IMPACT OF ACQUISITION PARAMETERS ON DIFFUSION MAGNETIC RESONANCE IMAGING QUALITY

Author: Marina Gómez Fernández
Director: Saúl Pascual-Díaz
Tutor: Alberto Prats Galino
ACKNOWLEDGMENTS

First and foremost, I would like to express all my gratitude to my director, Saül Pascual-Díaz, for guiding and helping me throughout the entire project’s development. But most of all, for making me feel, like the (quite comfortable) chair and the computer (which seemed to work in a different dimension where time went by much faster), became not only somehow mine, but also home for those few months. It has truly been a pleasure both working with you and learning from you.

Following the same line, I would also love to thank the entire IDIBAPS’ MRI platform team; this TFG could have not been conceived without any of their help and dedication. It has been my first experience working in a biomedical engineering related field and I could not have thought of a more enriching one; many thanks for giving me this opportunity.

Last but not least, I sincerely appreciate all those people who, by placing little grains of sand, made the greatest outcome possible, from my tutor Dr. Alberto Prats-Galino for allowing me to discover what’s behind the word neuroimaging; to the MRI technician that was there to pick me up (somehow holding back her laugh) the second I fainted after the scanner. In an in between, my family and friends for answering each one of all the desperately silly TFG-related questions I have had during those last weeks, and the ones that are yet to come when preparing the oral presentation. My deepest thanks, always.

Each one of you has made this final procedure to obtain the degree something worthwhile going through.
ABSTRACT

Acquisition parameters play a crucial role in Diffusion Tensor Imaging (DTI), having such an impact on white matter (WM) scalar measures values such as Fractional Anisotropy (FA), Signal to Noise Ratio (SNR) or even when it comes to whole brain tractography studies.

Among those acquisitions parameters we find the b-values (e.g. b-value is a factor that reflects the strength and timing of those gradients used to generate Diffusion-Weighted Images; the higher the b-value, the stronger the diffusion effects), voxel size (the smaller the voxel size the higher the quality) and diffusion directions (as the number of directions increases the acquisition time increases as well).

This project, places all of them of the focus of several clinical MRI studies performed on healthy subjects in the IDIBAPS’ MRI core facility. Those studies have been grouped into 5 datasets according to different acquisition protocols; in addition, 8 new acquisition proposal are first undergone and analysed.

Results show that as expected those parameters play a crucial role in image quality; however, each one of them has a different weight over those quality measurements. This will be discussed by splitting the results into datasets. Furthermore, the analysis performed on the protocol proposals confirm that they have reached what was their goal, to demonstrate that depending on the purpose of the scanner (emergency MRI, clinical MRI and research MRI) a specific acquisition sequence can be used.

KEYWORDS:

Magnetic Resonance Imaging; Diffusion Tensor Imaging; Diffusion-Weighted Imaging; Tractography; MRI Acquisition Parameters; Fractional Anisotropy; Signal to Noise Ratio; b-values; voxel size.
LIST OF FIGURES

Figure 1. Methodology flowchart
Figure 2. Only those spins whose Larmor frequency is equivalent to the frequency of the applied RF field will be excited. [11]
Figure 3. In each excitation, and MR signal is acquired as a function of time and recorded as a row of numbers in a data array (k-space). Reiteration of this process with phase-encoding gradients with a later Fourier transform gives the image in the right. [11]
Figure 4. a) MRI T1- weighted image. b) MRI T2- weighted image. c) T1, longitudinal relaxation time. d) T2, transverse relaxation time. [11]
Figure 5. RF pulses transmitted with a flip angle (FA). Gz, Gx and Gx signify the magnetic field gradients in the slice-select, phase-encoding and frequency-encoding directions respectively. [11]
Figure 6. Two RF are applied per TR. Data are acquired after a phase-encoding gradient pulse, in the presence of a frequency-encoding gradient. [11]
Figure 7. DW slices acquired different b-values. Left to right (mm/s²): 0, 1500, 3000. [12]
Figure 8. Anisotropic diffusion, in the ideal case of a coherently oriented tissue. [11]
Figure 9. Three examples of DT to illustrate differences in anisotropy. [11]
Figure 10. Scalar DTI maps of the different measures. [13]
Figure 11. Whole- brain streamline CSD tractography. Colors were assigned automatically according to an atlas-based tractography segmentation method.
Figure 12. FACT representation, the first ROI at voxels 1 and 2 and the second one at voxels 3 and 4, 2 continuous streamlines can be traced. [11]
Figure 13. Examples of complex fibers that can result in very similar FACT-based ellipsoids. [12]
Figure 14. Both tractographies using a DTI or CSD approach and voxels showing their behaviour in complex fiber regions. [14]
Figure 15. Pie chart showing the global MRI market share, by application, 2020 (%) [14]
Figure 16. Histogram showing Brain MRI related articles per year
Figure 17. Magnetic gradients distortion (dataset: ALBUCAT)
Figure 18. BIDS structure for dataset BIOMARCADORS
Figure 19. Step 1, brain extraction (ANTs).
Figure 20. Step 2, DWI registration (ANTs). [23]
Figure 21. Step 3, corrections (MRtrix3).[30]
Figure 22. Brain tractography. ROI: language region (CSD, Dataset: LAB_IMATGE)
Figure 23. SNR CC
Figure 24. FA map difference with the standard DTI and free water DTI
Figure 25. Atlas labels registered into the EPILENG dataset
Figure 26. SNR per each b-value interval and voxel size
Figure 27. WM atlas labels mean FA value per dataset
Figure 28. FA per each mask and dataset (2mm)
Figure 29. FA per each mask and dataset (1.5mm)
Figure 30. NOS per each algorithm, dataset and voxel size
Figure 31. Number of streamlines in accordance to length (CSD, DTI, 1.5mm and 2mm)
Figure 32. Number of streamlines in accordance to length (CSD, DTI, 1.5mm and 2mm). Zoomed in version
Figure 33. WBS
Figure 34. PERT chart
Figure 35. Pie chart, cost divided per specific activity.

