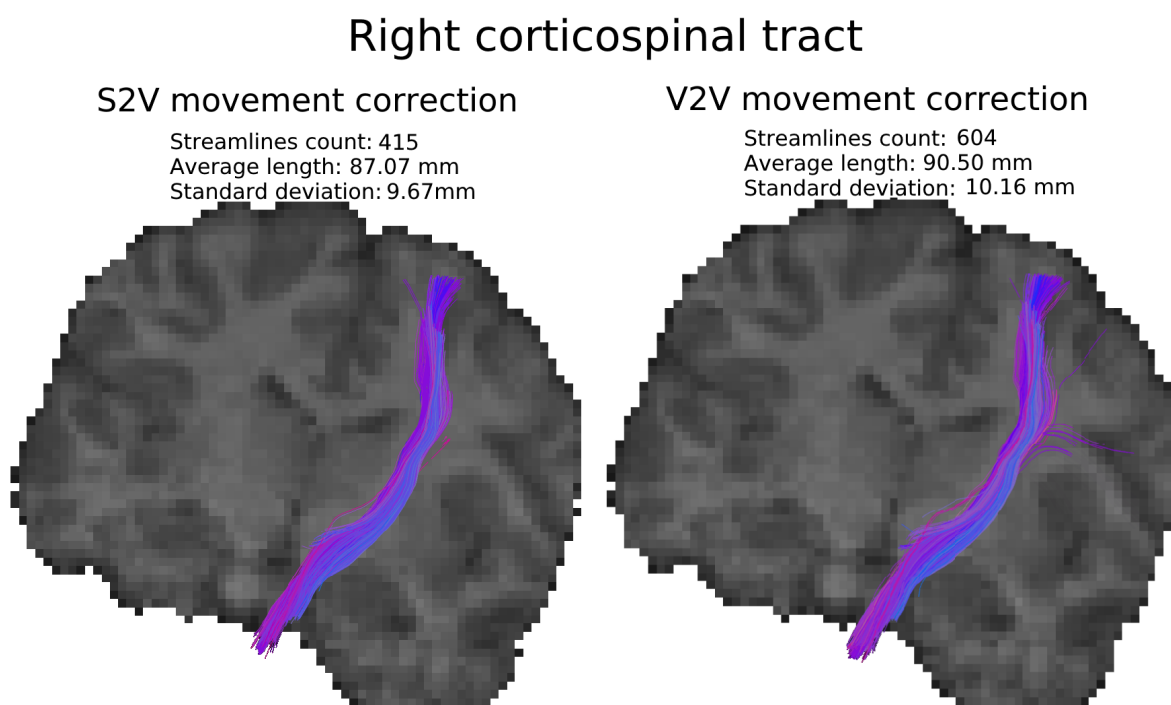


Figure C.11: Comparison of the effect of movement correction on **deterministic** tracking of the right corticospinal tract on the subject with **severe** slice-to-volume movement artifact.

C.3 Tracts definition

```

1 CST.side = (endpoints_in(brainstem)
2             and (endpoints_in(precentral.side)
3                 or endpoints_in(postcentral.side))
4 )
5
6 SLF2.side = (endpoints_in(inferiorparietal.side)
7             and endpoints_in(caudalmiddlefrontal.side
8                             or rostralmiddlefrontal.side
9                             or parsopercularis.side
10                            or parstriangularis.side)
11             not in posterior_of(supramarginal.side)
12             not in anterior_of(caudalmiddlefrontal.side)
13             not in medial_of(inferiorparietal.side)
14             and only(frontal.side
15                     or parietal.side
16                     or centrum_semiovale.side))
17
18 CC_bundle_post = (endpoints_in(parietal.left
19                             or occipital.left
20                             or temporal.left
21                             or cingulate.left)
22                 and endpoints_in(parietal.right
23                                 or occipital.right
24                                 or temporal.right or
25                                 cingulate.right)
26
27                 and CC_Posterior
28                 not in brainstem
29                 not in (Thalamus_Proper.left
30                         or Thalamus_Proper.right)
31                 not in (Putamen.left
32                         or Putamen.right)
33                 not in (pallidum.left
34                         or pallidum.right)
35                 not in (caudate.left
36                         or caudate.right)
37                 not in (choroid_plexus.left
38                         or choroid_plexus.right)
39                 not in (VentralDC.left
40                         or VentralDC.right)
41                 not in anterior_of(CC_Anterior))

```

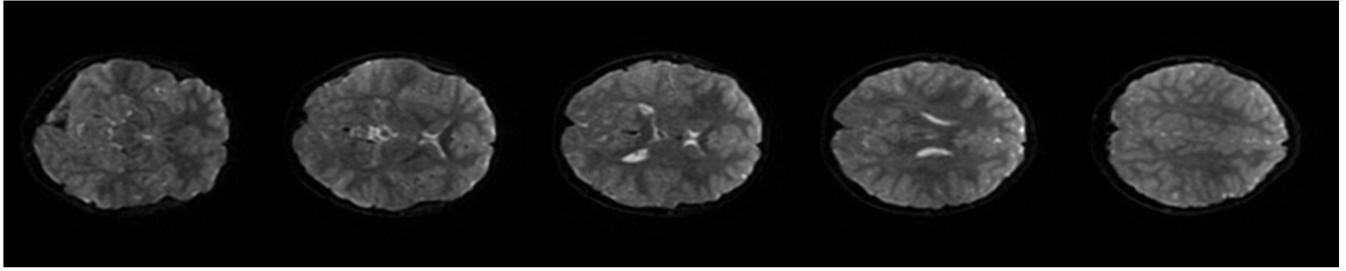
Appendix D

Quality control report of the subject
21TD.

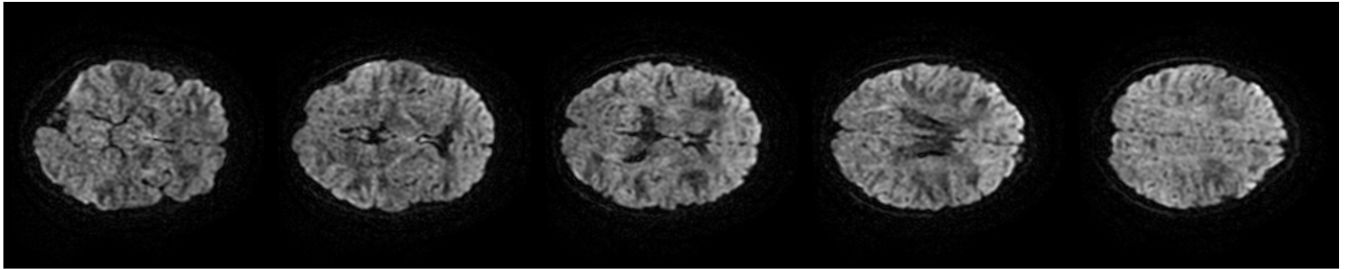
Elikopy : Quality control report - Preprocessing

	Input parameters
patient id	21TD
reslice	True
denoising	True
gibbs	True
topup	True
eddy	True
bet_median_radius	2
bet_numpass	1
bet_dilate	2
cuda	True
s2v	[22, 10, 6, 'trilinear']
olrep	[True, 6, 250, 'both']