LIST OF TABLES

Table 1. Solutions studied per each step developed throughout the project
Table 2. Acquisition sequence parameters per dataset
Table 3. PERT matrix
Table 4. GANTT chart
Table 5. SWOT
Table 6. Economic analysis
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AD</td>
<td>Axial Diffusivity</td>
</tr>
<tr>
<td>CSD</td>
<td>Constrained Spherical Deconvolution</td>
</tr>
<tr>
<td>CSF</td>
<td>Cerebrospinal Fluid</td>
</tr>
<tr>
<td>CC</td>
<td>Corpus Callosum</td>
</tr>
<tr>
<td>CT</td>
<td>Computerized Tomography</td>
</tr>
<tr>
<td>dMRI</td>
<td>Diffusion Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>DTI</td>
<td>Diffusion Tensor Imaging</td>
</tr>
<tr>
<td>DWI</td>
<td>Diffusion-Weighted Imaging</td>
</tr>
<tr>
<td>FA</td>
<td>Fractional Anisotropy</td>
</tr>
<tr>
<td>FACT</td>
<td>Fiber Assignment Through Continuous Tracking</td>
</tr>
<tr>
<td>fODF</td>
<td>fiber Orientation Distribution Function</td>
</tr>
<tr>
<td>GM</td>
<td>Grey Matter</td>
</tr>
<tr>
<td>HARDI</td>
<td>High Angular Resolution Diffusion Imaging</td>
</tr>
<tr>
<td>MD</td>
<td>Mean Diffusivity</td>
</tr>
<tr>
<td>MRI</td>
<td>Magnetic Resonance Imaging</td>
</tr>
<tr>
<td>NMR</td>
<td>Nuclear Magnetic Resonance</td>
</tr>
<tr>
<td>NOS</td>
<td>Number of Streamlines</td>
</tr>
<tr>
<td>RD</td>
<td>Radial Diffusivity</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>STD</td>
<td>Standard Deviation</td>
</tr>
<tr>
<td>WM</td>
<td>White Matter</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

Acknowledgments .................................................................................................................. 1  
Abstract ................................................................................................................................. 2  
List of figures ......................................................................................................................... 3  
List of tables ......................................................................................................................... 4  
List of abbreviations ............................................................................................................ 5  

1. INTRODUCTION ........................................................................................................... 8  
   A) MOTIVATION ............................................................................................................... 8  
   B) OBJECTIVES ............................................................................................................. 8  
   C) METHODOLOGY ....................................................................................................... 9  
   D) SCOPE AND SPAN ................................................................................................. 9  
   E) REGULATIONS AND LEGAL ASPECTS ............................................................... 10  

2. THEORETICAL BACKGROUND .................................................................................... 11  
   a) STATE OF THE ART .............................................................................................. 11  

3. MARKET ANALYSIS .................................................................................................. 21  
   a) TARGET SECTORS ............................................................................................... 21  
   b) HISTORICAL EVOLUTION .................................................................................. 22  
   c) FUTURE PERSPECTIVES .................................................................................... 23  