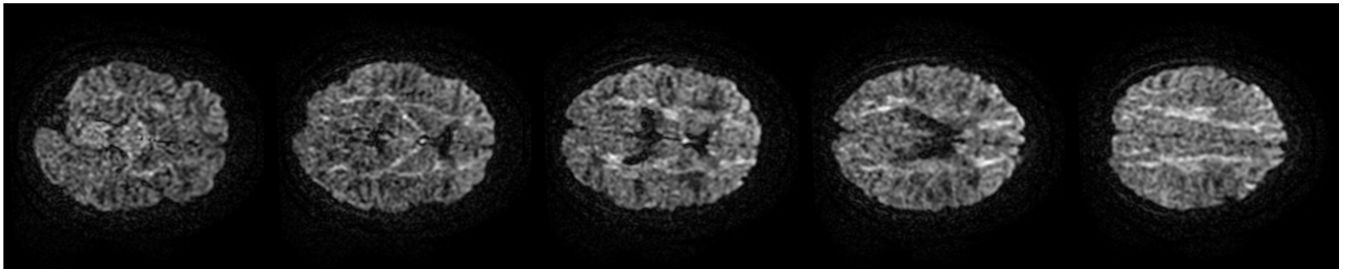
Raw data at b=0.0



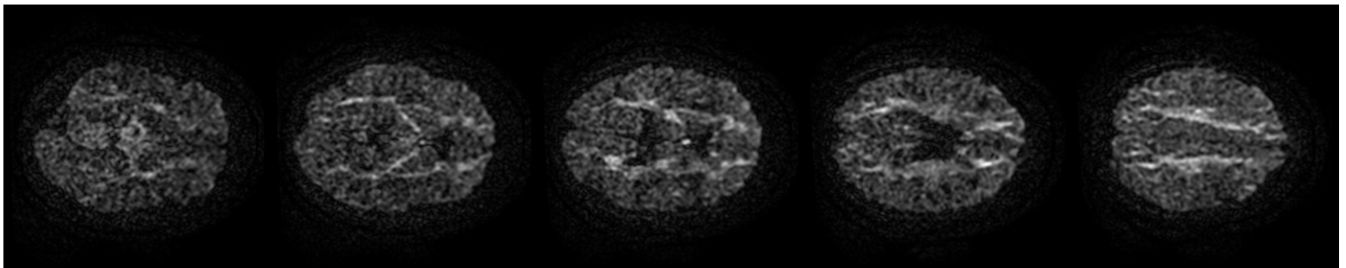
Raw data at b=1000.0



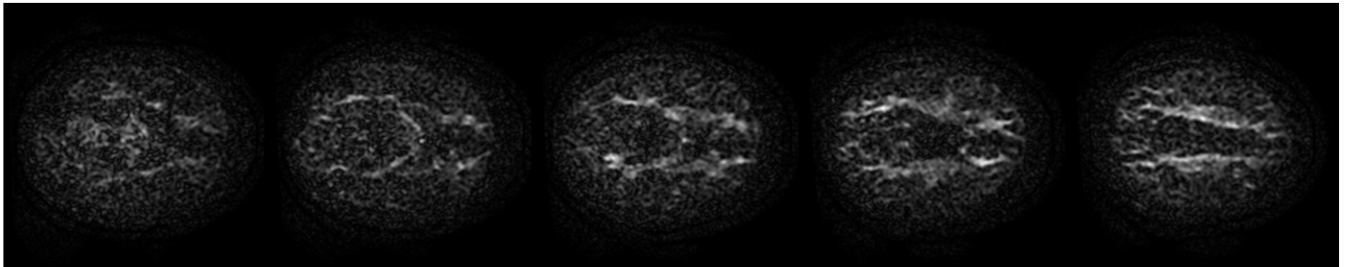
Raw data at b=2000.0



Raw data at b=3000.0

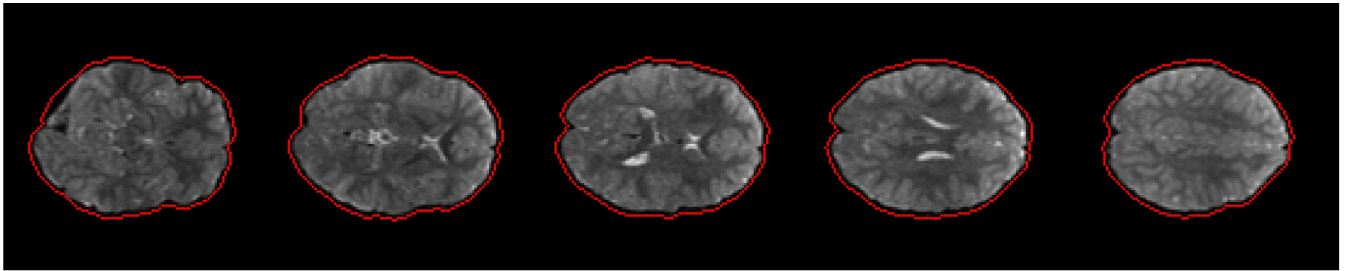


Raw data at b=5000.0

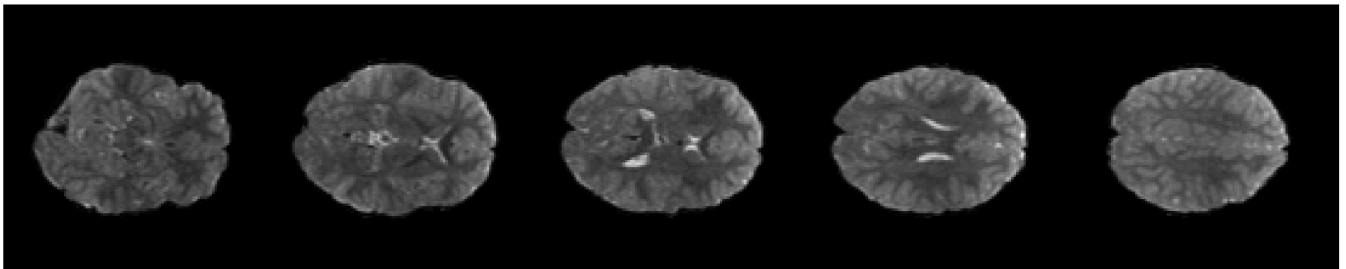


Overview of processing steps for $b=0.0$

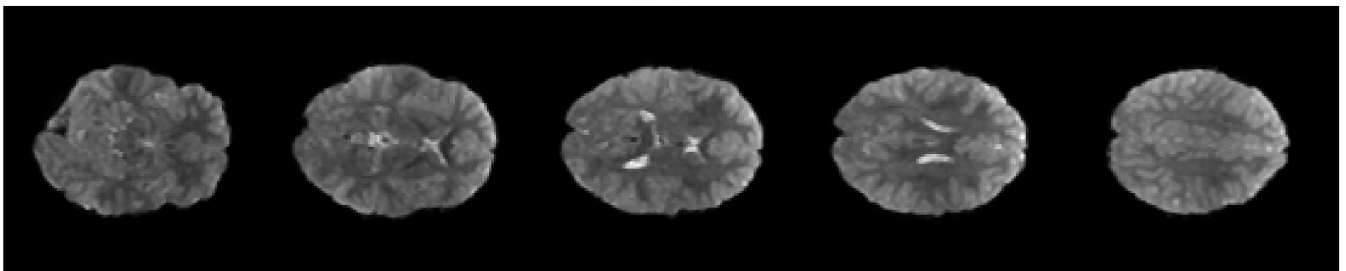
brain extraction



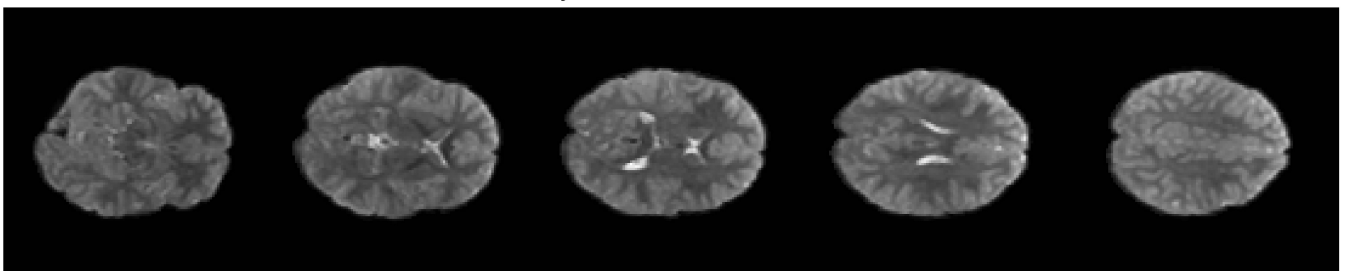
Denoising



Gibbs ringing correction

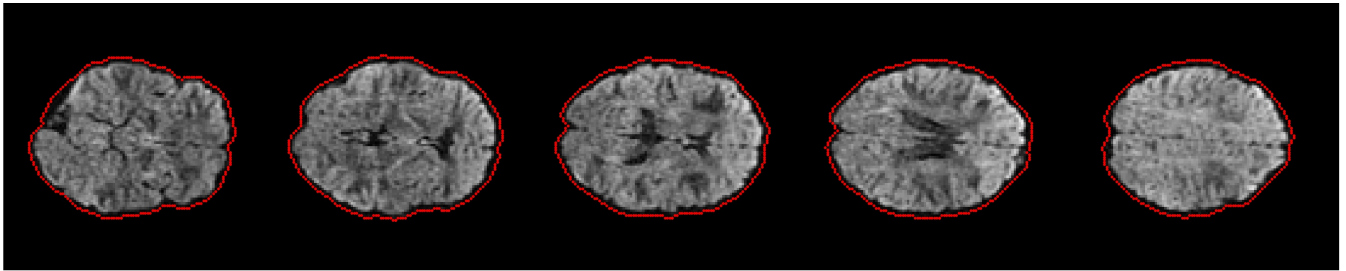


Eddy and motion correction

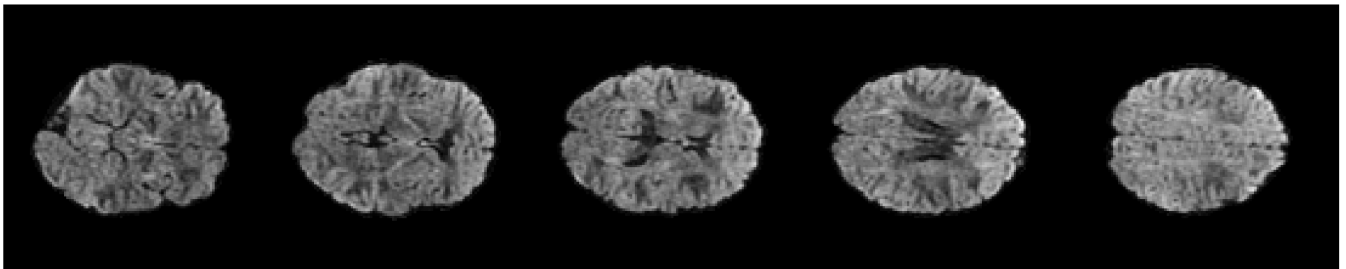


Overview of processing steps for b=1000.0

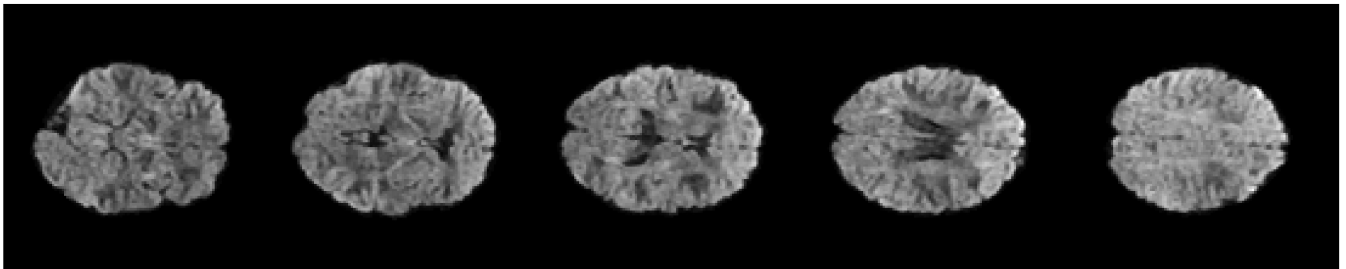
brain extraction



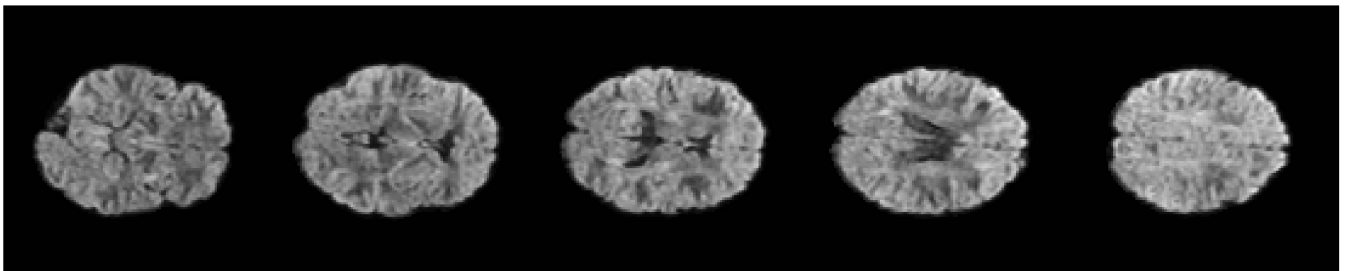
Denoising



Gibbs ringing correction

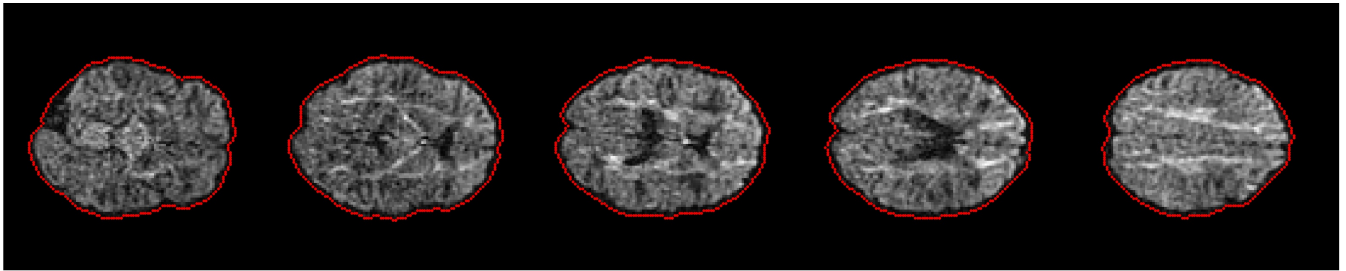


Eddy and motion correction

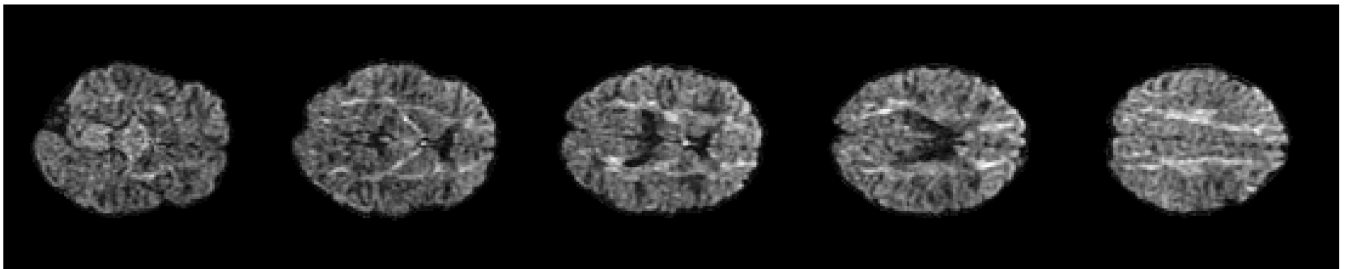


Overview of processing steps for b=2000.0

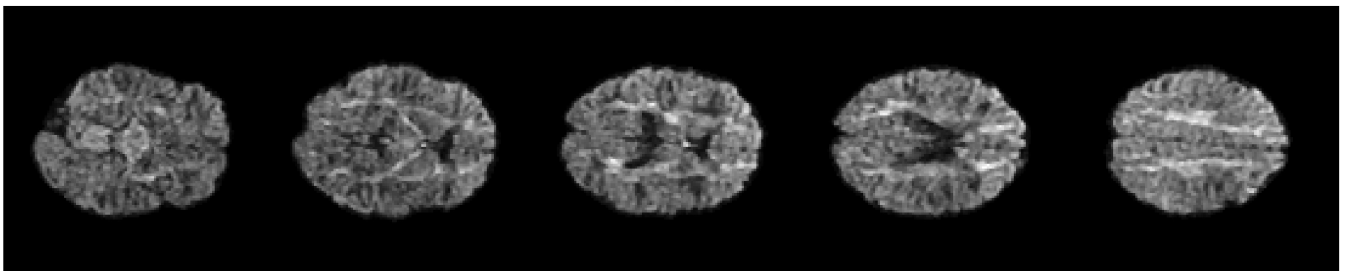
brain extraction



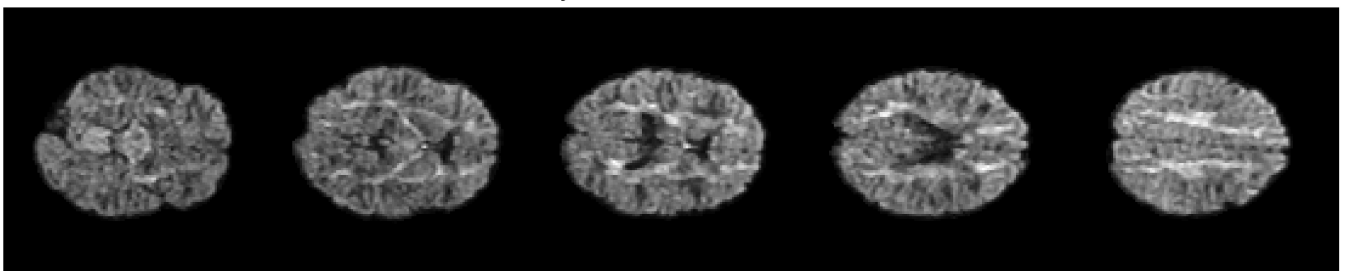
Denoising



Gibbs ringing correction

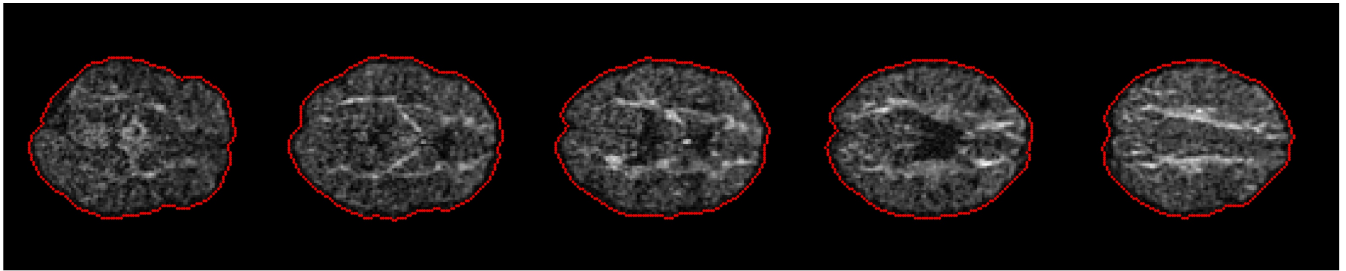


Eddy and motion correction

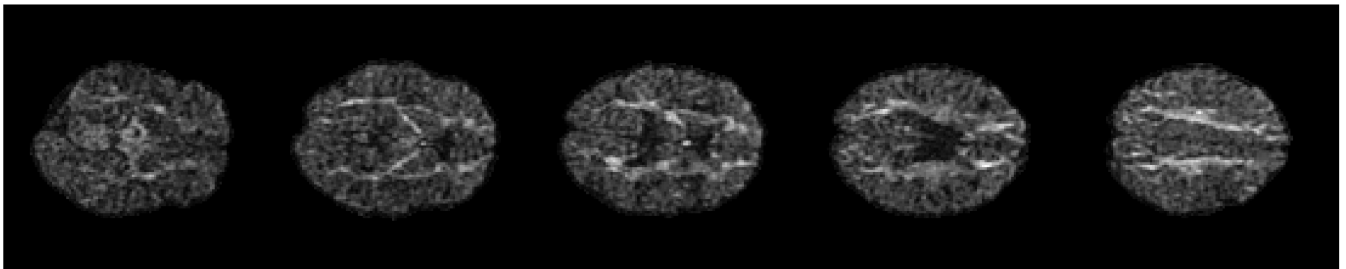


Overview of processing steps for b=3000.0

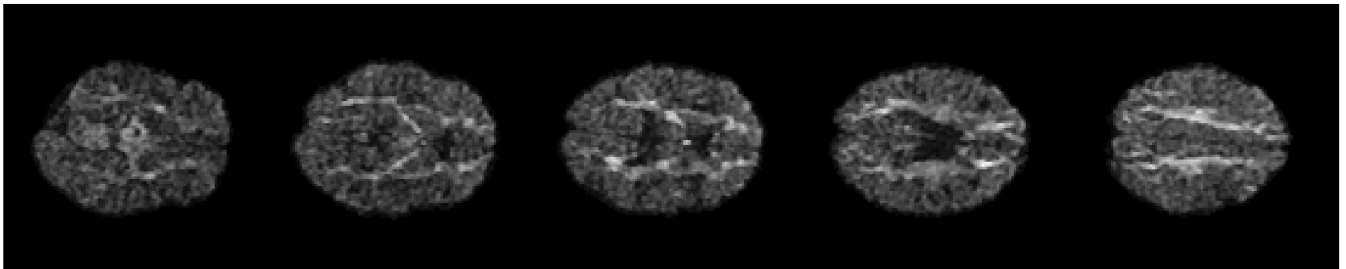
brain extraction



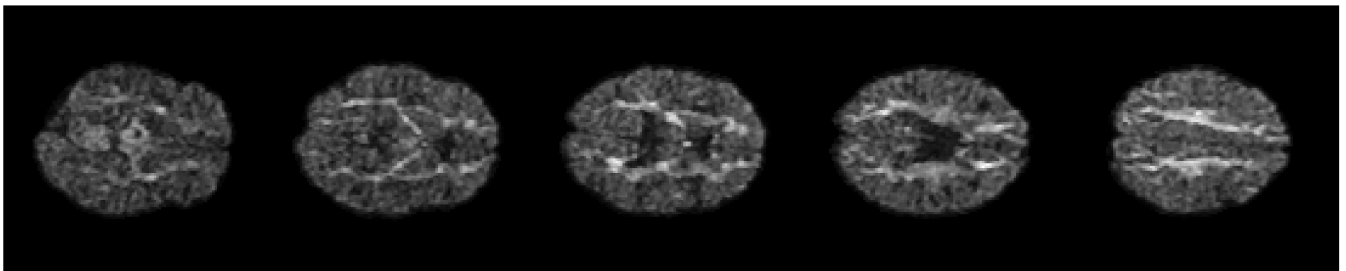
Denoising



Gibbs ringing correction

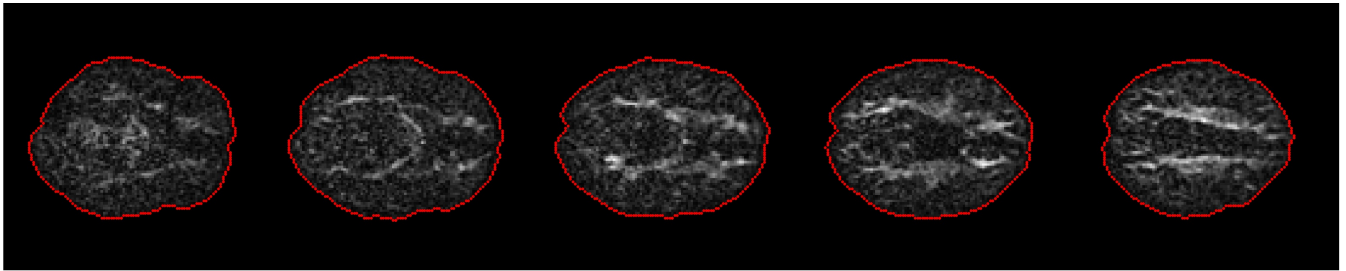


Eddy and motion correction

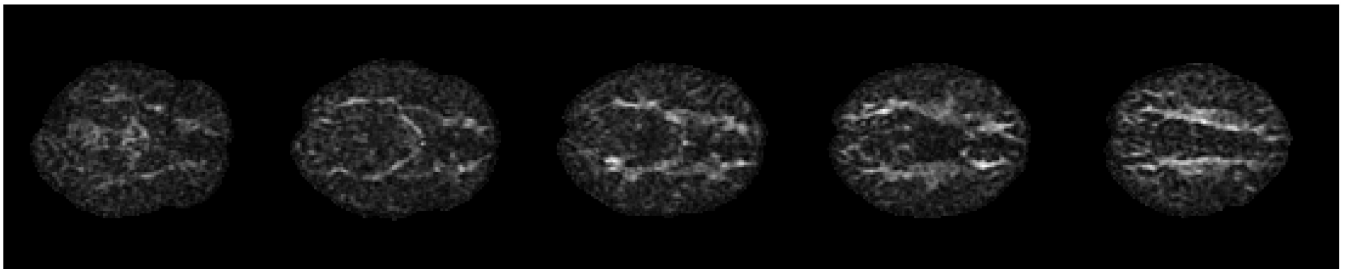


Overview of processing steps for b=5000.0

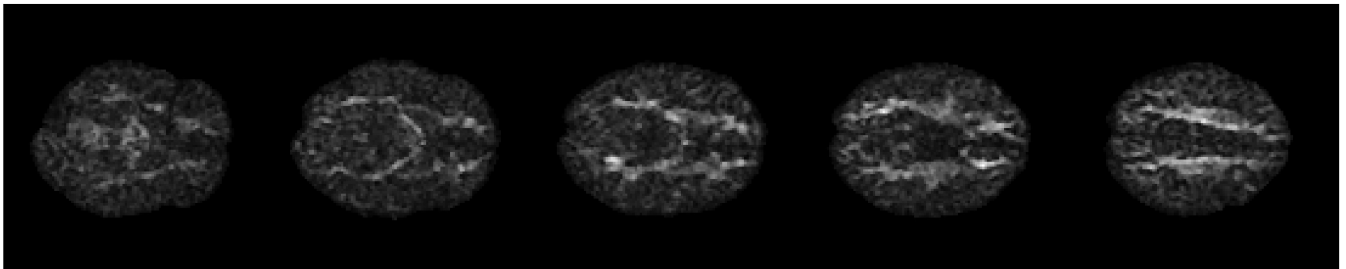
brain extraction



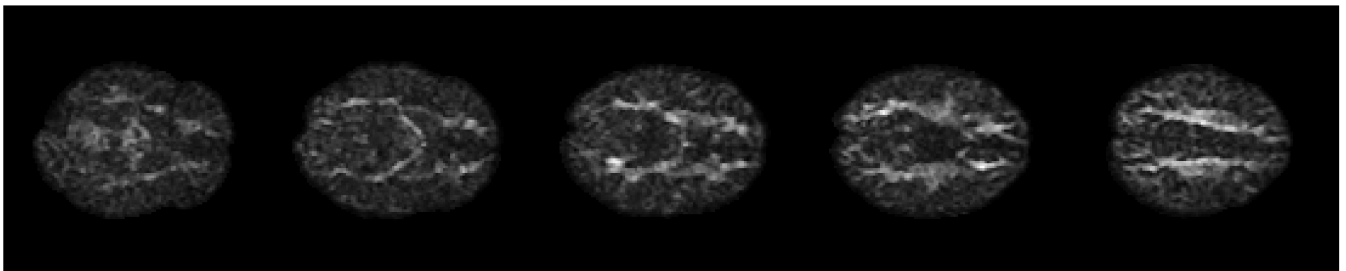
Denoising



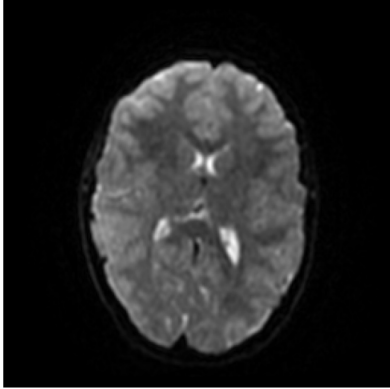
Gibbs ringing correction



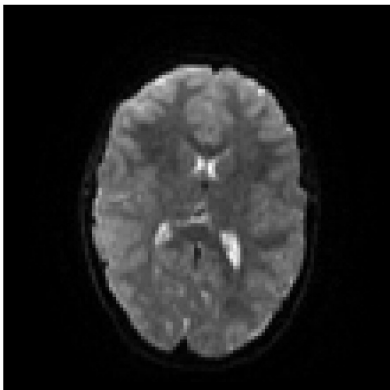
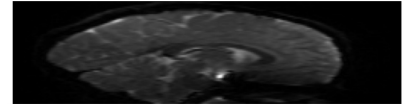
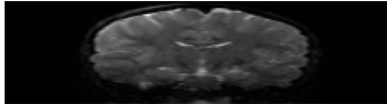
Eddy and motion correction



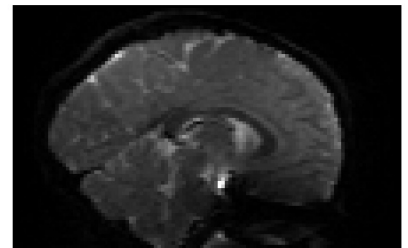
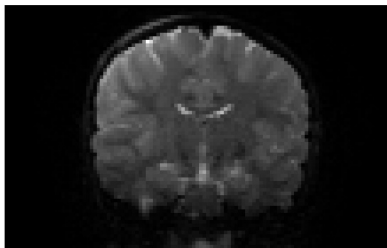
Data reslice



Raw data

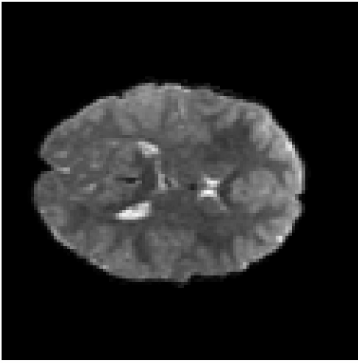


Reslice data

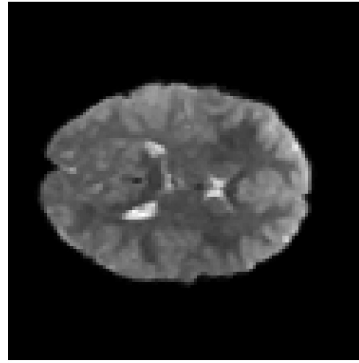


Quality control report - Preprocessing

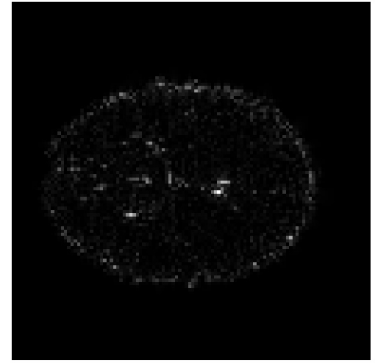
Gibbs uncorrected at b=0.0



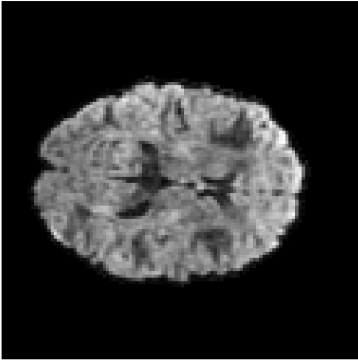
Gibbs corrected at b=0.0



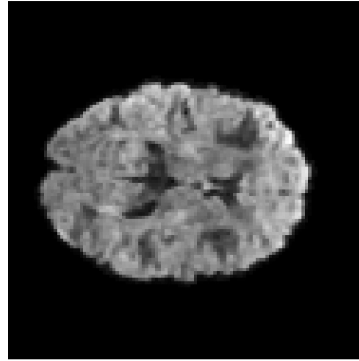
Residual at b=0.0



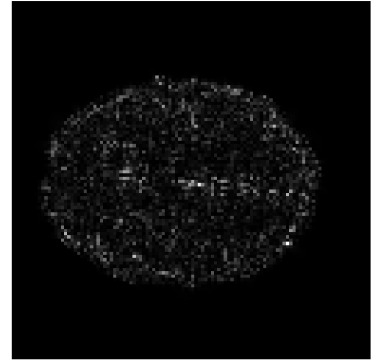
Gibbs uncorrected at b=1000.0



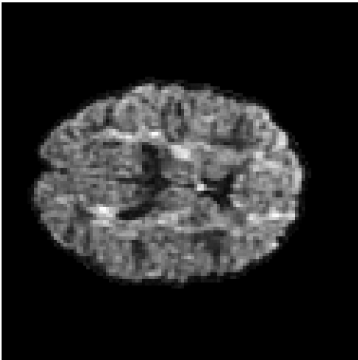
Gibbs corrected at b=1000.0



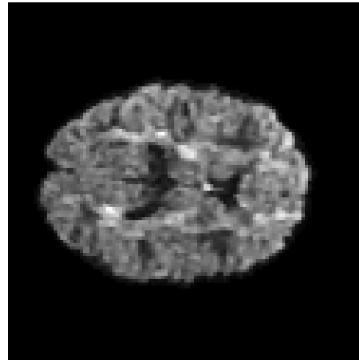
Residual at b=1000.0



Gibbs uncorrected at b=2000.0



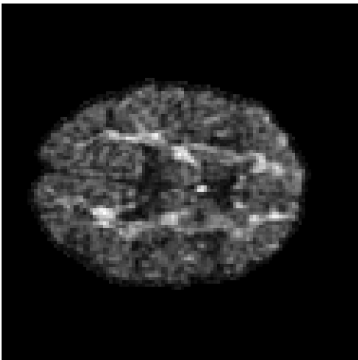
Gibbs corrected at b=2000.0



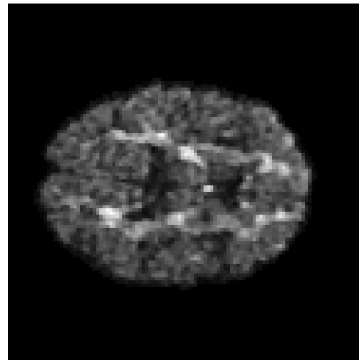
Residual at b=2000.0



Gibbs uncorrected at b=3000.0



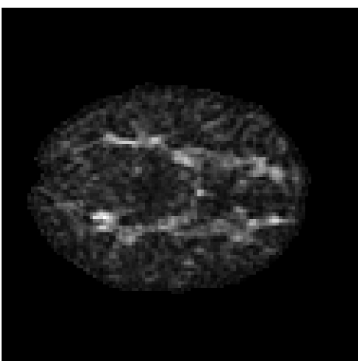
Gibbs corrected at b=3000.0



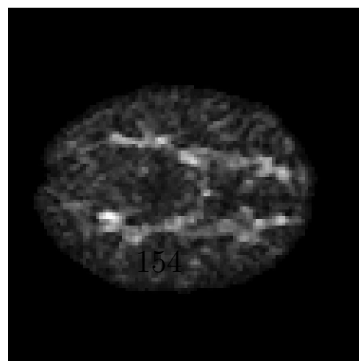
Residual at b=3000.0



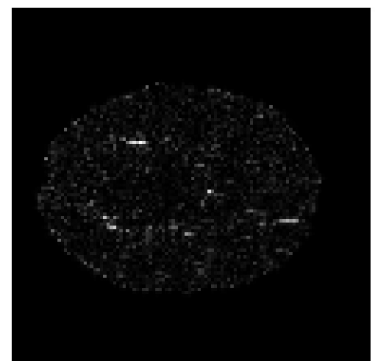
Gibbs uncorrected at b=5000.0



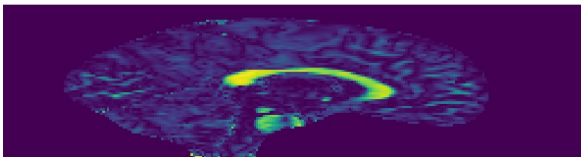
Gibbs corrected at b=5000.0



Residual at b=5000.0



Corpus callosum (CC)

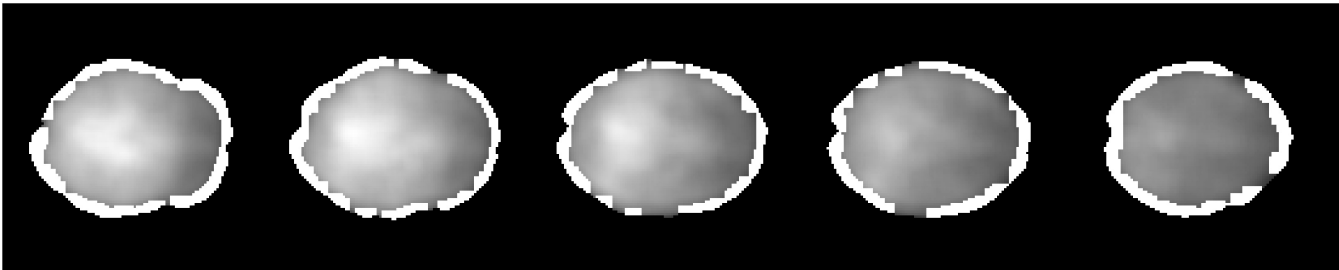


CC mask used for SNR computation



SNR of the b0 image	97
Estimated SNR range	24 - 53

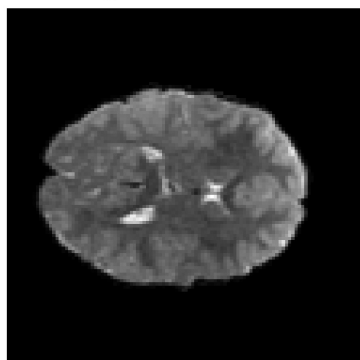
PCA Noise standard deviation estimation



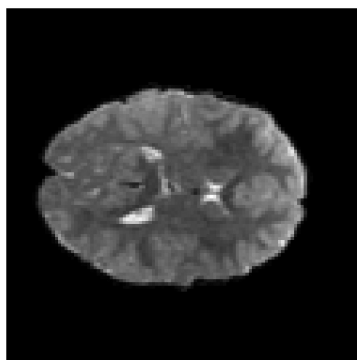
MPPCA SNR estimation	79.57409308400413
----------------------	-------------------