4. CONCEPTION ENGINEERING .................................................................................... 24  
   4.1. DATA POST-PROCESSING ............................................................................... 24  
   4.2. PROGRAMMING ENVIRONMENT ................................................................... 26  

5. DETAILED ENGINEERING .......................................................................................... 27  
   5.1. DATA ACQUISITION ......................................................................................... 27  
   5.2. DATA POST-PROCESSING .............................................................................. 29  
   5.3. PARAMETERS’ EXTRACTION ........................................................................... 31  
   5.4. STATISTICAL ANALYSIS ............................................................................... 34  
   5.5. RESULTS .......................................................................................................... 35  

6. EXECUTION SCHEDULE ............................................................................................... 41  
   a) WORK BREAKDOWN STRUCTURE (WBS) ...................................................... 41  
   b) DESCRIPTION OF THE TASKS ....................................................................... 41  
   c) PROGRAM EVOLUTION AND REVIEW TECHNIQUES (PERT) .................... 42  
   d) GANTT CHART .................................................................................................... 43  

7. TECHNICAL AND ECONOMICAL FEASIBILITY ....................................................... 44  
   a) TECHNICAL ........................................................................................................ 44  
   b) ECONOMICAL ....................................................................................................... 45
1. INTRODUCTION

a) MOTIVATION

A brain MRI can help doctors look for conditions such as bleeding, swelling, problems with the way brain developed, tumors, infections, inflammation, damage from an injury or a stroke among many others [1].

However, not all MRI scanners are performed in the same way, several acquisition sequences are available depending on both the reason and purpose. The reason why this is done is very simple, if a patient suffering from a possible stroke arrives and needs to be diagnosed what we want to do is a fast acquisition able to diagnose him; a quite similar situation happens for patients that cannot undergo longer scanner times (e.g. they suffer from a neurological disorder, children….) [2]. On the other hand, if the MRI scanner is scheduled for clinical purposes such as tumour diagnosis those acquisition parameters will be different, which will result in images with better quality and less noisy. The opposite case of the stroke patient happens for clinical research, cases where what we want to do is a diffusion study with a whole brain tractography. In here we will need the best acquisition parameters, which will result in a longer scanner.

What is expected up to the completion of this study is to propose several acquisition protocols, each one in accordance with the situations mentioned above, which will be first undergone by controls, analysed and once proven its effectiveness implemented in Hospital Clínic.

The implementation of those sequences gives us as well a major future advance, if in 10 years a specific study wants to be conducted, it will be necessary that all acquisitions are done following the same parameters. Otherwise we would not know for sure if those changes are due the subjects or the acquisition parameters themselves.

b) OBJECTIVES

The main objective of the following final degree project is to compare the most used MRI acquisition sequences during the last 5 years by performing a statistical analysis on several relevant parameters regarding image quality. Furthermore, eight sequences being developed in IDIBAPS will also be analysed.

The sentence above can be split down into more specifical secondary objectives:

I. To establish a relationship between the Signal to Noise Ratio (SNR) and b-value's acquisition, is there any optimal b-value?

II. To establish a relationship between the SNR and voxel's size acquisition.

III. To determine how the Fraction Anisotropy (FA) changes depending on the brain area (white matter, grey matter and cerebrospinal fluid). Does this value change when the acquisition’s sequence is different? Does this value change when the voxel size is different?
IV. To compare two tractography algorithms by displaying streamline's length of the same tractogram. Does this length change when the acquisition's sequence is different?

c) METHODOLOGY

The entire project has been developed under the framework of the Insititut D'Investigacions Biomèdiques Agust Pi i Sunyer (IDIBAPS) (https://idibaps.org) at Hospital Clinic. It is considered the leading biomedical research center in Spain that strives for excellence in biomedical research to improve people's health and ensure translational research. Its research is broken down into six different platforms, among them is found the magnetic resonance imaging platform. It works under the coordination of Emma Muñoz-Moreno and is the place in which I had a computer assigned and did the TFG under the supervision, guidance and help of my tutor, Saül Pascual- Díaz (Imaging scientist). In addition, the platform is equipped with an MRI scanner, the Siemens MAGNETOM Prisma 3T scanner for human clinical investigation.

On the other hand, it has been divided into four different stages in order to have a practical structure and successfully achieve the main goal of the project. The flowchart of the followed methodology is shown in Figure 1.

![Methodology flowchart](image)

Figure 1. Methodology flowchart

d) SCOPE AND SPAN

First of all, it has to be stated that this is a bachelor's final degree project with one main drawback above all, time. The duration is established deadlines, set for June 2022. This implies only being able to focus on particular aspects of the MRI post-processing such as parameters' extraction done by matrixial operations and their posterior statistical analysis. Since trying to encompass the entire post-processing would be way more complex and time-consuming than the span allows us, it has been left out of the scope of the actual TFG. However, given the relevance it has not only for the project understanding but also for its development it will be briefly explained in Section 5.

With this being said, the scope is in accordance with the accomplishment of its objectives and included in the following breakdown:

I. Bibliographic search for previous studies related to address a base-knowledge of the project's main technical issues such as dMRI acquisitions.

II. Performing MRI synthetic acquisitions by using the nowadays most used sequences.

III. Implementation of different algorithms and post-processing methods of the images acquired to establish which acquisition allows a better description of the synthetic model.
IV. Implementation of both the previous results and algorithms in MRI acquisitions.

V. Statistical analysis, comparison and discussion of the results obtained with previous studies.

The entire span of the project has been from March 2022 up to June 2022. Further details will be given in Section 6.

e) REGULATIONS AND LEGAL ASPECTS

This project (including data acquisition and posterior analysis) has been developed in the IDIBAPS’ MRI core facility. Meaning that all the legal requirements have to be in accordance with the Spanish regulations.

All data that has been analysed is from MRI core facility acquisitions, and has been anonymised, i.e. there is no feasible way to obtain personal information about the subject, following the *Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales*; this is more specifically stated in the health data treatment section in *Disposición adicional decimoséptima* [3].

MRI is both a medical device and a radiation-emitting electronic product. Following the World Health Organization, a medical device is: "Any instrument, apparatus, implement, machine, appliance, implant, reagent for in vitro use, software, material or another similar or related article, intended by the manufacturer to be used, alone or in combination for a medical purpose." [4]. All medical device regulations are stated in the European Commission Council Directive 93/42/EEC legislation, [5] there are four classes, ranging from low to high risk, MRI lie in Class II (moderate risk). This implies that several certifications need to be followed, such as ISO-13485 and ISO-14971, for quality management throughout its life cycle, and principles and processes for risk management respectively. [6]

On the other hand, regarding the radiation-emitting electronic products they are defined by the FDA as: 'Any electrically-powered product that can emit any form of radiation on the electromagnetic spectrum. These include a variety of medical and non-medical products'. [7] Regulations related to those products are stated by the European Commission Council Directive 2017/745/EC legislation. [5]. This implies the following standard, ISO 14630 for patients’ whom wearing an active implantable device safety [8].

Lattermost, during MRI diagnostic both individuals being scanned and those in the immediate vicinity of the equipment can be the exposed to a static magnetic field, a time-varying magnetic field gradients and radiofrequency. This implies that they must not possess any conductive, metallic nor magnetic material when under the influence. Furthermore, those gradient fields produce quite an important amount of acoustic noise, reaching an unacceptable and even dangerous level when it comes to prolonged exposition. [9] All those latter aspects have to be both explicitly warned by hospital’s signs and professionals’ indications.
2. THEORETICAL BACKGROUND

a) STATE OF THE ART

1. MAGNETIC RESONANCE IMAGING [10]

MRI is a non-invasive technology that uses non-ionizing electromagnetic radiation to generate cross-sectional images of interior structures to provide extensive, multi-parametric information on brain anatomy, function, and metabolism. MRI typically exploits Nuclear Magnetic Resonance (NMR), a phenomenon in which atomic nuclei subjected to a strong magnetic field absorb and reemit electromagnetic waves at a characteristic ‘resonant frequency’, which falls into the radiofrequency range.

Given the amount of information contained in the signal, several techniques have been tailored to augment factors of interest providing images of particular structures; including white matter (WM) tracts, lesions and arteries. Clinical applications of MRI are vast, encompassing neurological, psychiatric, cardiac, abdominal, musculoskeletal and vascular applications, making MRI one of the most powerful and flexible imaging tools.

1.1. MRI Image Acquisition: [11]

1.1.1. Image Formation: (resumir les fases de baix en una)

When nuclei in a certain slice or slab of tissue are excited, they generate a signal in which each point's contribution must be identified. This is accomplished by encoding spatial information into the signal's phase and frequency, which is accomplished through the use of magnetic field gradients.

1.1.1.1. Slice-Selective Excitation:

By stimulating nuclear magnetisation just in the slice of interest, imaging of specific slices of tissue can be obtained. This is accomplished by applying an RF pulse in the presence of a magnetic field gradient, causing the Larmor frequency to vary spatially. Only those spins whose Larmor frequency is equal to the applied RF field's frequency will be activated. The amplitude of the gradient and the bandwidth of the RF pulse determine the slice thickness.