Quality control report - Preprocessing

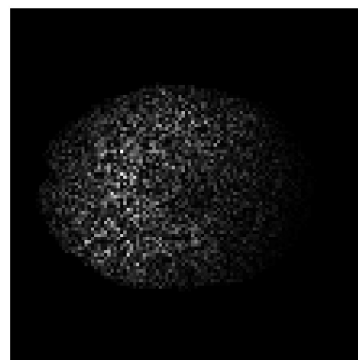
Original at $b=0.0$



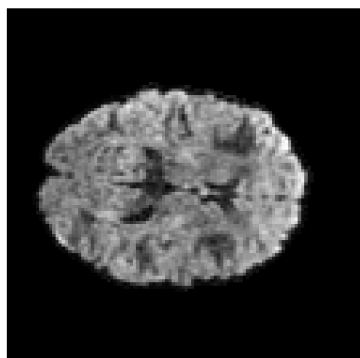
MPPCA denoised at $b=0.0$



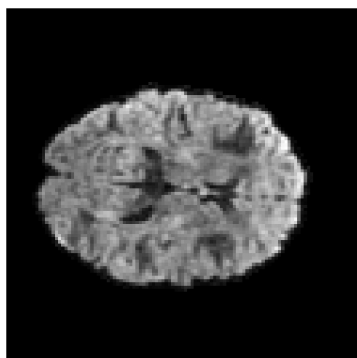
Residual at $b=0.0$



Original at $b=1000.0$



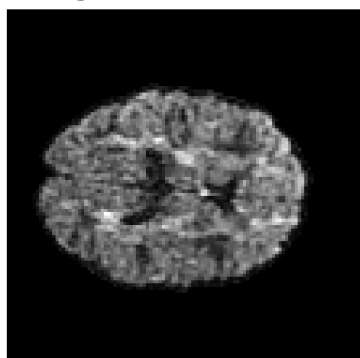
MPPCA denoised at $b=1000.0$



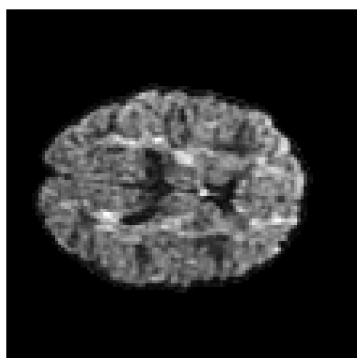
Residual at $b=1000.0$



Original at $b=2000.0$



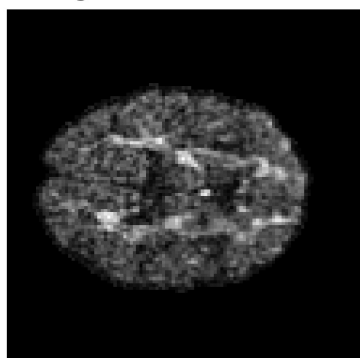
MPPCA denoised at $b=2000.0$



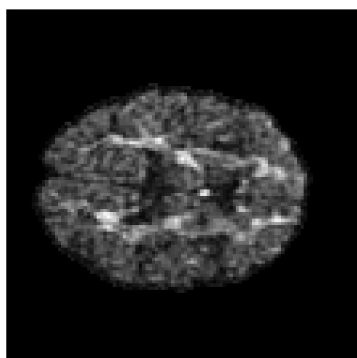
Residual at $b=2000.0$



Original at $b=3000.0$



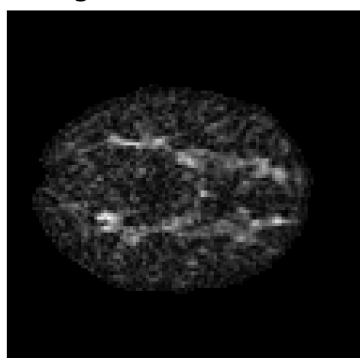
MPPCA denoised at $b=3000.0$



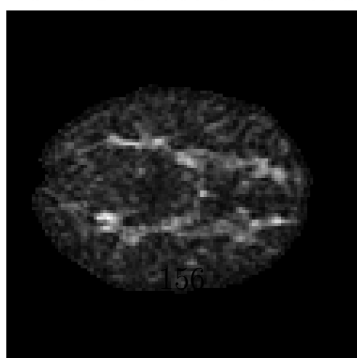
Residual at $b=3000.0$



Original at $b=5000.0$



MPPCA denoised at $b=5000.0$

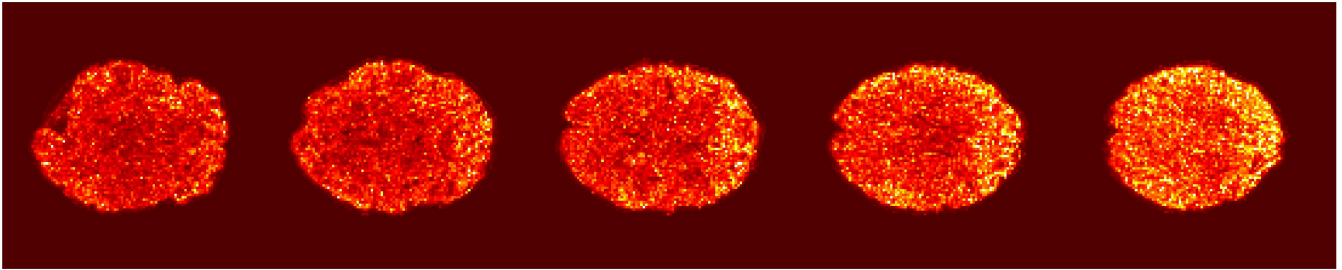


Residual at $b=5000.0$

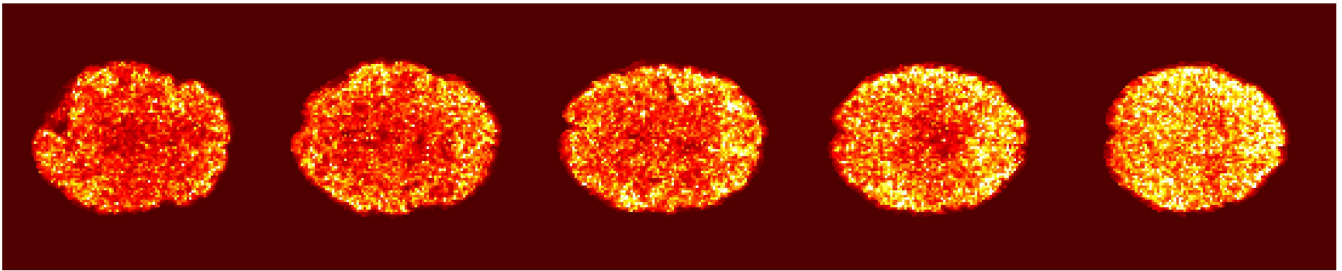


tSNR for shell b=0.0

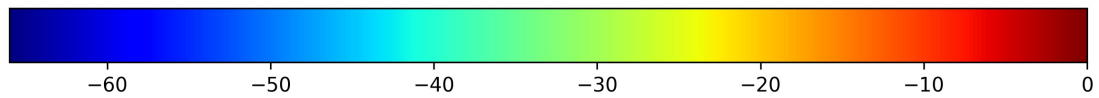
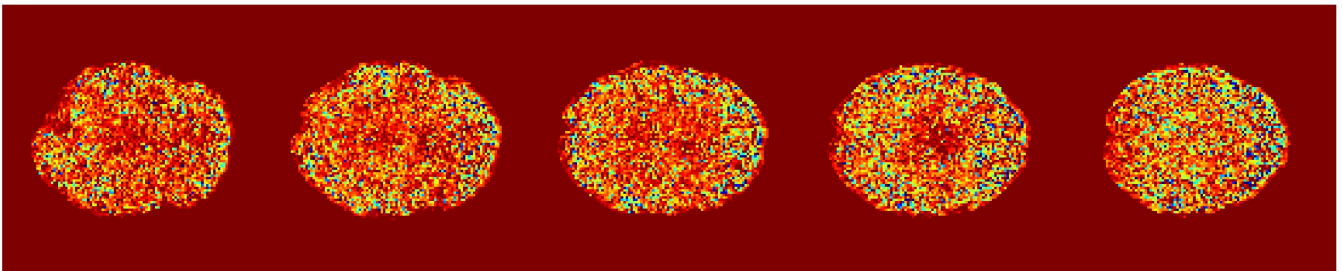
Raw data tSNR



Processed data tSNR

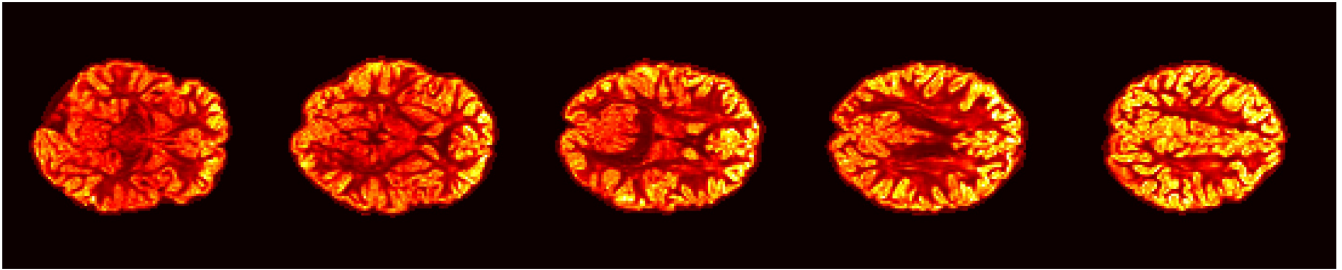


difference: raw_tSNR - processed_tSNR

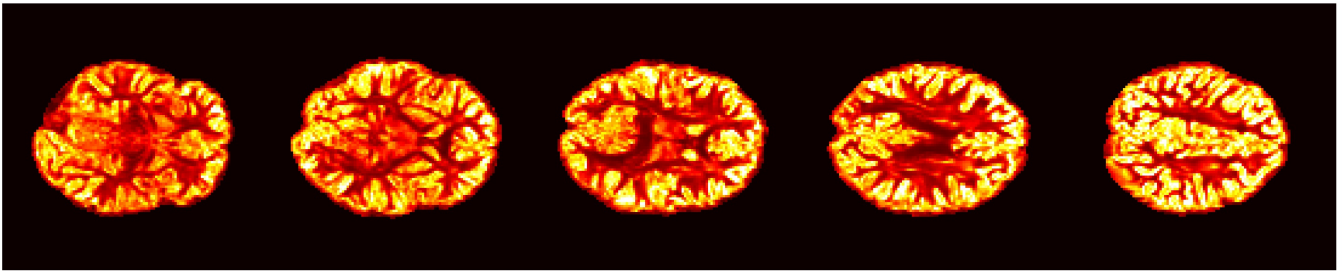


tSNR for shell b=1000.0

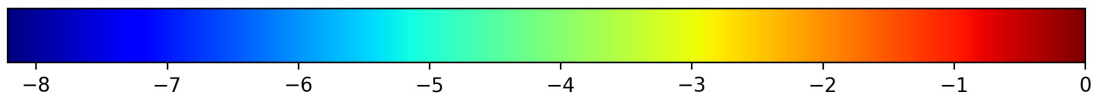
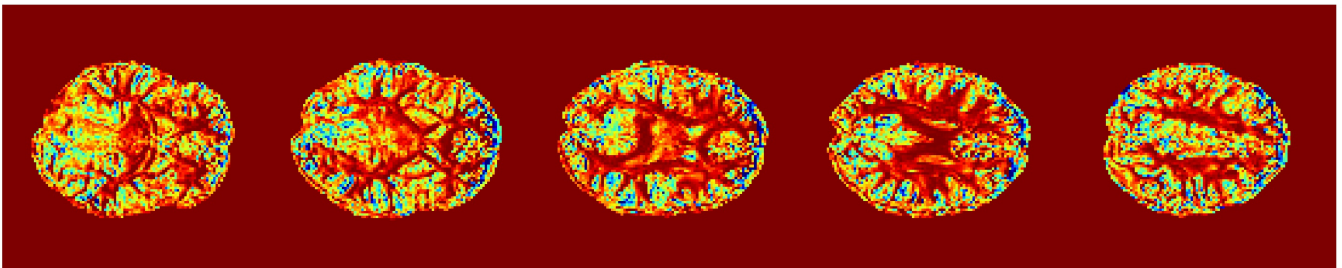
Raw data tSNR



Processed data tSNR

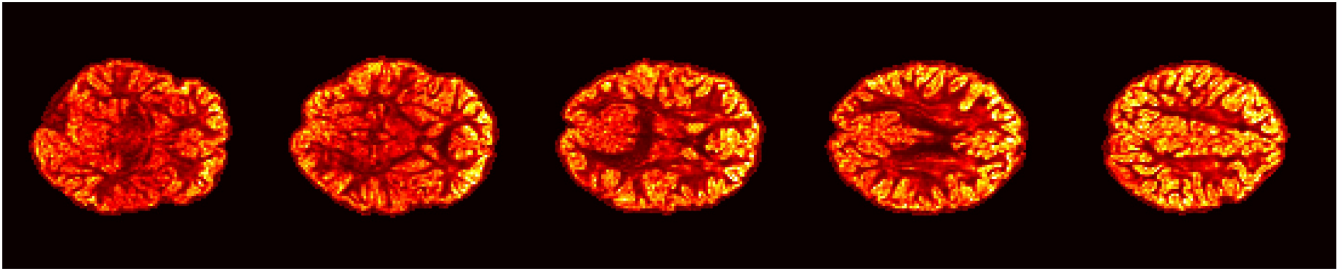


difference: raw_tSNR - processed_tSNR

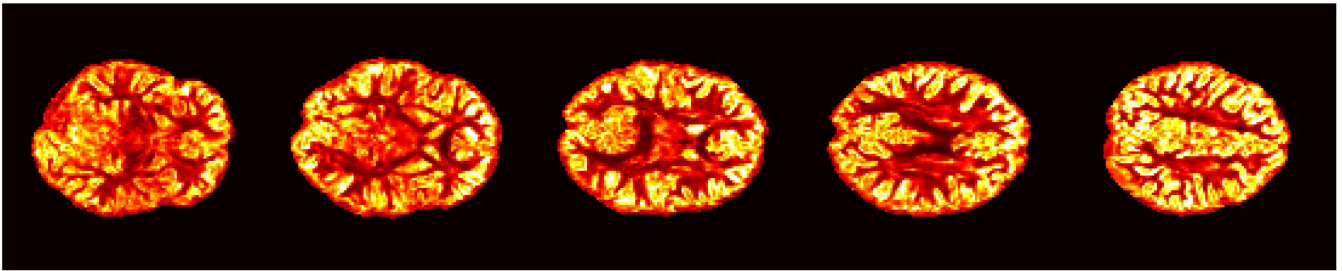


tSNR for shell b=2000.0

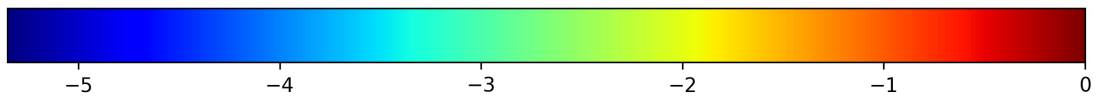
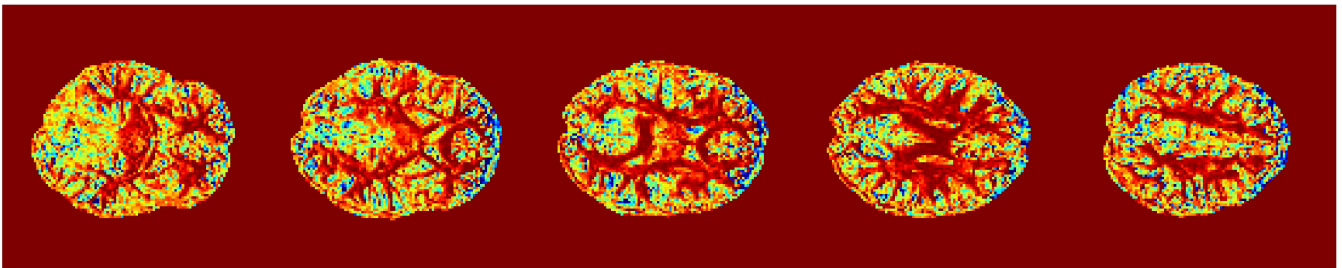
Raw data tSNR



Processed data tSNR

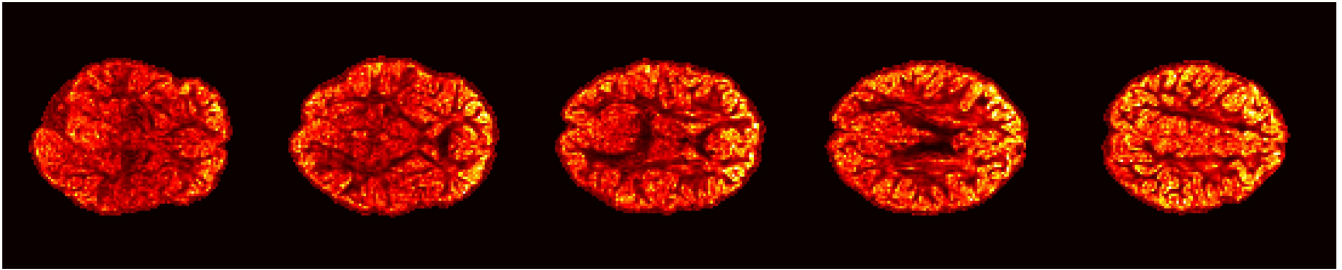


difference: raw_tSNR - processed_tSNR

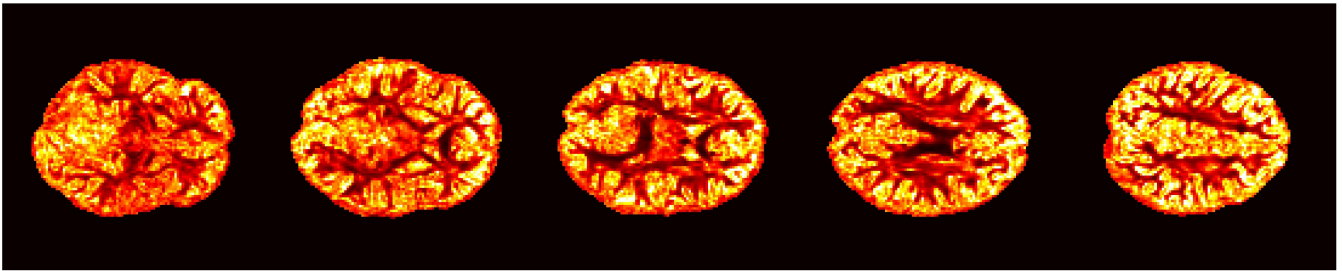


tSNR for shell b=3000.0

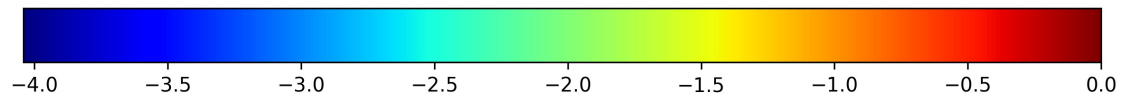
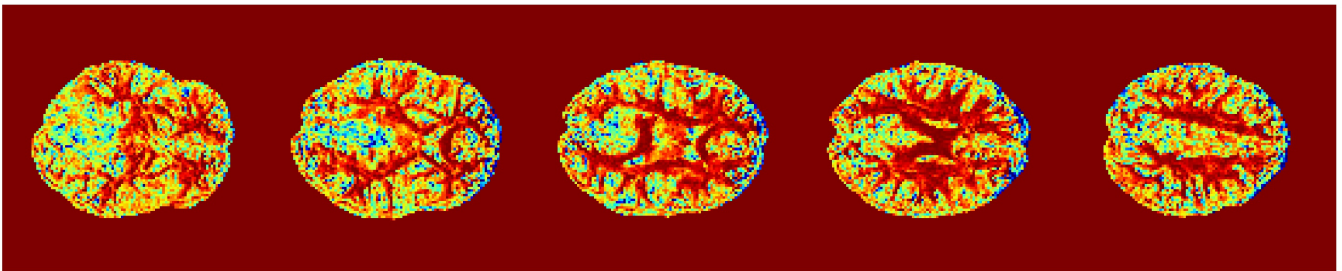
Raw data tSNR



Processed data tSNR

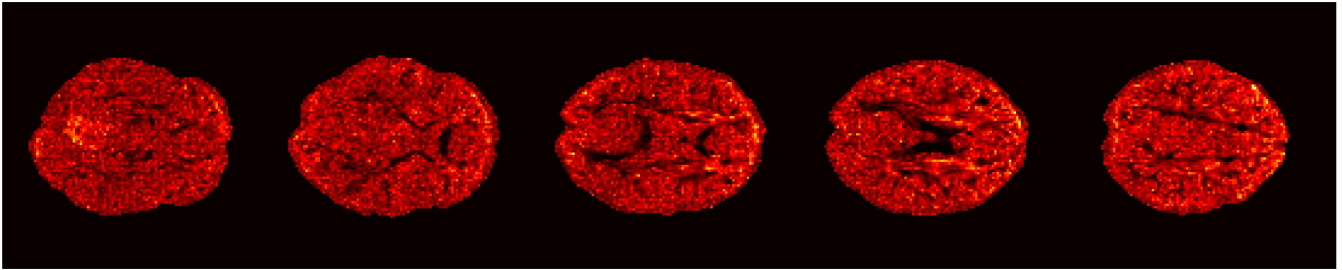


difference: raw_tSNR - processed_tSNR

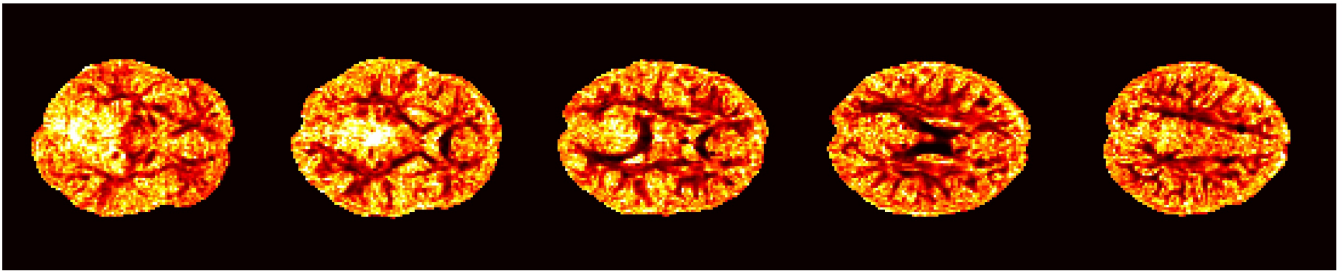


tSNR for shell b=5000.0

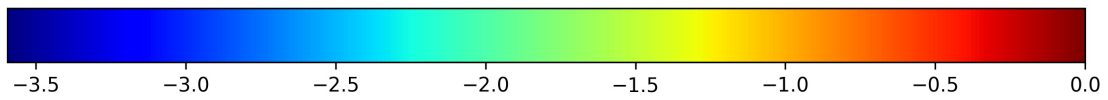
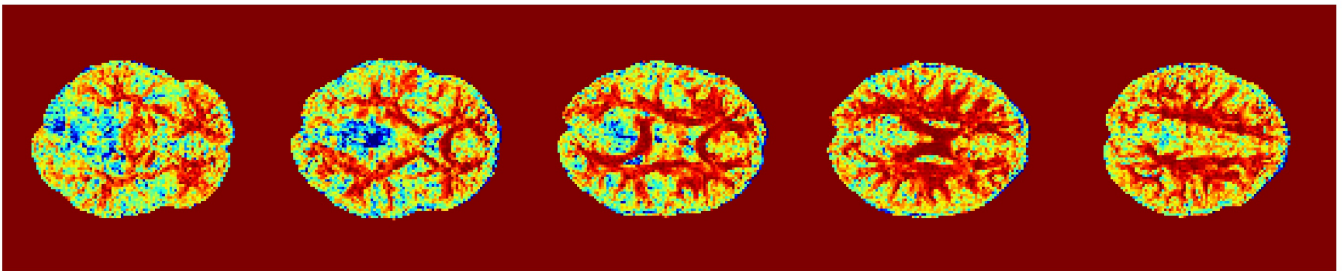
Raw data tSNR