1.1.1.2. Spatial Encoding:

A frequency encoding in one direction and a phase encoding in the perpendicular one is used to determine the position of the spins inside the imaging plane, as shown in Figure 2. During signal capture for frequency encoding, a magnetic field gradient is applied, which provides information about the position of the spins along the gradient's direction. Prior to image acquisition, a magnetic field gradient is introduced as a short pulse, forcing a phase shift among the spins that is imprinted on their signals during phase encoding. The entire process must be repeated numerous times to obtain position information from the phase.
1.1.1.3. Image Reconstruction:

A combination of frequency and phase encoding is required to generate a 2D image. To do so, the same slice of tissue must be excited multiple times, with the signal sampled as a function of time after each stimulation. The amplitude of the frequency-encoding gradient remains constant with each iteration, whereas the amplitude of the phase-encoding gradient increases with each repetition. The output is collected in a 2D array, and the signal's spatial distribution is recovered using the Fourier transform (Fig 3).

1.1.2. Image Contrast (resumir en 1 frase)

The relative signal appearance of particular tissues is determined by the sequence type, sequence parameters and tissue properties. Decay and recovery of the MR signal are depicted by several relaxation constants. The longitudinal relaxation time, T1, signifies the rate at which magnetization return to its equilibrium value, M₀, following RF excitation. The value of T1 differs depending on the tissue type; tissues with longer T1 appear hypointense on a T1-weighted image. The transverse relaxation time, T2, refers to the rate of signal decay following RF excitation; tissues with longer T2 appear hyperintense on T2-weighted images. (Fig 4)

Furthermore, TR and TE are also critical factors, with the first referring to the time between subsequent RF excitations and the second to the time between those RF excitations and signal capture. The relative contrast weighting of different tissues can be calculated by combining TE and TR; acquisitions with long TR and short TE produce proton-density-weighted images, in which the image signal is mostly dictated by the voxel hydrogen concentration.
1.1.3. Pulse Sequences: (resumir molt)

For explicit applications, the acquisition of an MR image necessitates optimal MR sequences. Sequences are a set of well-defined repetitive RF pulses and signal acquisition that must be coordinated with magnetic field gradients. The most frequent ones are known as gradient-echo and spin-echo sequences. This procedure is known as pulse sequence and can be changed to offer best signal contrast for a certain application.

1.1.3.1. Gradient-Echo Sequences:

During one of them, a single RF pulse is administered to each TR period individually (Fig 5). TE denotes the time between RF excitation and the gradient's echo center in this context. Short TR values are frequently used in gradient-echo sequences, resulting in T1-weighting. This image is also affected by the flip angle, with the highest value indicating more saturation; the value generally ranges from 10º to 40º. Tissues having a shorter T1 tend to be more hyperintense than those with a longer T1.

1.1.3.2. Spin-Echo Sequences:

Employs two RF pulses per TR period, while the initial excitation pulse has a flip angle of 90º the second one is of 180º (which required long TRs) (Fig 6). Data are collected during the spin echo once the spins have refocused. As T2 relaxation affects the amplitude of the spin echo, the resulting images are T2-weighted. The degree of the T2-weighting is dictated by the TE value.
1.1.4. Imaging Parameters:

There are a range of factors that enable the diversity of MR imaging for every particular pulse sequence, and these are the ones that investigators take advantage of. The flip angle and timing parameters (TR, TE) affect signal contrast, image resolution by spatial parameters, and total SNR by the number of signal averages. The size of the matrix and the number of pixels in the image establish the raw data dimensions. Each pixel represents a voxel, and the size of each voxel indicates the image’s resolution. The flip angle, acquisition spatial resolution, and timing parameters can all affect image SNR. Despite SNR can also be influenced independently of these parameters by altering the number of excitations or signal averages it increases the scan time. The scan time is proportional to the number of excitations, phase-encoding steps, and TR; by using a gradient-echo sequence with a short TR, this can be kept within reasonable limits.

2. DIFFUSION-WEIGHTED IMAGING

The Brownian motion of molecules coupled with their thermal energy is referred to as diffusion. Diffusion is constrained by the presence of cell membranes when tissue is intact; consequently, an increase in diffusivity indicates membrane disruption and is frequently used to detect lesions and degenerative processes.

Diffusion Weighted Imaging (DWI) is an MRI imaging technique that allows for the estimation of water molecules diffusivity inside tissue, providing information about the architecture of the brain’s WM and the direction of its fiber tracts. Molecular diffusion is a term that refers to the Brownian motion of particles caused by the thermal energy they carry, and can be described as follows: [10]

\[ < r^2 > = 6Dt \] (Eq 1).

where \( < r^2 > \) is the mean squared displacement, \( t \) the diffusion time and \( D \) de diffusion constant, which can be described as: [10]

\[ D = \mu K_B T \] (Eq 2).

For homogeneous materials, \( D \) is the same in every direction, although for highly structured materials and some biological tissues as WM, it becomes anisotropic and thus has different \( D \) coefficients along different directions. Because of the linkages discovered between a variety of neurological and neurosurgical disorders and WM alterations in recent years, this approach has sparked the scientific community’s interest.

2.1. DWI Acquisition Sequence

To obtain DWI acquisitions from an MRI machine it is needed to measure the spin-echo signal after applying a set of diffusion pulse gradients; the spin-echo signal of the particles that lost energy will be reduced during this procedure. As a result of having a degree of freedom in the gradient direction, the voxels with higher isotropy display higher intensity in the final volume.
Furthermore, those particles must diffuse for a given amount of time in order to characterize the diffusion of water in a medium. This quantity is known as the b-value in diffusion sequences, and it represents the time that passes per each mm$^2$. The strength of the observed signal for each voxel is determined not only by the diffusion restriction for that voxel, but also by the b-value chosen. However, despite obtaining a greater diffusion contrast with higher b-values we would be also obtaining a drop in the signal as can be seen in Figure 7.

The most used b-value is 1000 s/mm$^2$, as it offers a good signal/contrast relation. [12]

![Figure 7. DW slices acquired different b-values. Left to right (mm/s$^2$): 0, 1500, 3000.](image)

While DWI refers to the contrast of the acquired images, Diffusion Tensor Imaging (DTI) is characterised by the application of the Diffusion Tensor (DT) model, which enables the indirect quantification of the degree of anisotropy and structural orientation.


The most popular diffusion profile model is based on the Gaussian assumption, which allows the diffusion to be modelled with a single covariance matrix. This is known as diffusion tensor imaging (DTI); which is a sensitive probe of cellular structure that works by measuring the diffusion of water molecules.

What is being quantified is the diffusion coefficient (a proportionally constant that relates diffusive flux to a concentration gradient) in mm$^2$/s. When measuring diffusion through tissue, unlike in a glass of pure water, which is isotropic (all in the same direction), the directions fluctuate, making it anisotropic. This anisotropy is mostly created by cellular membranes in the WM of the brain, with some contributions from myelination and axon packing. Anisotropic diffusion can reveal the underlying tissue orientation, as shown in Figure 8, with the quickest diffusion measured parallel to the axons in a WM fiber tract. As a result of DTI analysis, interferences concerning features including diffusion directional preference (fractional anisotropy, mean diffusivity), diffusion rate along the primary axis, and transverse direction can be established within each voxel.

![Figure 8. Anisotropic diffusion, in the ideal case of a coherently oriented tissue.](image)
3.1. Diffusion Tensor image analysis

Magnetic field gradients are used to create an image that is sensitized to diffusion in a certain direction, which is then used to assess diffusion using MRI. A 3D diffusion model can be estimated by repeating the diffusion weighting process in several directions. As a result, extra gradient pulses are introduced, whose effect balances out for stationary water molecules, generating a random phase shift for diffusing molecules; this random phase causes a loss of signal from the diffusing molecules, resulting in darker voxels. This means that in the DWI for that direction, WM fiber tracts parallel to the gradient direction will appear darker.

The diffusion tensor \( D \) is then calculated by solving the Stejskal-Tanner equation (Eq 3) to compare the decreased signal \( S_k \) to the original signal \( S_0 \), which depicts how the signal intensity at each voxel drops in the presence of Gaussian diffusion.

\[
S_k = S_0 e^{-b g_k D g_k} \quad (Eq \ 3)
\]

The product \( g_k D \) represents the diffusion coefficient (diffusivity) in the \( g_k \) direction. This leaves us with an equations’ system that is solved for \( D \), the diffusion tensor (Eq 4).

\[
D = \begin{bmatrix}
D_{xx} & D_{xy} & D_{xz} \\
D_{yx} & D_{yy} & D_{yz} \\
D_{zx} & D_{zy} & D_{zz}
\end{bmatrix} \quad (Eq \ 4)
\]

DTI is usually displayed taking into account that the information contained in the tensor is usually condensed into a scalar number, or into four numbers equivalent to the R (red, x-axis, right to left), G (green, y-axis, anterior to posterior), B (blue, z-axis, head to feet) color, and a brightness value; and viewed by estimating the course of WM tracts through the brain via a process called tractography.