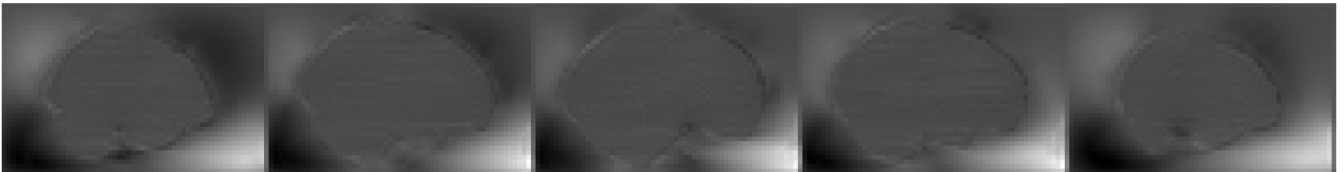
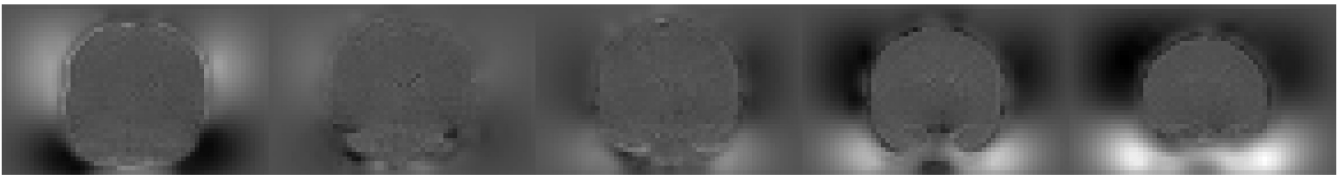
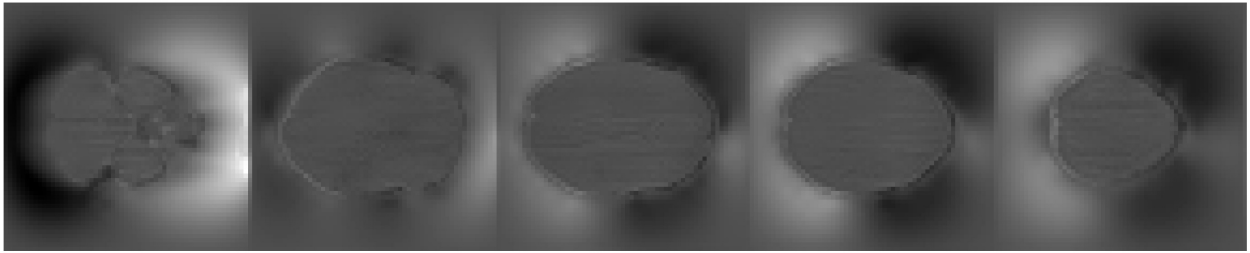
Processed data tSNR

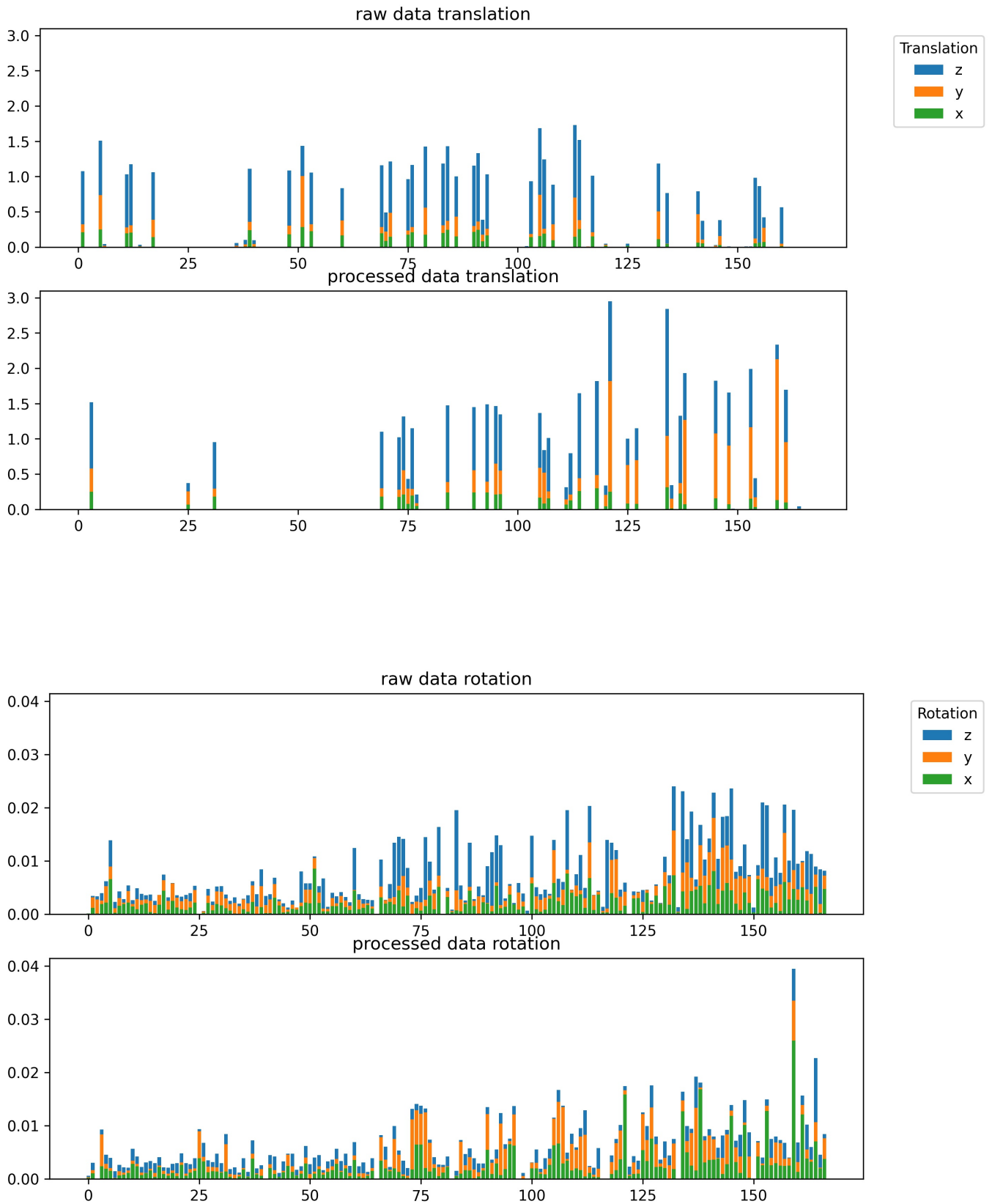


difference: raw_tSNR - processed_tSNR



Topup estimated field coefficients







Single subject QC report generated using eddy quad v1.0.3

When using eddy and its QC tools, we ask you to please reference the papers describing the different aspects of the modelling and corrections. The following suggestion for a methods section and list of references has been tailored for you based on your eddy command line.

METHODS

The susceptibility induced off-resonance field was estimated from spin-echo EPI images acquired with different phase-encode directions (Andersson et al., 2003). This field was passed to "eddy", a tool that combined it with estimating gross subject movement and eddy current-induced distortions (Andersson & Sotiropoulos, 2016). The quality of the dataset was assessed using the eddy QC tools (Bastiani et al., 2019). Slices with signal loss caused by subject movement coinciding with the diffusion encoding were detected and replaced by predictions made by a Gaussian Process (Andersson et al., 2016). Intra-volume subject movement, leading to disjoint slices that when stacked does not amount to a valid volumetric representation of the object, was corrected using slice-to-volume alignment (Andersson et al., 2017).

REFERENCES

Jesper L.R. Andersson, Stefan Skare and John Ashburner. 2003. How to correct susceptibility distortions in spin-echo echo-planar images: application to diffusion tensor imaging. *NeuroImage* 20:870-888

Jesper L.R. Andersson and Stamatios N. Sotiropoulos. 2016. An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging. *NeuroImage* 125:1063-1078

Matteo Bastiani, Michiel Cottaar, Sean P. Fitzgibbon, Sana Suri, Fidel Alfaro-Almagro, Stamatios N. Sotiropoulos, Saad Jbabdi and Jesper L.R. Andersson. 2019. Automated quality control for within and between studies diffusion MRI data using a non-parametric framework for movement and distortion correction. *NeuroImage* 184:801-812

Jesper L.R. Andersson, Mark S. Graham, Eniko Zsoldos and Stamatios N. Sotiropoulos. 2016. Incorporating outlier detection and replacement into a non-parametric framework for movement and distortion correction of diffusion MR images. *NeuroImage* 141:556-572

Jesper L.R. Andersson, Mark S. Graham, Ivana Drobnyak, Hui Zhang, Nicola Filippini and Matteo Bastiani. 2017. Towards a comprehensive framework for movement and distortion correction of diffusion MR images: Within volume movement. *NeuroImage* 152:450-466

Volume-to-volume motion

.	Average abs. motion (mm)	0.35
.	Average rel. motion (mm)	0.13
.	Average x translation (mm)	0.04
.	Average y translation (mm)	-0.26
.	Average z translation (mm)	-0.11
.	Average x rotation (deg)	0.17
.	Average y rotation (deg)	-0.14
.	Average z rotation (deg)	-0.04

Outliers

.	Total outliers (%)	0.01
.	Outliers (b=999 s/mm ²)	0.02
.	Outliers (b=2000 s/mm ²)	0.00
.	Outliers (b=3000 s/mm ²)	0.00
.	Outliers (b=5000 s/mm ²)	0.00
.	Outliers (PE dir=[0. 1. 0.])	0.01

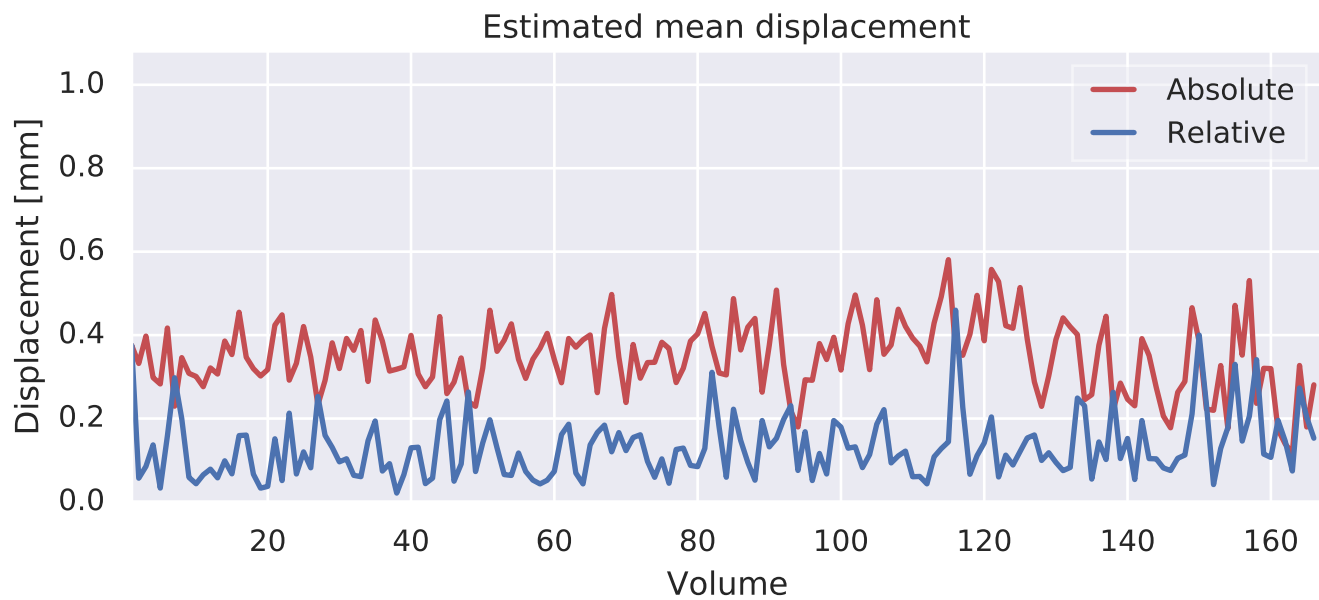
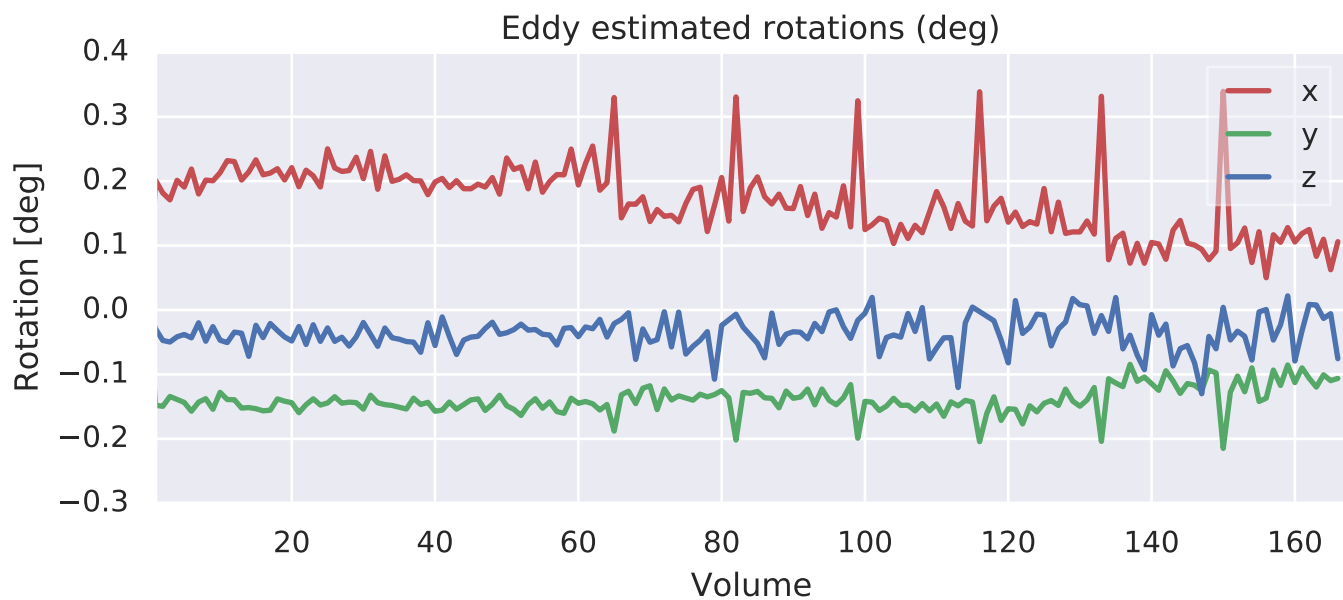
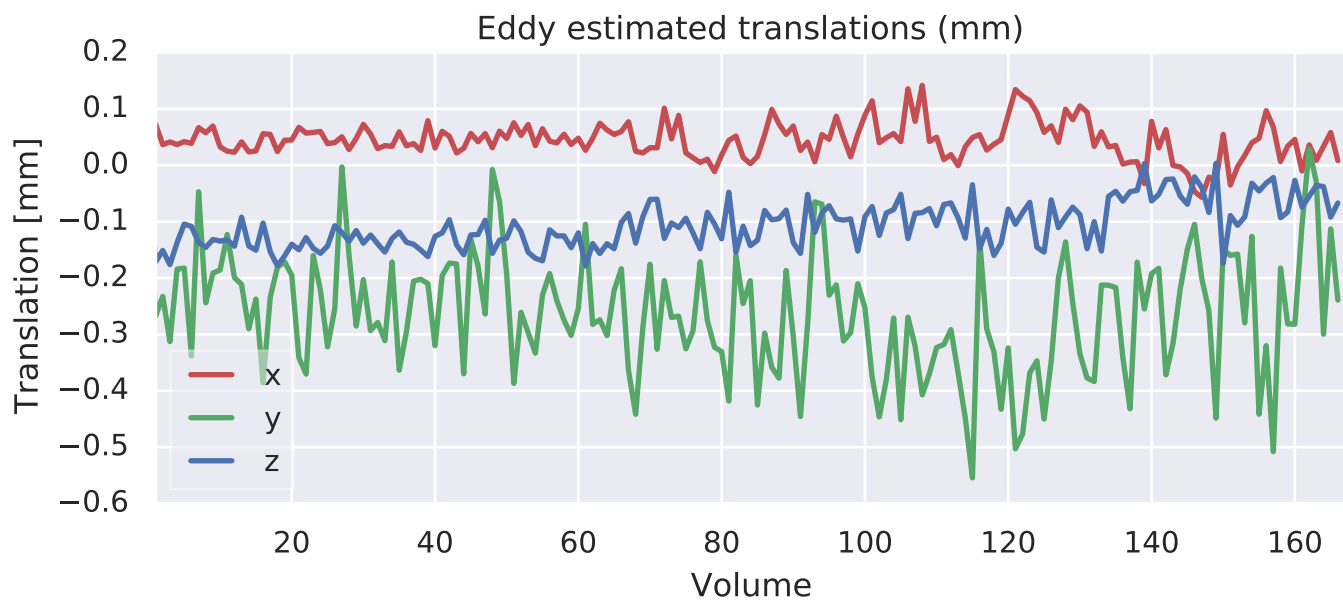
SNR/CNR

.	Average SNR (b=0 s/mm ²)	37.72
.	Average CNR (b=999 s/mm ²)	1.79
.	Average CNR (b=2000 s/mm ²)	1.67
.	Average CNR (b=3000 s/mm ²)	1.55
.	Average CNR (b=5000 s/mm ²)	1.49