3.2. Diffusion Tensor Model

Peter Basser proposed the diffusion tensor (DT) for use in magnetic resonance imaging (MRI) in 1994. It was essential in proving the efficacy of diffusion MRI in describing the microstructure of white matter tissue and its biophysical properties.

DT is proportional to the covariance matrix of a three-dimensional Gaussian distribution that models the displacements of the molecules and describes the diffusion of water molecules using a Gaussian model. As a result, we're talking about a 3x3 positive-definite matrix with three orthogonal and mutually perpendicular eigenvectors \( (\epsilon_1, \epsilon_2, \epsilon_3) \) and three positive eigenvalues \( (\lambda_1, \lambda_2, \lambda_3) \) (Eq 5). The major DT’s eigenvector points towards the principal diffusion direction (i.e. the direction of the fastest diffusion); while the eigenvalues give the diffusivity in the direction of each eigenvector. Together, eigenvectors and eigenvalues define an ellipsoid that represents an isosurface of diffusion probability, the higher the anisotropy the more ellipsoid the shape gets as can be seen in Figure 9.
The fundamental disadvantage of DT is that there is only one main diffusion direction. As a result, in areas of the brain containing multiple fiber crossing each other, the tensor model may show that the primary diffusion direction is intermediate to these directions, causing inaccuracies in track definition. Other reconstruction approaches can, fortunately, be utilised to represent diffusion and fiber orientation in certain areas.

3.3.1. Diffusion Tensor measures [13]

In DTI, several quantitative measures can be defined from the eigenvalues of the tensor. The Axial Diffusivity (Eq 6) is defined as the diffusion coefficient along the principal direction of the fastest diffusion of the tensor; while the Radial Diffusivity (Eq 7) is defined as the average diffusivity perpendicularly to the principal diffusion direction. On the other hand, the Mean Diffusivity (Eq 8) is a quantitative map that describes the average amount of diffusion in a voxel, which is obtained by averaging the eigenvalues. (Fig 10)

\[
AD = \lambda_1 \quad (Eq \ 6)
\]

\[
RD = \frac{\lambda_2 + \lambda_3}{2} \quad (Eq \ 7)
\]

\[
MD = \frac{\lambda_1 + \lambda_2 + \lambda_3}{3} \quad (Eq \ 8)
\]

3.3.3.1.1. Fractional Anisotropy

The FA, which is used to quantify the degree to which the diffusion distribution in a voxel is directed, can be calculated using the DT's eigenvalues. That is, whether there is relatively unrestricted diffusion in one particular direction; it is also normalised and ranges from 0 (isotropic) to 1 (anisotropic). Furthermore, it is the most extensively used anisotropy measure, with the name attributed to the fact that it measures the anisotropic fraction of diffusion, which can be thought of as the difference between the form of the tensor ellipsoid and that of a perfect sphere (Eq 9).

\[
FA = \sqrt{\frac{1}{2} \left( \frac{(\lambda_1 - \lambda_2)^2 + (\lambda_1 - \lambda_3)^2 + (\lambda_2 - \lambda_3)^2}{\lambda_1^2 + \lambda_2^2 + \lambda_3^2} \right)} \quad (Eq \ 9)
\]

Figure 9. Three examples of DT to illustrate differences in anisotropy.

Figure 10. Scalar DTI maps of the different measures.
Nonetheless, while FA is frequently used to assess WM integrity, it should be interpreted with caution since, while it can indicate the density of fiber packing in a voxel and the quantity of myelin enveloping these axons, it is not always a measure of ‘tissue integrity.’ In other words, FA may be reduced in areas where white matter fibers are fanning out or where more than one population of white matter fibers crosses.

Alternatively, more complex models have been developed, such as the High Angular Resolution Diffusion Imaging (HARDI) model, which is capable of acquiring a larger number of DWI and thus better modelling of more convoluted fiber topologies.

4. TRACTOGRAPHY

The information obtained by the measures seen above allows us to obtain a set of diffusion-derived parameters very much useful in the development of microstructural biomarkers in the WM. However, those maps are not a direct representation of structural connectivity in the brain. That is when tractography takes the lead, which uses the diffusion information to achieve a representation of the neural pathway. Nonetheless, fibers are not by any means a representation of the axons, but a representation of regions in which water diffusion is favored.