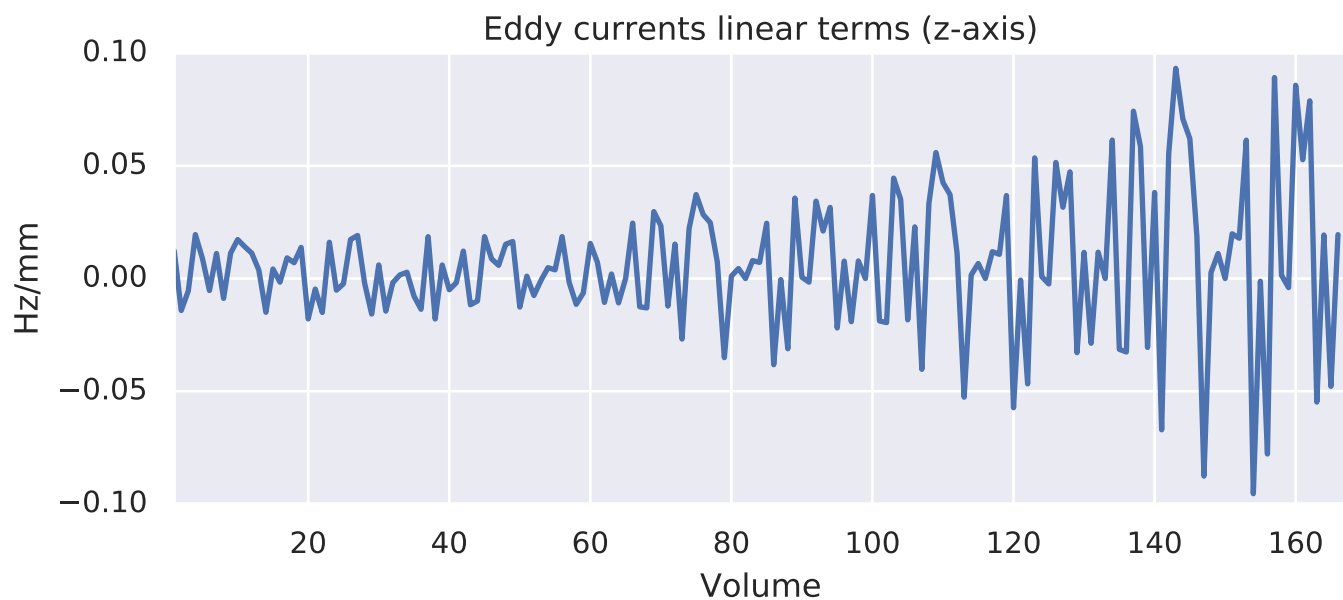
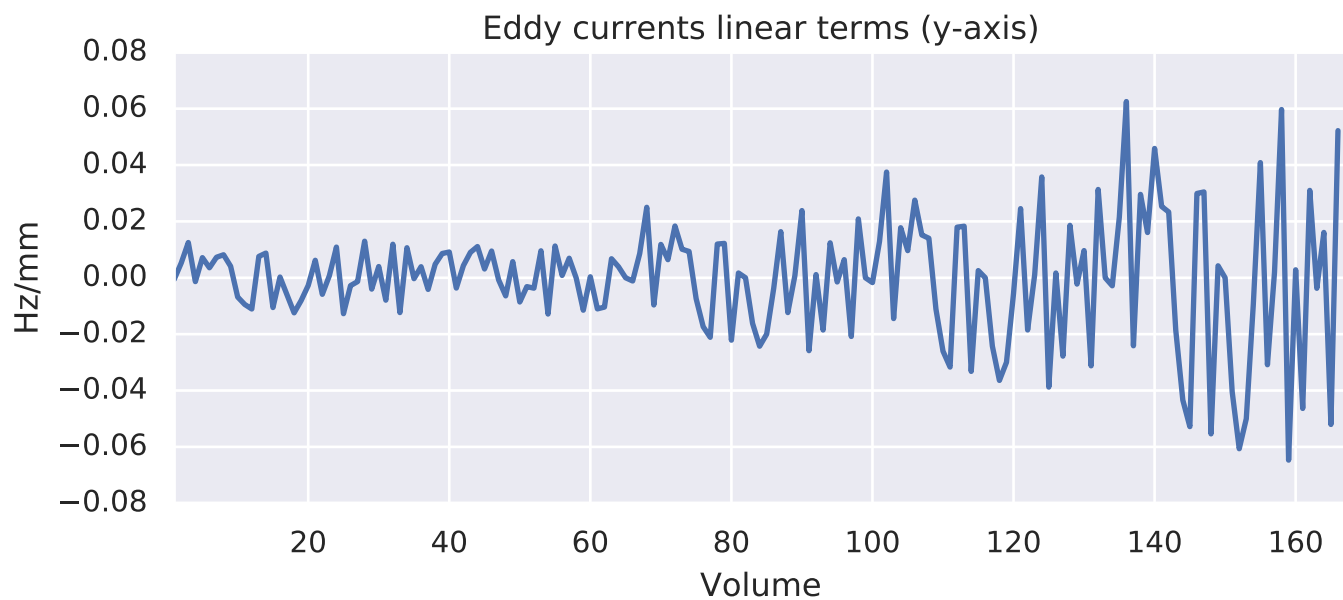
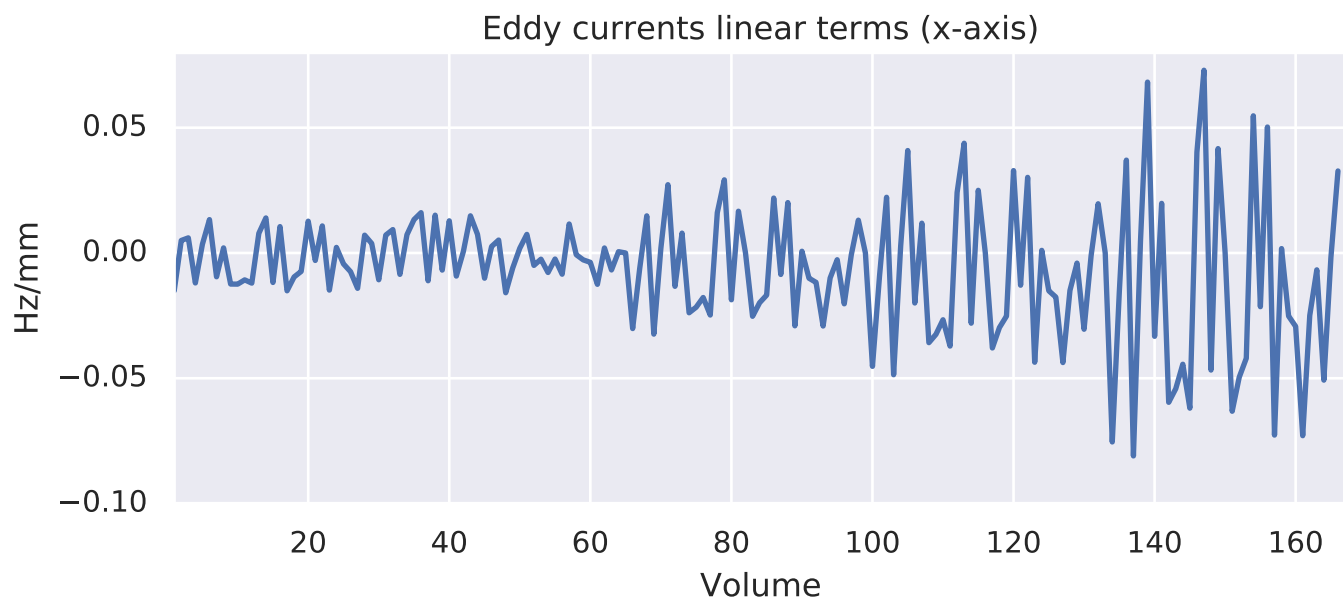
Eddy currents

.	Std Dev EC linear term (x)	0.03
.	Std Dev EC linear term (y)	0.02
.	Std Dev EC linear term (z)	0.03

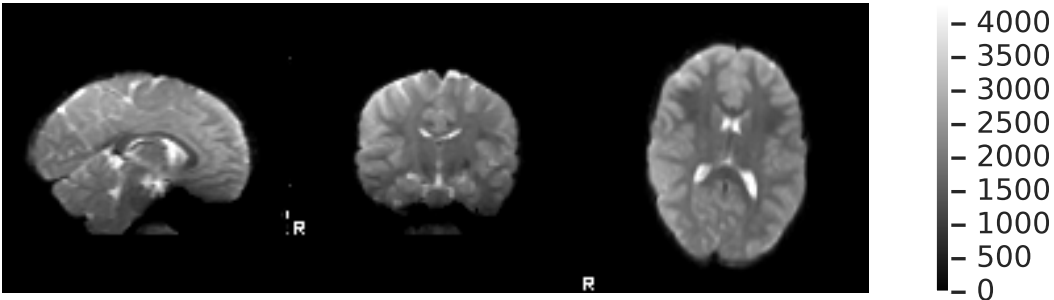
Subject /CECI/proj/pilab/dyslexia_delilab/data_S2V_r6/subjects/21TD/dMRI/preproc/
eddy/21TD_eddy_corr.nii.gz



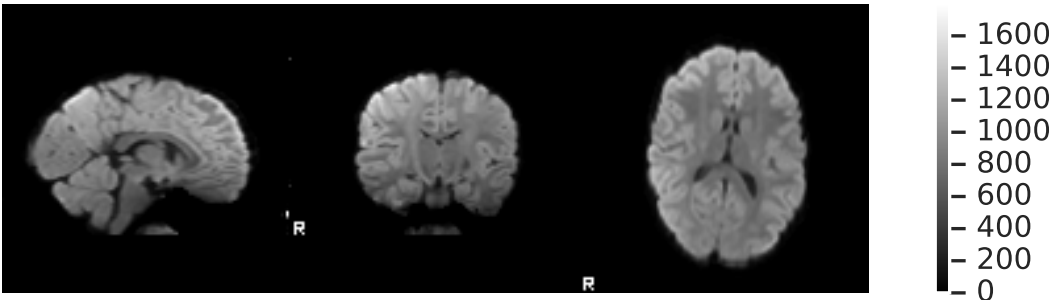
Subject /CECI/proj/pilab/dyslexia_delilab/data_S2V_r6/subjects/21TD/dMRI/preproc/
eddy/21TD_eddy_corr.nii.gz



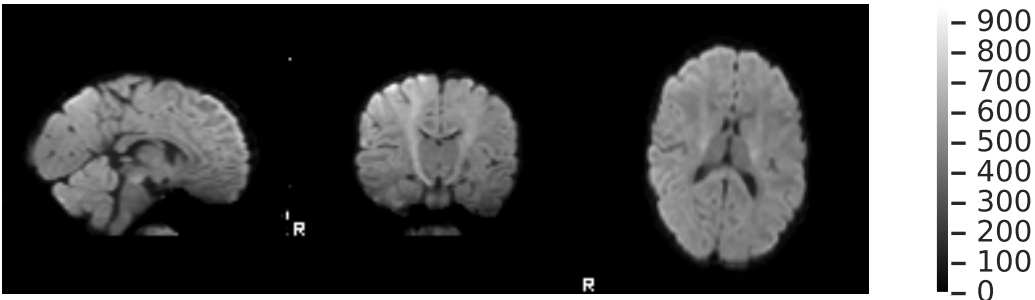
Average DW signal (b=0)



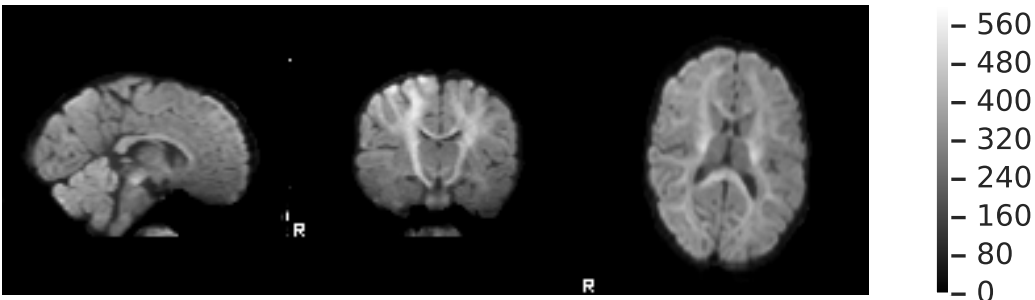
Average DW signal (b=999)



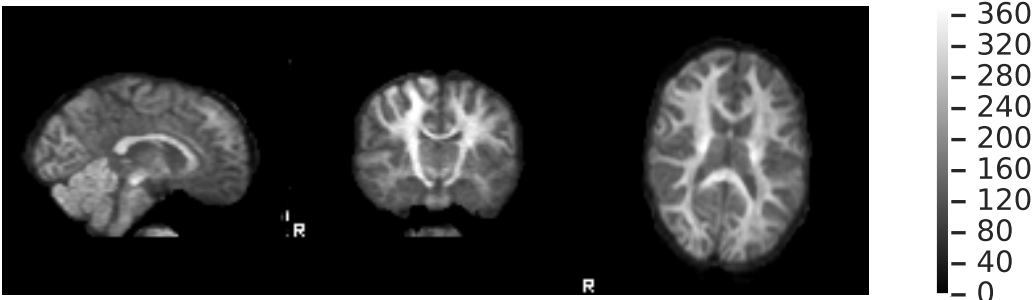
Average DW signal (b=2000)



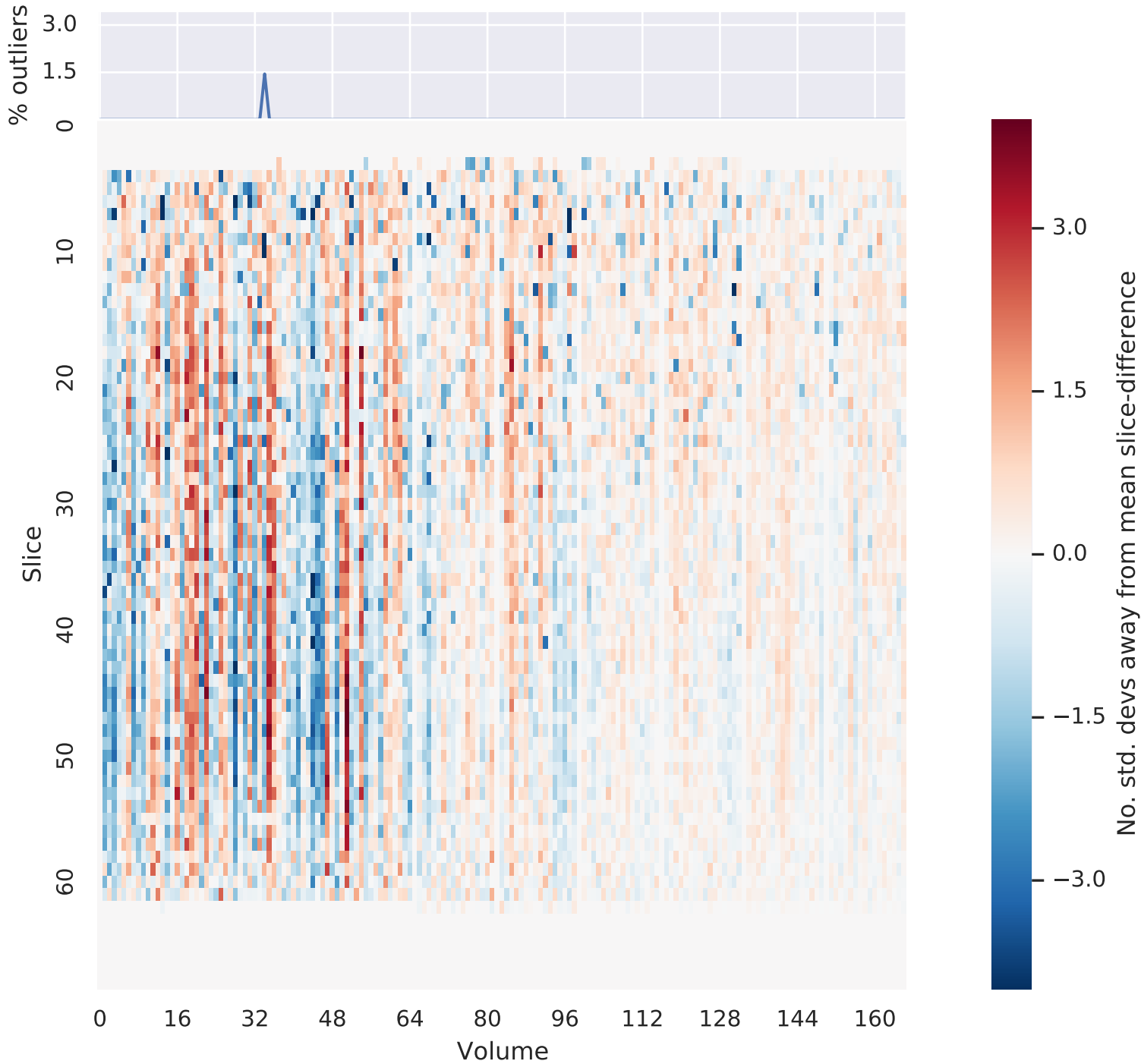
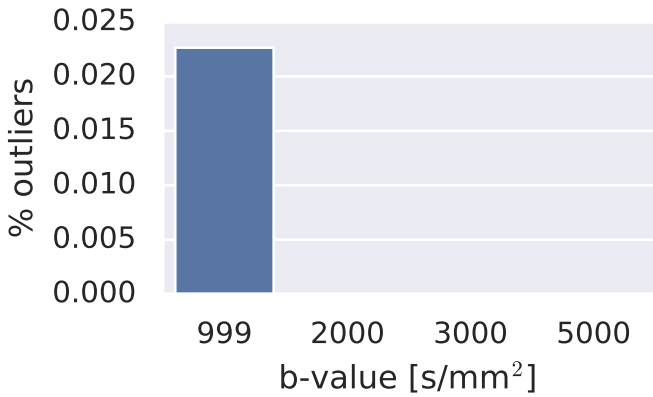
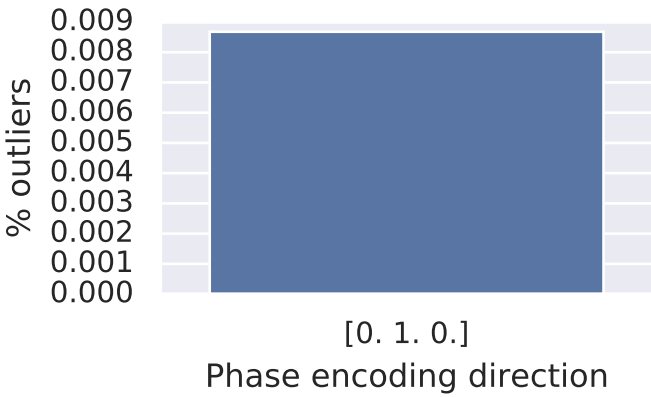
Average DW signal (b=3000)



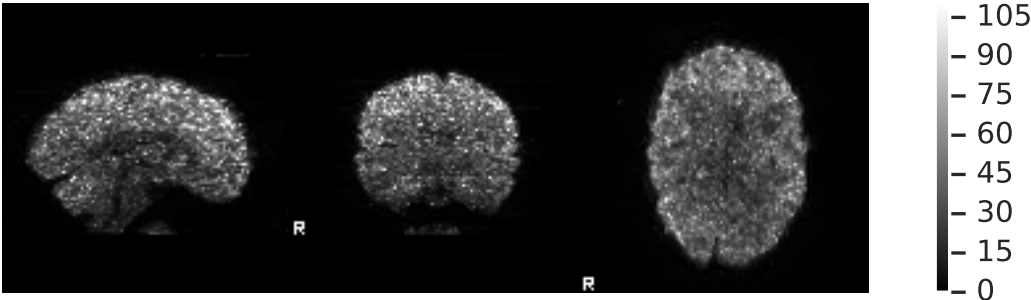
Average DW signal (b=5000)



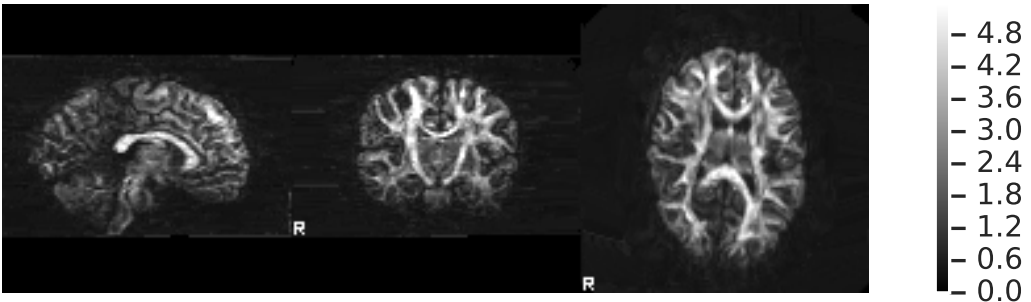
Subject /CECI/proj/pilab/dyslexia_delilab/data_S2V_r6/subjects/21TD/dMRI/preproc/
eddy/21TD_eddy_corr.nii.gz



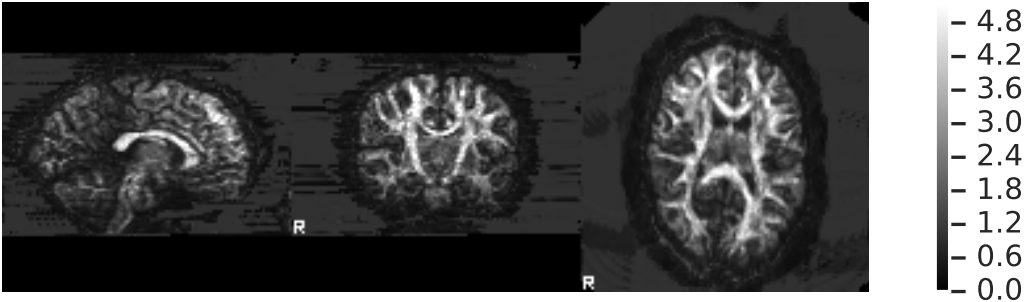
tSNR map (b=0)



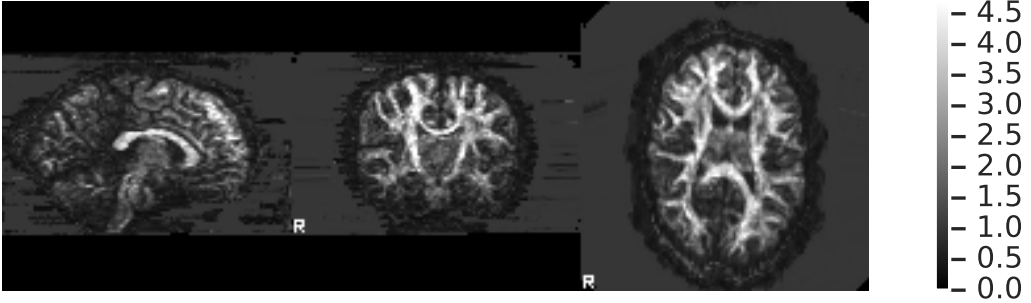
CNR map (b=999)



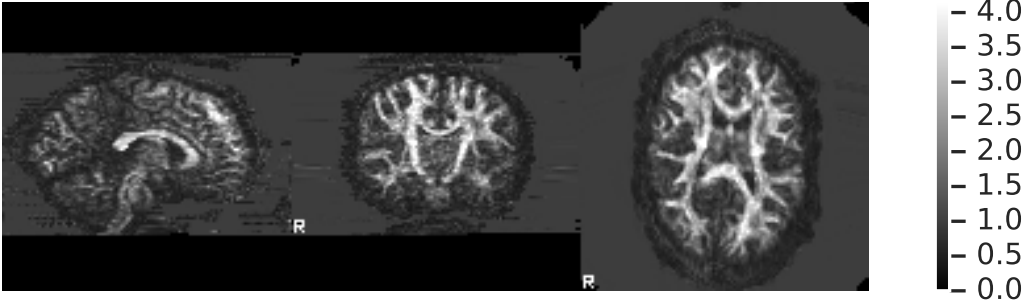
CNR map (b=2000)



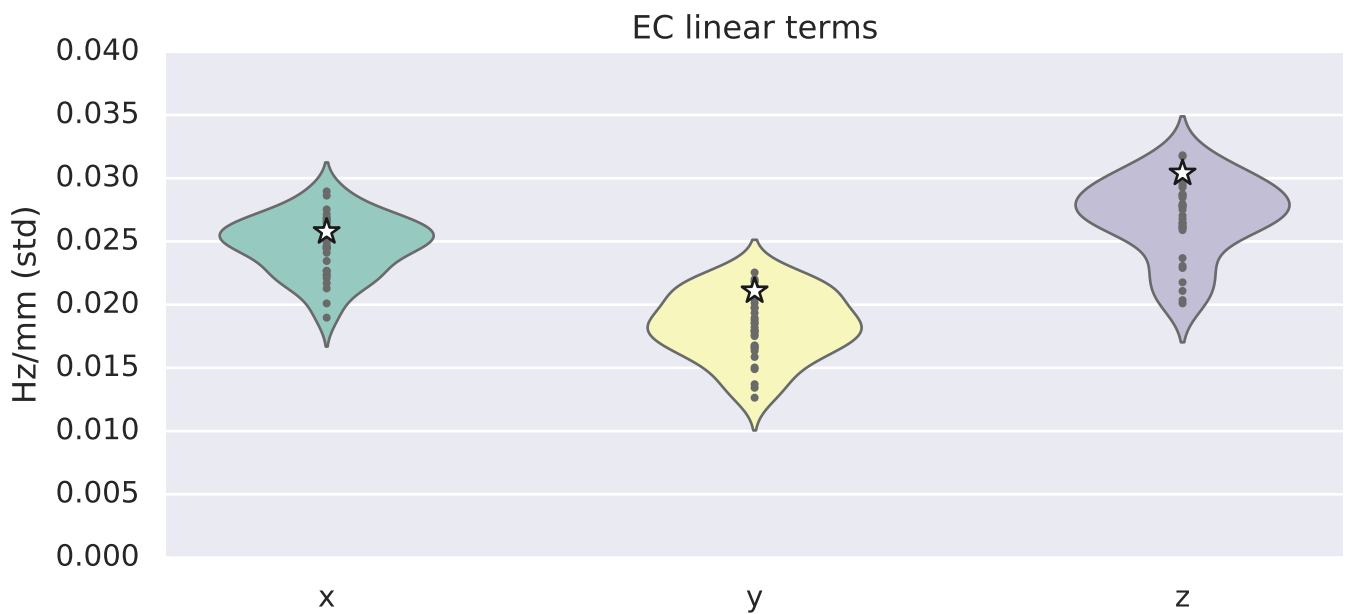
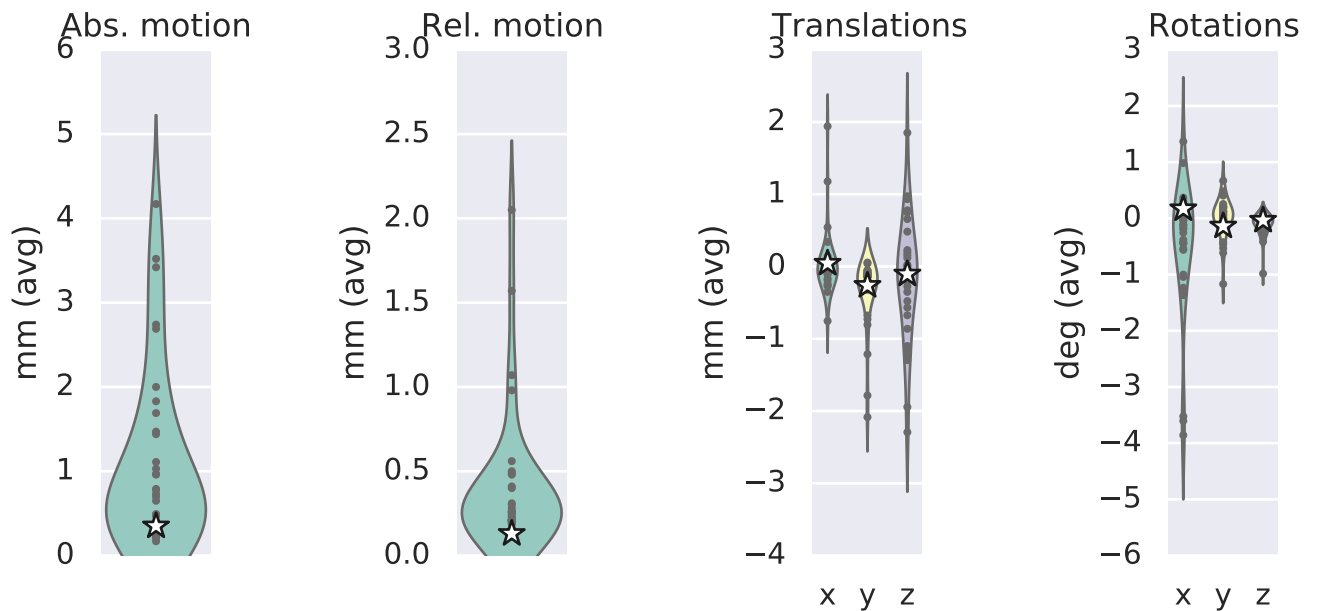
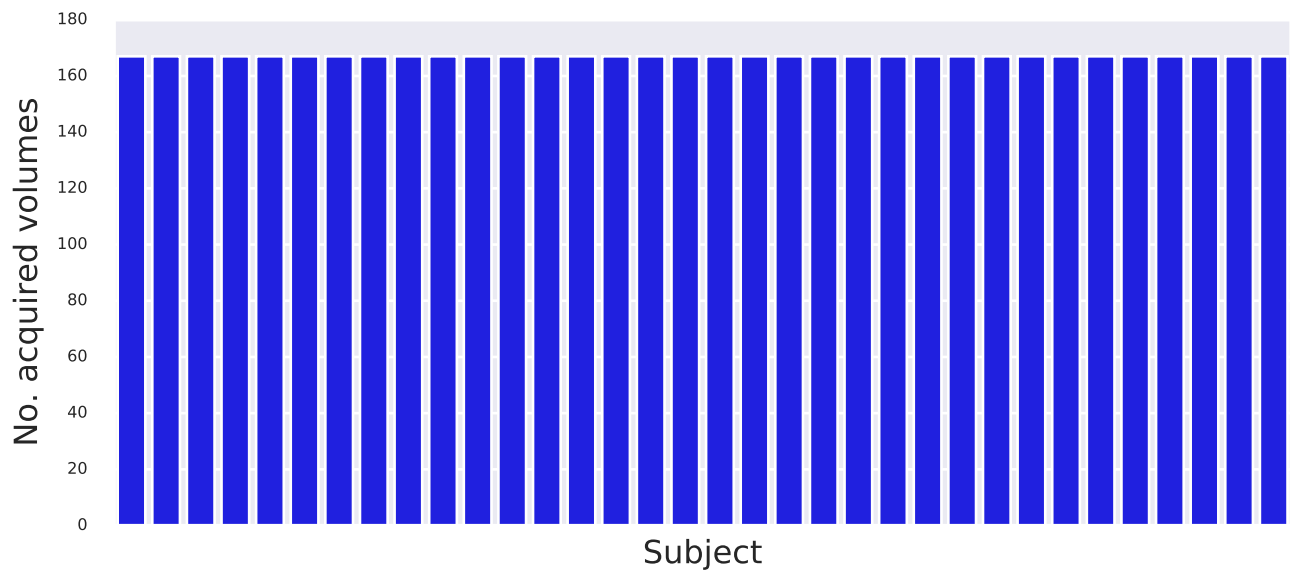
CNR map (b=3000)



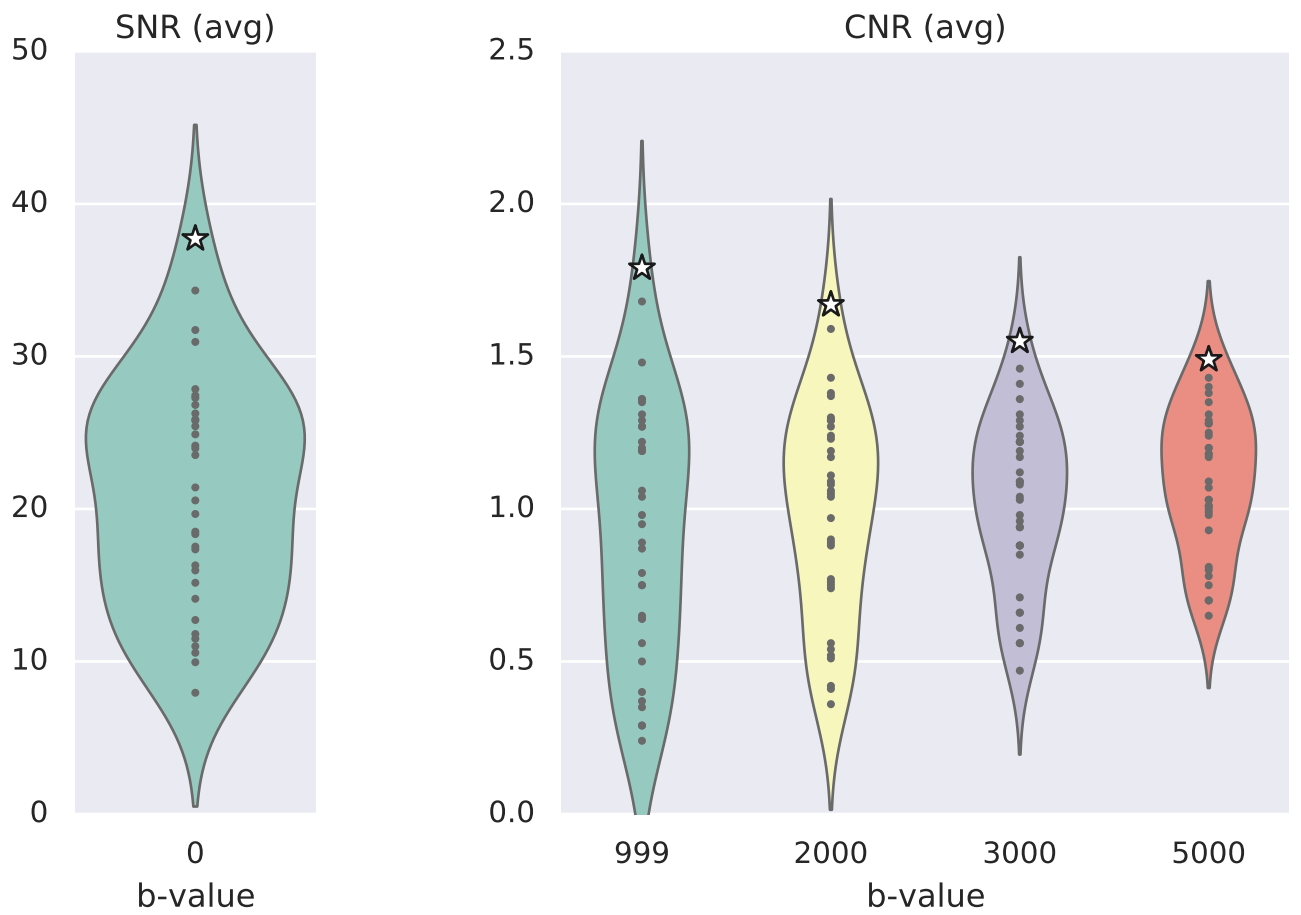
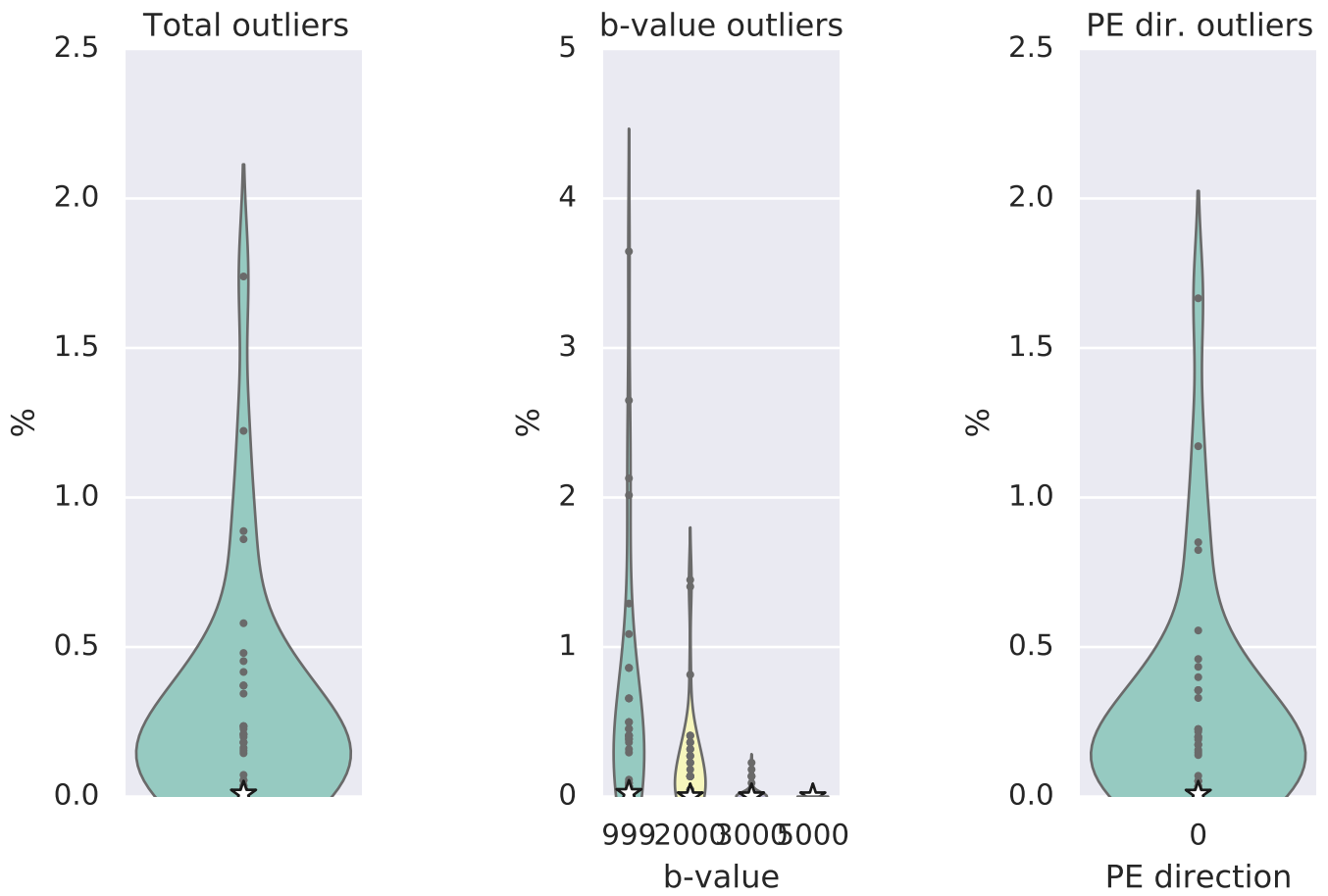
CNR map (b=5000)



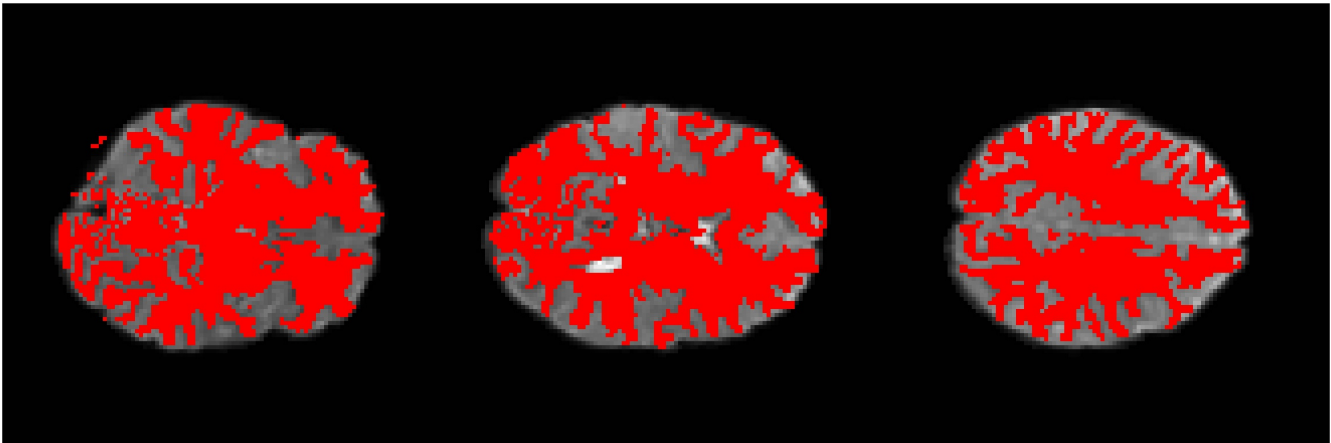
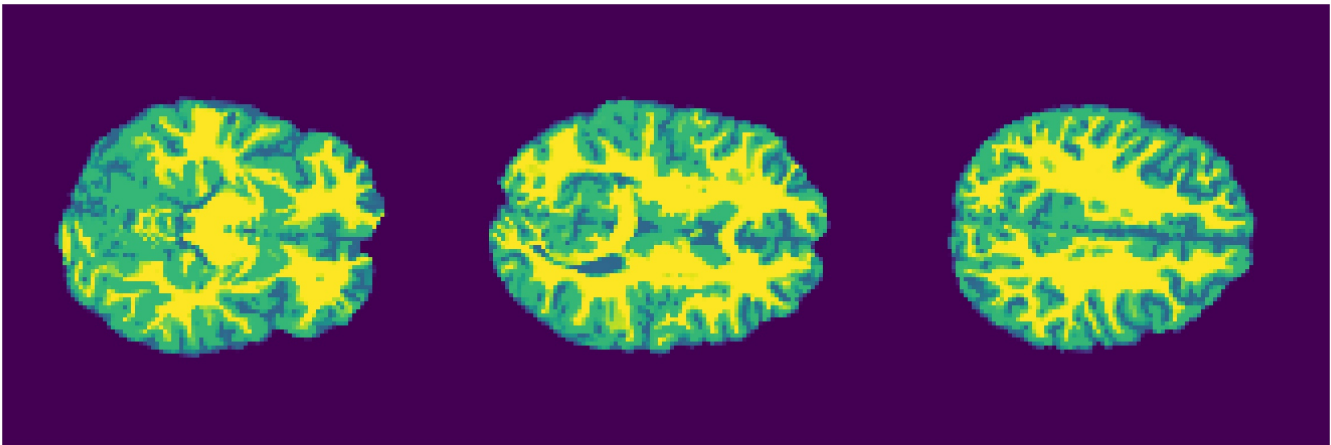
SQUAD: Group report



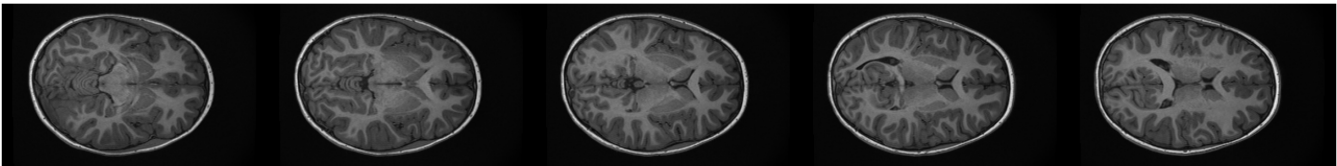
SQUAD: Group report



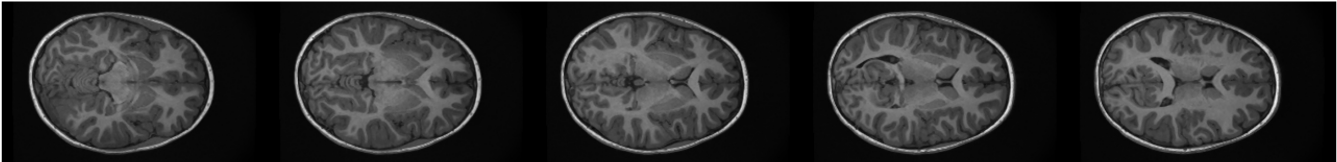
Elikopy : Quality control report - White matter mask - 21TD



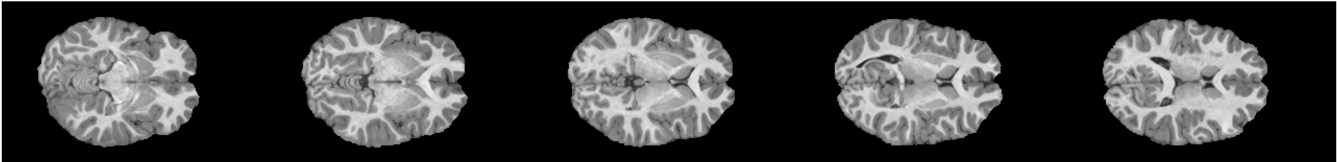
T1



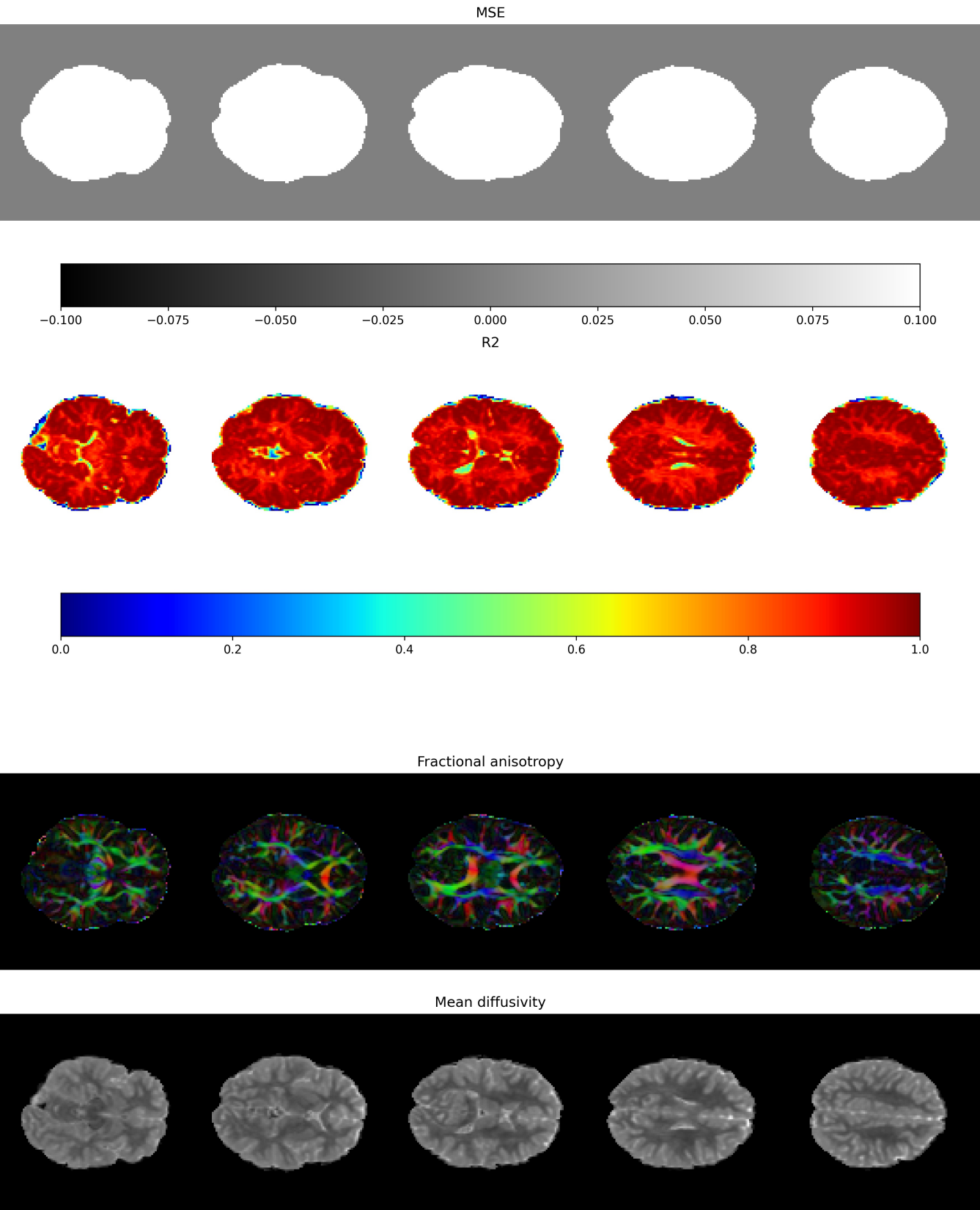
T1 gibbs ringing corrected



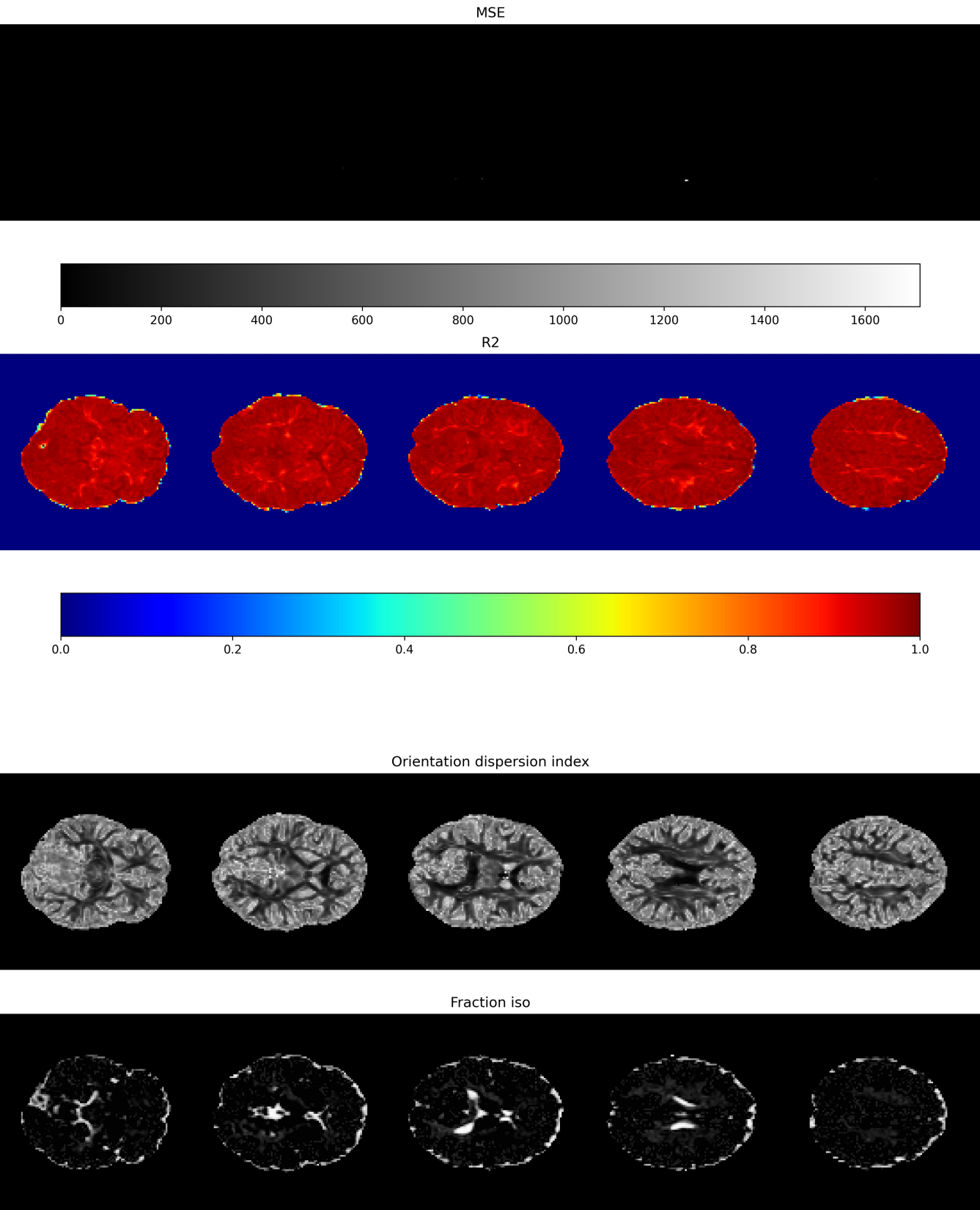
T1 brain extracted



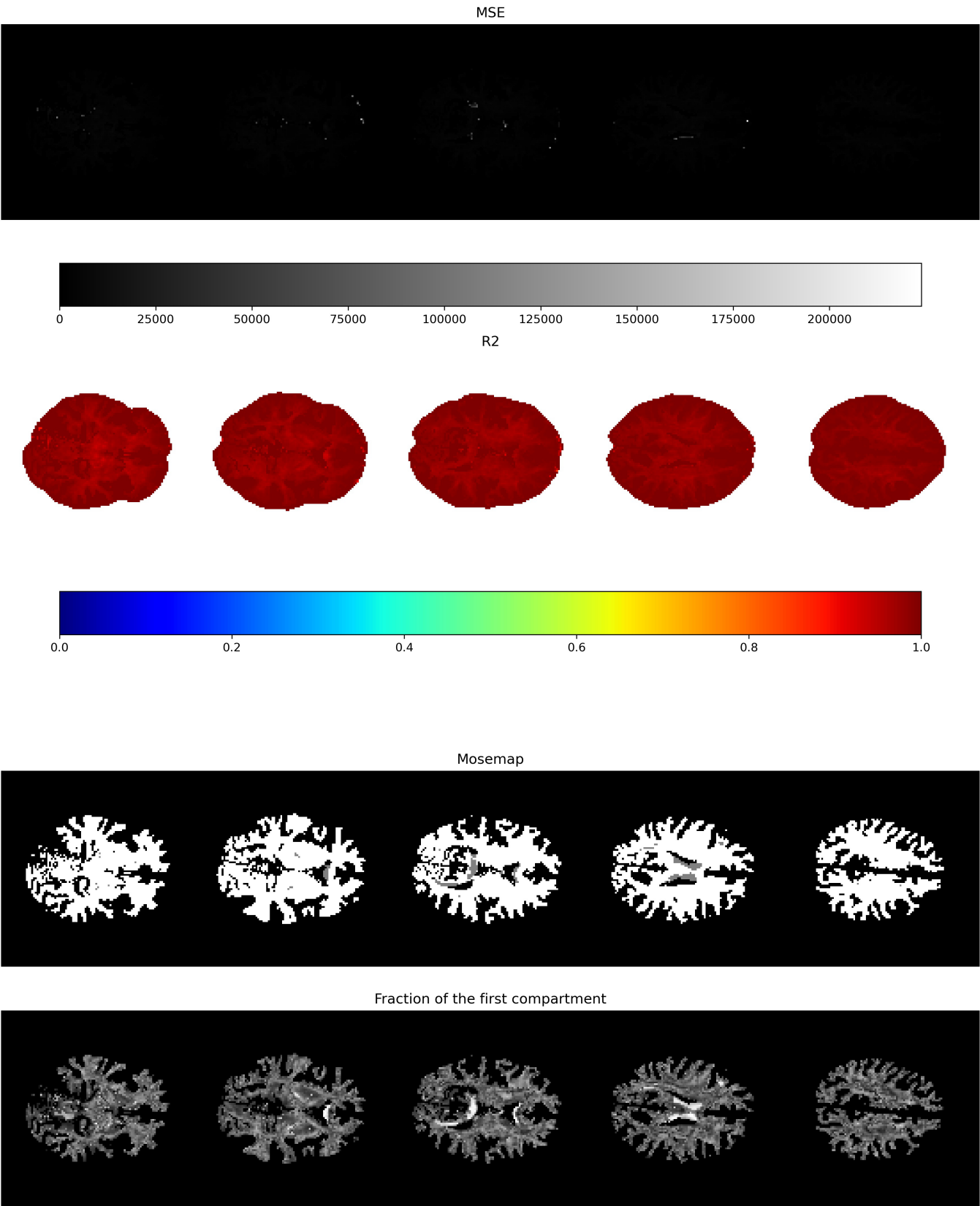
Elikopy : Quality control report - DTI



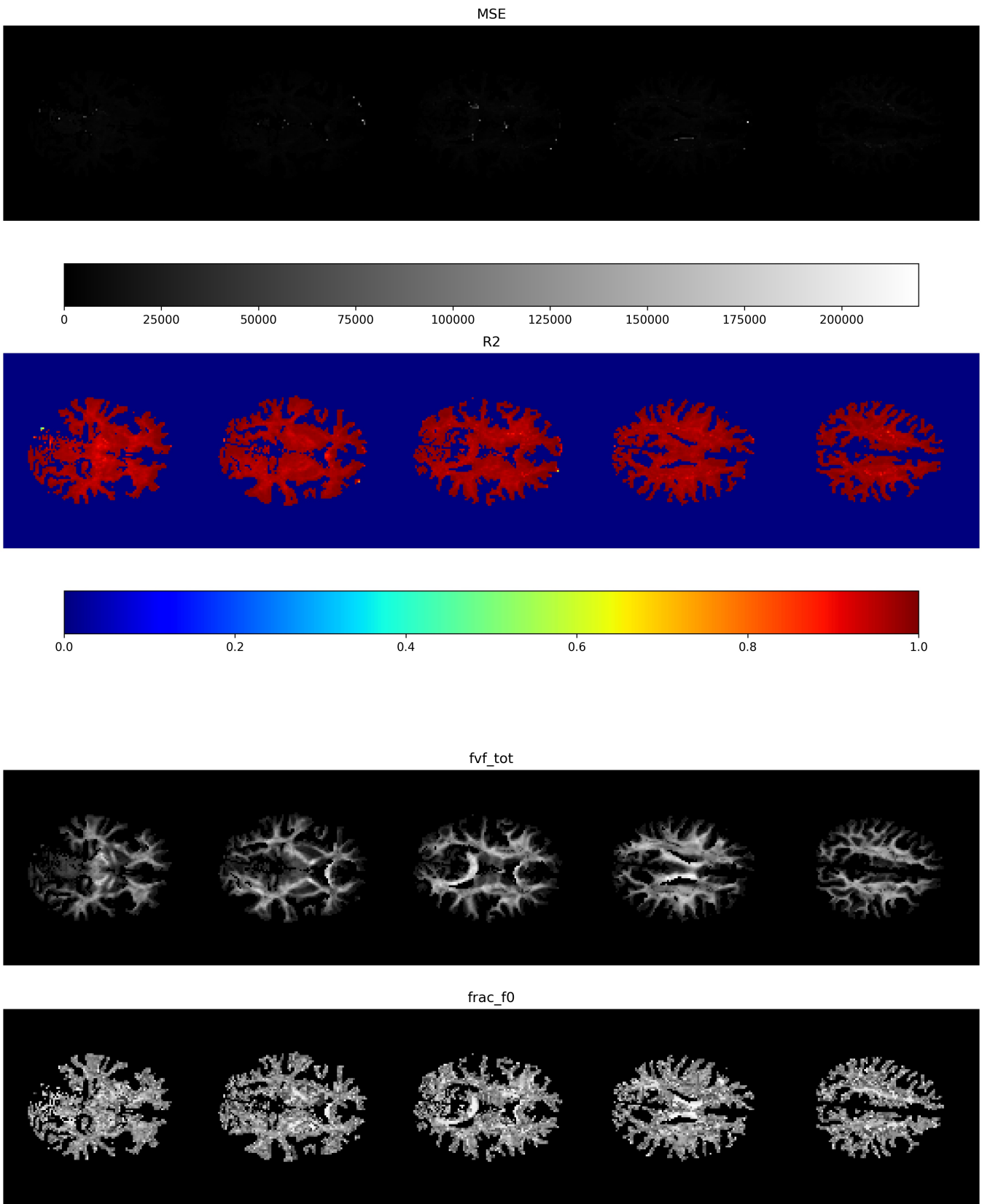
Elikopy : Quality control report - NODDI



Elikopy : Quality control report - DIAMOND



Elikopy : Quality control report - Microstructure fingerprinting



Appendix E

API of the package ElikoPy

ElikoPy API

Quentin Dessain, Mathieu Simon

Jun 05, 2021

ELIKOPY PACKAGE

1.1 Submodules

1.2 elikopy.core module

Elikopy @author: qdessain, msimon

class `elikopy.core.Elikopy(folder_path, cuda=False, slurm=False, slurm_email='example@example.com')`
 Bases: `object`

Main class containing all the necessary function to process and preprocess a specific study.

diamond(`folder_path=None, patient_list_m=None, slurm=None, slurm_email=None, slurm_timeout=None, cpus=None, slurm_mem=None`)

Wrapper function for DIAMOND. Perform diamond and store the data in the `<folder_path>/subjects/<subjects_ID>/dMRI/microstructure/diamond/`.

Parameters

- **folder_path** – path to the root directory.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 14h by a custom timeout.
- **cpus** – Replace the default number of slurm cpus of 4 by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (6096MO by cpu) by a custom amount of ram.

dti(`folder_path=None, patient_list_m=None, slurm=None, slurm_email=None, slurm_timeout=None, slurm_cpus=None, slurm_mem=None`)

Wrapper function for tensor reconstruction and computation of DTI metrics using Weighted Least-Squares. Performs a tensor reconstruction and saves the DTI metrics.

Parameters

- **folder_path** – the path to the root directory.
 - **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
 - **slurm** – Whether to use the Slurm Workload Manager or not.
 - **slurm_email** – Email adress to send notification if a task fails.
-

ElikoPy

- **slurm_timeout** – Replace the default slurm timeout of 1h by a custom timeout.
- **slurm_cpus** – Replace the default number of slurm cpus of 1 by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

export(*folder_path=None, raw=False, preprocessing=False, dti=False, nodd=False, diamond=False, mf=False, wm_mask=False, report=False, preprocessed_first_b0=False, patient_list_m=None, tractography=False*)

Wrapper function for tensor reconstruction and computation of DTI metrics using Weighted Least-Squares. Performs a tensor reconstruction and saves the DTI metrics.

Parameters

- **folder_path** – the path to the root directory.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **raw** – If true, export files corresponding to the parameter's name.
- **preprocessing** – If true, export files corresponding to the parameter's name.
- **dti** – If true, export files corresponding to the parameter's name.
- **nodd** – If true, export files corresponding to the parameter's name.
- **diamond** – If true, export files corresponding to the parameter's name.
- **mf** – If true, export files corresponding to the parameter's name.
- **wm_mask** – If true, export files corresponding to the parameter's name.
- **report** – If true, export files corresponding to the parameter's name.
- **preprocessed_first_b0** – If true, export files corresponding to the parameter's name.
- **tractography** – If true, export files corresponding to the parameter's name.

fingerprinting(*dictionary_path, folder_path=None, CSD_bvalue=None, slurm=None, patient_list_m=None, slurm_email=None, slurm_timeout=None, cpus=None, slurm_mem=None*)

Wrapper function for microstructure estimation. Perform microstructure fingerprinting and store the data in the <folder_path>/subjects/<subjects_ID>/dMRI/microstructure/mf/.

Parameters

- **folder_path** – the path to the root directory.
- **dictionary_path** – Path to the dictionary to use
- **CSD_bvalue** – Define a csd value.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 20h by a custom timeout.
- **cpus** – Replace the default number of slurm cpus of 1 by a custom number of cpus.

- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

noddi (*folder_path=None, patient_list_m=None, force_brain_mask=False, slurm=None, slurm_email=None, slurm_timeout=None, cpus=None, slurm_mem=None*)

Wrapper function for noddi. Perform noddi and store the data in <folder_path>/subjects/<subjects_ID>/dMRI/microstructure/noddi/.

Parameters

- **folder_path** – path to the root directory.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **force_brain_mask** – Force the use of a brain mask even if a whitematter mask exist.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 10h by a custom timeout.
- **cpus** – Replace the default number of slurm cpus of 1 by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

noddi_amico (*folder_path=None, patient_list_m=None, force_brain_mask=False, slurm=None, slurm_email=None, slurm_timeout=None, slurm_cpus=None, slurm_mem=None*)

Wrapper function for noddi amico. Perform noddi and store the data in <folder_path>/subjects/<subjects_ID>/dMRI/microstructure/noddi_amico/.

Parameters

- **folder_path** – path to the root directory.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **force_brain_mask** –
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 10h by a custom timeout.
- **slurm_cpus** – Replace the default number of slurm cpus of 1 by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

noddi_fix_icvf_thresholding (*folder_path=None, patient_list_m=None, fintra_threshold=0.99, fbundle_threshold=0.05, use_brain_mask=False, use_wm_mask=False*)

A function to quickly change the treshold value applied on the icvf metric of noddi without the needs of executing again the full noddi core function.

Parameters

- **folder_path** – path to the root directory.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **fintra_threshold** – Threshold applied on the fintra.
- **fbundle_threshold** – Threshold applied on the fbundle.
- **use_brain_mask** – Set to 0 values outside the brain mask.

1.2. elikopy.core module

ElikoPy

- **use_wm_mask** – Set to 0 values outside the white matter mask.

patient_list(*folder_path=None, bids_path=None*)

Verify the validity of all the nifti present in the root folder. If some nifti does not possess an associated bval and bvec file, they are discarded and the user is notified by a summary file named `patient_error.json` generated in the out sub-directory. All valid patients are stored in a file named `patient_list.json`

Parameters

- **folder_path** – Path to root folder of the study.
- **bids_path** – Path to the optional folder containing subjects' data in the BIDS format.

patientlist_wrapper(*function, func_args, folder_path=None, patient_list_m=None, filename=None, function_name=None, slurm=False, slurm_email=None, slurm_timeout=None, cpus=None, slurm_mem=None*)

A wrapper function that apply a function given as an argument to every patients of the study. The wrapped function must takes two arguments as input, the `patient_name` and the path to the root of the study.

Parameters

- **folder_path** – path to the root directory.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **function** – The pointer to the function (only used without slurm)
- **func_args** – Additional arguments to pass to the wrapped function (only used without slurm)
- **filename** – The name of the file containing the wrapped function (only used with slurm)
- **function_name** – The name of the wrapped function (only used with slurm)
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 20h by a custom timeout.
- **cpus** – Replace the default number of used cpus by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

preproc(*folder_path=None, reslice=False, reslice_addSlice=False, denoising=False, gibbs=False, topup=False, topupConfig=None, forceSynb0DisCo=False, eddy=False, biasfield=False, biasfield_bsplineFitting=[100, 3], biasfield_convergence=[1000, 0.001], patient_list_m=None, starting_state=None, bet_median_radius=2, bet_numpass=1, bet_dilate=2, cuda=None, cuda_name='eddy_cuda10.1', s2v=[0, 5, 1, 'trilinear'], olrep=[False, 4, 250, 'sw'], slurm=None, slurm_email=None, slurm_timeout=None, cpus=None, slurm_mem=None, qc_reg=True, niter=5, slspec_gc_path=None, report=True*)

Wrapper function for the preprocessing. Perform brain extraction and optionally reslicing, denoising, gibbs correction, susceptibility field estimation using topup, movement correction using eddy and biasfield correction. Generated data are stored in `bet`, `reslice`, `mppca`, `gibbs`, `topup`, `eddy`, `biasfield` and final directory located in the folder `<folder_path>/subjects/<subjects_ID>/dMRI/preproc`.

Parameters

- **folder_path** – the path to the root directory.
- **reslice** – If true, data will be resliced with a new voxel resolution of 2*2*2.
- **reslice_addSlice** – If true, an additional slice will be added to each volume to allow gradient cycling eddy motion correction.

- **denoising** – If true, PCA-based denoising using the Marcenko-Pastur distribution will be performed.
- **gibbs** – If true, Gibbs ringing artifacts of images volumes will be suppressed.
- **topup** – If true, topup will estimate and correct susceptibility induced distortions.
- **topupConfig** – If not None, topup will use these additionnal parameters based on the supplied config file.
- **forceSynb0DisCo** – If true, topup will always estimate the susceptibility field using the T1 structural image.
- **eddy** – If true, eddy will correct eddy currents and movements in diffusion data.
- **biasfield** – If true, low frequency intensity non-uniformity present in MRI image data known as a bias or gain field will be corrected.
- **biasfield_bsplineFitting** – Define the initial mesh resolution in mm and the bspline order of the biasfield correction tool.
- **biasfield_convergence** – Define the maximum number of iteration and the convergences threshold of the biasfield correction tool.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **starting_state** – Manually set which step of the preprocessing to execute first. Could either be None, denoising, gibbs, topup, eddy, biasfield, report or post_report.
- **bet_median_radius** – Radius (in voxels) of the applied median filter during bet.
- **bet_numpass** – Number of pass of the median filter during bet.
- **bet_dilate** – Number of iterations for binary dilation during bet.
- **cuda** – If true, eddy will run on cuda with the command name specified in cuda_name.
- **cuda_name** – name of the eddy command to run when cuda==True.
- **s2v** – list of parameters eddy for slice-to-volume correction (see Eddy FSL documentation): [mporder,s2v_niter,s2v_lambda,s2v_interp].
- **olrep** – list of parameters eddy outlier replacement (see Eddy FSL documentation): [re-pol,ol_nstd,ol_nvox,ol_type].
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout by a custom timeout.
- **cpus** – Replace the default number of slurm cpus by a custom number of cpus of using slum, or for standard processing, its the number of core available for processing
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task by a custom amount of ram.
- **qc_reg** – If true, the motion registration step of the quality control will be performed.
- **niter** – Define the number of iterations for eddy volume-to-volume
- **slspec_gc_path** – Path to the folder containing volume specific slice-specification for eddy. If not None, eddy motion correction with gradient cycling will be performed.
- **report** – If False, no quality report will be generated.

1.2. elikopy.core module

ElikoPy

```
randomise_all(folder_path=None, randomise_numberofpermutation=5000, skeletonised=True,  
               metrics_dic={'FA': 'dti', '_diamond_kappa': 'diamond', '_mf_fyf_tot': 'mf', '_noddi_odi':  
                             'noddi'}, regionWiseMean=True, slurm=None, slurm_email=None, slurm_timeout=None,  
               cpus=None, slurm_mem=None)
```

Wrapper function for randomise_all. Perform tract base spatial statistics between the control data and case data and region wise stats. DTI needs to have been performed on the data first !!

Parameters

- **folder_path** – path to the root directory.
- **randomise_numberofpermutation** – Define the number of permutation used by randomize.
- **skeletonised** – If True, randomize will be using the only skeleton instead of the whole brain.
- **metrics_dic** – Dictionary containing multiple metrics. For each metrics, the metric name is the key and the folder metric's is the value.
- **regionWiseMean** – If true, csv containing atlas-based region wise mean will be generated.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 20h by a custom timeout.
- **cpus** – Replace the default number of used cpus by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

```
regall(folder_path=None, grp1=None, grp2=None, metrics_dic={'_diamond_kappa': 'diamond',  
                  '_mf_fyf_tot': 'mf', '_noddi_odi': 'noddi'}, slurm=None, slurm_email=None, slurm_timeout=None,  
        cpus=None, slurm_mem=None)
```

Register all the subjects' metric passed in parameters into a common space using the transformation computed when registering the FA. The function regall_FA needs to be applied first!

Parameters

- **folder_path** – path to the root directory.
- **grp1** – List of number corresponding to the type of the patients to put in the first group.
- **grp2** – List of number corresponding to the type of the patients to put in the second group.
- **metrics_dic** – Dictionary containing multiple metrics. For each metrics, the metric name is the key and the folder metric's is the value.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 20h by a custom timeout.
- **cpus** – Replace the default number of used cpus (1) by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

```
regall_FA(folder_path=None, grp1=None, grp2=None, starting_state=None, registration_type='-T',  
           postreg_type='-S', prestats_treshold=0.2, slurm=None, slurm_email=None,  
           slurm_timeout=None, cpus=None, slurm_mem=None)
```

Register all the subjects' FA computed using the DTI model into a common space.

Parameters

- **folder_path** – path to the root directory.
- **grp1** – List of number corresponding to the type of the patients to put in the first group.
- **grp2** – List of number corresponding to the type of the patients to put in the second group.
- **starting_state** – Manually set which step of TBSS to execute first. Could either be None, reg, post_reg, prestats, design or randomise.
- **registration_type** – Define the argument used by the tbss command tbss_2_reg. Could either be '-T', '-t' or '-n'. If '-T' is used, a FMRIB58_FA standard-space image is used. If '-t' is used, a custom image is used. If '-n' is used, every FA image is align to every other one, identify the “most representative” one, and use this as the target image.
- **postreg_type** – Define the argument used by the tbss command tbss_3_postreg. Could either be '-S' or '-T'. If you wish to use the FMRIB58_FA mean FA image and its derived skeleton, instead of the mean of your subjects in the study, use the '-T' option. Otherwise, use the '-S' option.
- **prestats_treshold** – Thresholds the mean FA skeleton image at the chosen threshold during prestats.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 20h by a custom timeout.
- **cpus** – Replace the default number of used cpus of by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

tbss(*folder_path=None, grp1=None, grp2=None, starting_state=None, last_state=None, registration_type='-T', postreg_type='-S', prestats_treshold=0.2, randomise_numberofpermutation=5000, slurm=None, slurm_email=None, slurm_timeout=None, slurm_tasks=None, slurm_mem=None*)

Wrapper function for TBSS. Perform tract base spatial statistics between the control data and case data. DTI needs to have been performed on the data first !!

Parameters

- **folder_path** – path to the root directory.
- **grp1** – List of number corresponding to the type of the patients to put in the first group.
- **grp2** – List of number corresponding to the type of the patients to put in the second group.
- **starting_state** – Manually set which step of TBSS to execute first. Could either be None, reg, post_reg, prestats, design or randomise.
- **last_state** – Manually set which step of TBSS to execute last. Could either be None, preproc, reg, post_reg, prestats, design or randomise.
- **registration_type** – Define the argument used by the tbss command tbss_2_reg. Could either be '-T', '-t' or '-n'. If '-T' is used, a FMRIB58_FA standard-space image is used. If '-t' is used, a custom image is used. If '-n' is used, every FA image is align to every other one, identify the “most representative” one, and use this as the target image.
- **postreg_type** – Define the argument used by the tbss command tbss_3_postreg. Could either be '-S' or '-T'. If you wish to use the FMRIB58_FA mean FA image and its derived

ElikoPy

skeleton, instead of the mean of your subjects in the study, use the ‘-T’ option. Otherwise, use the ‘-S’ option.

- **prestats_treshold** – Thresholds the mean FA skeleton image at the chosen threshold during prestats.
- **randomise_numberofpermutation** – Define the number of permutations.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 20h by a custom timeout.
- **slurm_tasks** – Replace the default number of slurm task of 8 by a custom number of tasks.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

white_mask(*folder_path=None, patient_list_m=None, corr_gibbs=True, forceUsePowerMap=False, debug=False, slurm=None, slurm_email=None, slurm_timeout=None, cpus=None, slurm_mem=None*)

Wrapper function for whitematter mask computation. Compute a white matter mask of the diffusion data for each patient based on T1 volumes or on diffusion data if T1 is not available. Otherwise, compute the whitematter mask based on an anisotropic power map.

Parameters

- **folder_path** – path to the root directory.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **corr_gibbs** – Correct for gibbs oscillation.
- **forceUsePowerMap** – Force the use of an AnisotropicPower map for the white matter mask generation.
- **debug** – If true, additional intermediate output will be saved.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout of 3h by a custom timeout.
- **cpus** – Replace the default number of slurm cpus of 1 by a custom number of cpus.
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task (8096MO by cpu) by a custom amount of ram.

elikopy.core.dicom_to_nifti(*folder_path*)

Convert dicom data into nifti. Converted dicom are then moved to a sub-folder named original_data Parameters

Parameters **folder_path** – Path to root folder containing all the dicom

1.3 elikopy.individual_subject_processing module

`elikopy.individual_subject_processing.diamond_solo(folder_path, p, core_count=4)`

Perform diamond and store the data in <folder_path>/subjects/<subjects_ID>/dMRI/microstructure/diamond/.

Parameters

- **folder_path** – the path to the root directory.
- **p** – The name of the patient.
- **core_count** – Number of allocated cpu core.

`elikopy.individual_subject_processing.dti_solo(folder_path, p)`

Tensor reconstruction and computation of DTI metrics using Weighted Least-Squares. Performs a tensor reconstruction and saves the DTI metrics.

Parameters

- **folder_path** – the path to the root directory.
- **p** – The name of the patient.

`elikopy.individual_subject_processing.mf_solo(folder_path, p, dictionary_path, CSD_bvalue=None, core_count=1)`

Perform microstructure fingerprinting and store the data in the <folder_path>/subjects/<subjects_ID>/dMRI/microstructure/mf/.

Parameters

- **folder_path** – the path to the root directory.
- **p** – The name of the patient.
- **dictionary_path** – Path to the dictionary to use
- **CSD_bvalue** – Define a csd value.
- **core_count** – Define the number of available core.

`elikopy.individual_subject_processing.noddi_amico_solo(folder_path, p)`

Perform noddi and store the data in the <folder_path>/subjects/<subjects_ID>/dMRI/microstructure/noddi_amico/.

Parameters

- **folder_path** – the path to the root directory.
- **p** – The name of the patient.

`elikopy.individual_subject_processing.noddi_solo(folder_path, p, force_brain_mask=False, lambda_iso_diff=3e-09, lambda_par_diff=1.7e-09, use_amico=False, core_count=1)`

Perform noddi and store the data in the <folder_path>/subjects/<subjects_ID>/dMRI/microstructure/noddi/.

Parameters

- **folder_path** – the path to the root directory.
- **p** – The name of the patient.
- **force_brain_mask** – Force the use of a brain mask even if a whitematter mask exist.
- **lambda_iso_diff** – Define the noddi lambda_iso_diff parameters.
- **lambda_par_diff** – Define the noddi lambda_par_diff parameters.
- **use_amico** – If true, use the amico optimizer.

1.3. elikopy.individual_subject_processing module

ElikoPy

```
elikopy.individual_subject_processing.preproc_solo(folder_path, p, reslice=False,
                                                  reslice_addSlice=False, denoising=False,
                                                  gibbs=False, topup=False, topupConfig=None,
                                                  forceSynb0DisCo=False, eddy=False,
                                                  biasfield=False, biasfield_bsplineFitting=[100,
                                                  3], biasfield_convergence=[1000, 0.001],
                                                  starting_state=None, bet_median_radius=2,
                                                  bet_numpass=1, bet_dilate=2, cuda=False,
                                                  cuda_name='eddy_cuda10.1', s2v=[0, 5, 1,
                                                  'trilinear'], olrep=[False, 4, 250, 'sw'],
                                                  qc_reg=True, core_count=1, niter=5,
                                                  report=True, slspec_gc_path=None)
```

Perform brain extraction and optionally reslicing, denoising, gibbs correction, susceptibility field estimation using topup, movement correction using eddy and biasfield correction. Generated data are stored in bet, reslice, mppca, gibbs, topup, eddy, biasfield and final directory located in the folder `folder_path/subjects/<subjects_ID>/dMRI/preproc`.

Parameters

- **folder_path** – the path to the root directory.
- **p** – The name of the patient.
- **reslice** – If true, data will be resliced with a new voxel resolution of 2*2*2.
- **reslice_addSlice** – If true, an additional slice will be added to each volume to allow gradient cycling eddy motion correction.
- **denoising** – If true, PCA-based denoising using the Marcenko-Pastur distribution will be performed.
- **gibbs** – If true, Gibbs ringing artifacts of images volumes will be suppressed.
- **topup** – If true, topup will estimate and correct susceptibility induced distortions.
- **topupConfig** – If not None, topup will use these additionnal parameters based on the supplied config file.
- **forceSynb0DisCo** – If true, topup will always estimate the susceptibility field using the T1 structural image.
- **eddy** – If true, eddy will correct eddy currents and movements in diffusion data.
- **biasfield** – If true, low frequency intensity non-uniformity present in MRI image data known as a bias or gain field will be corrected.
- **biasfield_bsplineFitting** – Define the initial mesh resolution in mm and the bspline order of the biasfield correction tool.
- **biasfield_convergence** – Define the maximum number of iteration and the convergences threshold of the biasfield correction tool.
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.
- **starting_state** – Manually set which step of the preprocessing to execute first. Could either be None, denoising, gibbs, topup, eddy, biasfield, report or post_report.
- **bet_median_radius** – Radius (in voxels) of the applied median filter during bet.
- **bet_numpass** – Number of pass of the median filter during bet.
- **bet_dilate** – Number of iterations for binary dilation during bet.
- **cuda** – If true, eddy will run on cuda with the command name specified in `cuda_name`.

- **cuda_name** – name of the eddy command to run when `cuda==True`.
- **s2v** – list of parameters eddy for slice-to-volume correction (see Eddy FSL documentation): `[mporder,s2v_niter,s2v_lambda,s2v_interp]`.
- **olrep** – list of parameters eddy outlier replacement (see Eddy FSL documentation): `[repol,ol_nstd,ol_nvox,ol_type]`.
- **slurm** – Whether to use the Slurm Workload Manager or not.
- **slurm_email** – Email adress to send notification if a task fails.
- **slurm_timeout** – Replace the default slurm timeout by a custom timeout.
- **cpus** – Replace the default number of slurm cpus by a custom number of cpus of using slum, or for standard processing, its the number of core available for processing
- **slurm_mem** – Replace the default amount of ram allocated to the slurm task by a custom amount of ram.
- **qc_reg** – If true, the motion registration step of the quality control will be performed.
- **niter** – Define the number of iterations for eddy volume-to-volume
- **slspec_gc_path** – Path to the folder containing volume specific slice-specification for eddy. If not None, eddy motion correction with gradient cycling will be performed.
- **report** – If False, no quality report will be generated.

`elikopy.individual_subject_processing.report_solo(folder_path, patient_path, slices=None, short=False)`

Legacy report function.

Parameters

- **folder_path** – path to the root directory.
- **patient_path** – Name of the subject.
- **slices** – Add additional slices cut to specific volumes
- **short** – Only output raw data, preprocessed data and FA data.

`elikopy.individual_subject_processing.white_mask_solo(folder_path, p, corr_gibbs=True, core_count=1, forceUsePowerMap=False, debug=False)`

Compute a white matter mask of the diffusion data for each patient based on T1 volumes or on diffusion data if T1 is not available. Otherwise, compute the whitematter mask based on an anisotropic power map.

Parameters

- **folder_path** – the path to the root directory.
- **p** – The name of the patient.
- **corr_gibbs** – Correct for gibbs oscillation.
- **core_count** – Number of allocated cpu cores.
- **forceUsePowerMap** – Force the use of an AnisotropicPower map for the white matter mask generation.
- **debug** – If true, additional intermediate output will be saved.

1.3. `elikopy.individual_subject_processing` module

ElikoPy

1.4 elikopy.utils module

`elikopy.utils.anonymise_nifti(rootdir, anonymize_json, rename)`

Anonymise all nifti present in rootdir by removing the PatientName and PatientBirthDate (only month and day) in the json and renaming the nifti file name to the PatientID.

Parameters

- **rootdir** – Folder containing all the nifti to anonymise.
- **anonymize_json** – If true, edit the json to remove the PatientName and replace the Patient-BirthDate by the year of birth.
- **rename** – If true, rename the nifti to the PatientID.

`elikopy.utils.export_files(folder_path, step, patient_list_m=None)`

Create an export folder in the root folder containing the results of step for each patient in a single folder

Parameters

- **folder_path** – root folder
- **step** – step to export
- **patient_list_m** – Define a subset a patient to process instead of all the available subjects.

example : `export_files('user/my_rootfolder', 'dMRI/microstructure/dti')`

`elikopy.utils.getJobsState(folder_path, job_list, step_name)`

Periodically check the status of all jobs in the job_list. When a job status change to complete or a failing state. Write the status in the log and remove the job from the job_list. This function end when all jobs are completed or failed.

Parameters

- **folder_path** – The path to the root dir of the study (used to write the logs.txt file)
- **job_list** – The list of job to check for state update
- **step_name** – The string value of the prefix to put in the log file

`elikopy.utils.get_job_state(job_id)`

Retrieve the state of a job through the sacct bash command offered by the lurm Workload Manager. :param job_id: The id of the job to retrieve the state of. :return state: The string value representing the state of the job.

`elikopy.utils.get_patient_list_by_types(folder_path, type=None)`

Print the list of patient corresponding to a specific type of patient.

Parameters

- **folder_path** – Path to the root folder of the study.
- **type** – The selected type

`elikopy.utils.inference(T1_path, b0_d_path, model, device)`

Parameters

- **T1_path** – Path to the normalized projected T1.
- **b0_d_path** – Path to the b0 atlases.
- **model** – DL Model
- **device** – Define if cuda or cpu is used.

Returns

`elikopy.utils.makedir(dir_path, log_path, log_prefix)`

Create a directory in the location specified by the `dir_path` and write the log in the `log_path`.

Parameters

- **dir_path** – The path to the directory to create.
- **log_path** – The path to the log file to write verbose data.
- **log_prefix** – The prefix to use in the log file.

`elikopy.utils.merge_all_reports(folder_path)`

Merge all subject's report into a single big report.

Parameters **folder_path** – Path to the root folder of the study.

`elikopy.utils.merge_all_specific_reports(folder_path, merge_wm_report=False, merge_legacy_report=False)`

Merge all selected specific subject's report into a single big report.

Parameters

- **folder_path** – Path to the root folder of the study.
- **merge_wm_report** – Select wm report.
- **merge_legacy_report** – Select legacy report.

`elikopy.utils.randomise_all(folder_path, randomise_numberofpermutation=5000, skeletonised=True, metrics_dic={'FA': 'dti', '_diamond_kappa': 'diamond', '_mf_fvf_tot': 'mf', '_noddi_odi': 'noddi'}, core_count=1, regionWiseMean=True)`

Perform tract base spatial statistics between the control data and case data and region wise stats. DTI needs to have been performed on the data first !!

Parameters

- **folder_path** – path to the root directory.
- **randomise_numberofpermutation** – Define the number of permutation used by randomize.
- **skeletonised** – If True, randomize will be using the only skeleton instead of the whole brain.
- **metrics_dic** – Dictionary containing multiple metrics. For each metrics, the metric name is the key and the folder metric's is the value.
- **core_count** – Number of allocated cpu core.
- **regionWiseMean** – If true, csv containing atlas-based region wise mean will be generated.

`elikopy.utils.regall(folder_path, grp1, grp2, core_count=1, metrics_dic={'_diamond_kappa': 'diamond', '_mf_fvf_tot': 'mf', '_noddi_odi': 'noddi'})`

register all diffusion metrics into a common space, skeletonised and non skeletonised using tract base spatial statistics (TBSS) between the control data and case data. DTI needs to have been performed on the data first.

Returns**Parameters**

- **folder_path** – path to the root directory.
- **grp1** – List of number corresponding to the type of the patients to put in the first group (control group). Used so data is in right order for statistics

1.4. elikopy.utils module

ElikoPy

- **grp2** – List of number corresponding to the type of the patients to put in the second group (case group). Used so data is in right order for statistics
- **metrics_dic** – Dictionnary containing the microstructural metric that will be registered.

`elikopy.utils.regall_FA(folder_path, grp1, grp2, starting_state=None, registration_type='-T', postreg_type='-S', prestats_treshold=0.2, core_count=1)`

register all diffusion metrics into a common space, skeletonised and non skeletonised using tract base spatial statistics (TBSS) between the control data and case data. DTI needs to have been performed on the data first.

Returns

Parameters

- **folder_path** – path to the root directory.
- **grp1** – List of number corresponding to the type of the patients to put in the first group (control group). Used so data is in right order for statistics
- **grp2** – List of number corresponding to the type of the patients to put in the second group (case group). Used so data is in right order for statistics
- **starting_state** – Manually set which step of TBSS to execute first. Could either be None, reg, post_reg, prestats.
- **registration_type** – Define the argument used by the tbss command `tbss_2_reg`. Could either be '-T', '-t' or '-n'. If '-T' is used, a FMRIB58_FA standard-space image is used. If '-t' is used, a custom image is used. If '-n' is used, every FA image is align to every other one, identify the “most representative” one, and use this as the target image.
- **postreg_type** – Define the argument used by the tbss command `tbss_3_postreg`. Could either be '-S' or '-T'. If you wish to use the FMRIB58_FA mean FA image and its derived skeleton, instead of the mean of your subjects in the study, use the '-T' option. Otherwise, use the '-S' option.
- **prestats_treshold** – Thresholds the mean FA skeleton image at the chosen threshold during prestats.

`elikopy.utils.submit_job(job_info)`

Submit a job to the Slurm Workload Manager using a crafted sbatch.

Parameters `job_info` – The parameters to use in the sbatch.

Return `job_id` The id of the submitted job.

`elikopy.utils.synb0DisCo(topuppath, patient_path, starting_step=None, topup=True, gpu=True)`
 synb0DISCO adapted from <https://github.com/MASILab/Synb0-DISCO>

Parameters

- **topuppath** – Path to the subject's topup folder.
- **patient_path** – Name of the subject.
- **starting_step** – Define the starting step, usefull if previous step had already been run.
- **topup** – If true, topup will be performed after synb0Disco.
- **gpu** – If true, torch will use the gpu.

Return type object

```
elikopy.utils.tbss_utils(folder_path, grp1, grp2, starting_state=None, last_state=None,  
                        registration_type='-T', postreg_type='-S', prestats_treshold=0.2,  
                        randomise_numberofpermutation=5000)
```

[Legacy] Perform tract base spatial statistics between the control data and case data. DTI needs to have been performed on the data first !!

Returns

Parameters

- **folder_path** – path to the root directory.
- **grp1** – List of number corresponding to the type of the patients to put in the first group (control group).
- **grp2** – List of number corresponding to the type of the patients to put in the second group (case group).
- **starting_state** – Manually set which step of TBSS to execute first. Could either be None, reg, post_reg, prestats, design or randomise.
- **last_state** – Manually set which step of TBSS to execute last. Could either be None, preproc, reg, post_reg, prestats, design or randomise.
- **registration_type** – Define the argument used by the tbss command tbss_2_reg. Could either be '-T', '-t' or '-n'. If '-T' is used, a FMRIB58_FA standard-space image is used. If '-t' is used, a custom image is used. If '-n' is used, every FA image is align to every other one, identify the “most representative” one, and use this as the target image.
- **postreg_type** – Define the argument used by the tbss command tbss_3_postreg. Could either be '-S' or '-T'. If you wish to use the FMRIB58_FA mean FA image and its derived skeleton, instead of the mean of your subjects in the study, use the '-T' option. Otherwise, use the '-S' option.
- **prestats_treshold** – Thresholds the mean FA skeleton image at the chosen threshold during prestats.

INDEX

A

`anonymise_nifti()` (in module *elikopy.utils*), 14

D

`diamond()` (*elikopy.core.Elikopy* method), 3
`diamond_solo()` (in module *elikopy.individual_subject_processing*), 11
`dicom_to_nifti()` (in module *elikopy.core*), 10
`dti()` (*elikopy.core.Elikopy* method), 3
`dti_solo()` (in module *elikopy.individual_subject_processing*), 11

E

elikopy
 module, 17
Elikopy (class in *elikopy.core*), 3
elikopy.core
 module, 3
elikopy.individual_subject_processing
 module, 11
elikopy.utils
 module, 14
`export()` (*elikopy.core.Elikopy* method), 4
`export_files()` (in module *elikopy.utils*), 14

F

`fingerprinting()` (*elikopy.core.Elikopy* method), 4

G

`get_job_state()` (in module *elikopy.utils*), 14
`get_patient_list_by_types()` (in module *elikopy.utils*), 14
`getJobsState()` (in module *elikopy.utils*), 14

I

`inference()` (in module *elikopy.utils*), 14

M

`makedirs()` (in module *elikopy.utils*), 15
`merge_all_reports()` (in module *elikopy.utils*), 15

`merge_all_specific_reports()` (in module *elikopy.utils*), 15
`mf_solo()` (in module *elikopy.individual_subject_processing*), 11
 module
 elikopy, 17
 elikopy.core, 3
 elikopy.individual_subject_processing, 11
 elikopy.utils, 14

N

`noddi()` (*elikopy.core.Elikopy* method), 5
`noddi_amico()` (*elikopy.core.Elikopy* method), 5
`noddi_amico_solo()` (in module *elikopy.individual_subject_processing*), 11
`noddi_fix_icvf_thresholding()`
 (*elikopy.core.Elikopy* method), 5
`noddi_solo()` (in module *elikopy.individual_subject_processing*), 11

P

`patient_list()` (*elikopy.core.Elikopy* method), 6
`patientlist_wrapper()` (*elikopy.core.Elikopy* method), 6
`preproc()` (*elikopy.core.Elikopy* method), 6
`preproc_solo()` (in module *elikopy.individual_subject_processing*), 11

R

`randomise_all()` (*elikopy.core.Elikopy* method), 7
`randomise_all()` (in module *elikopy.utils*), 15
`regall()` (*elikopy.core.Elikopy* method), 8
`regall()` (in module *elikopy.utils*), 15
`regall_FA()` (*elikopy.core.Elikopy* method), 8
`regall_FA()` (in module *elikopy.utils*), 16
`report_solo()` (in module *elikopy.individual_subject_processing*), 13

S

`submit_job()` (in module *elikopy.utils*), 16
`synb0DisCo()` (in module *elikopy.utils*), 16

ElikoPy

T

`tbss()` (*elikopy.core.Elikopy method*), 9

`tbss_utils()` (*in module elikopy.utils*), 16

W

`white_mask()` (*elikopy.core.Elikopy method*), 10

`white_mask_solo()` (*in module elikopy.individual_subject_processing*), 13

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
École polytechnique de Louvain

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/epl