The streamline tractography method is the most common approach; it creates discrete curves or trajectories, also known as tracts or fibers, as an output. And operates by successively stepping in the direction of the primary eigenvector (direction of the fastest diffusion). As a result, the eigenvectors are tangent to the generated trajectory. Each streamline shows the connectivity between two areas via the WM (or voxels), based on the path along which water molecules can diffuse more easily. They do not, however, include information regarding the connection’s orientation (i.e. it is not known if it goes from A to B, or B to A). (See Figure 11).

Expert knowledge or an automatic method are required to process DTI data to display fiber tracts of interest. After performing streamline tractography, the fiber trajectories of interest can be interactively selected using ‘virtual dissection,’ which involves defining inclusion/exclusion zones and using them to pick trajectories. There have also been created automated algorithms for atlas-based tractography segmentation that employ prior knowledge to determine trajectories. [10]
4.1. DTI tractography

One of the simplest tractography reconstruction methods is the Fiber Assignment Through Continuous Tracking (FACT) algorithm, which consists of connecting adjacent voxels through the direction obtained in turn from the first eigenvector ($\epsilon_1$).

The procedure is broken down into three stages: seeding, propagation, and termination. Seeding entails defining regions of interest and inserting one or more seeds in each of their voxels; the placement can be random or defined, and it commonly begins with a voxel located in the WM. Propagation (Fig 12) is where the fiber tracts are gradually generated, which can be performed with several algorithms besides FACT; finally, termination of the fiber tracking procedure is based on the termination criteria, which goal is to avoid propagation of the fibers in voxels where robustness of the vectorial field is not guaranteed. Common termination criteria include a minimum threshold for FA (ranging 0.1–0.3, which usually indicates that the fiber has propagated into the GM or cerebrospinal fluid) and turning angle threshold (ranging 40–70º). [11]

![FACT representation](image.jpg)

The tensor model, on the other hand, can only represent one primary fiber orientation in a voxel, which causes a difficulty in tractography when we reach a region of crossing fibers, as shown in Figure 13. And because those locations correspond to a large number of WM voxels in the brain with various fiber bundles oriented in different directions, the tensor model is unreliable for crossing, ‘kissing,’ and ‘fanning’ fiber bundles. Ambiguous configurations also cause a decline in the FA, forcing the reconstruction algorithm to stop propagating the streamline; in these instances, DTI reconstructions will be missing those specific structures. This can be partially solved by using higher-order models, which have been proposed to address regions with complex fiber configurations, such as Constrained Spherical Deconvolution (CSD).

![Examples of complex fibers](image2.jpg)
4.2. Constrained Spherical Deconvolution tractography

The CSD estimates the WM fiber Orientation Distribution Function (fODF) based on an estimate of the signal expected for a single-fiber WM population. This is used as the kernel in a deconvolution operation to extract a WM fODF from dMRI final measured within each voxel. In other words, it exploits the dependencies of the different macroscopic tissue types with the unique b-values to obtain a tissue-specific fODF.

To reconstruct anatomically constrained tractograms that better represent the structure of the WM in complex fiber configurations, CSD tractography algorithms use information calculated using spherical deconvolution methods in combination with anatomical information provided by a T1 sequence and different b-values.

Despite the advancements compared to the FACT algorithm, CSD has its own set of constraints. The first is that when it comes to fiber configurations, the number of diffusion gradient directions required grows exponentially with the complexity of the model, making it not feasible to collect sufficient information in all clinical circumstances. Figure 14 shows the difference when using DTI or CSD on the same voxel. [14]
3. MARKET ANALYSIS

The market sector contemplated in this project is the one related to MRI. Therefore, a deep analysis will be carried out throughout the following section.

a) TARGET SECTORS

The global MRI market size was valued at $6.7 billion in 2021 and expected to reach $11 billion by 2030, at a compound annual growth rate of 6.5%. It also accounted for the largest market share with 75.78% in 2020. Within those numbers, it has to be taken into account that the Covid-19 pandemic has had a negative impact in the MRI market, reporting an 87% reduction in outpatient imaging services over 2020 and early-mid 2021; this being equivalent to a decline of 6% in 2020 as compared to 2019. [15]

Main reason for the expected growth relies over the fact that MRI is an efficient diagnostic machine to identify diseases related to areas concerning blood vessels and brain; diseases that are not only increasing their prevalence (e.g. breast cancer, cardiovascular and neurological disorders), but also where early detection is critical and decisive of treatment procedure and life expectancy (e.g. osteoporosis, spinal infection and tumors). According to GLOBOCAN2020, 19.3M cases of cancer and 10M cancer deaths were estimated in 2020; this data compares to just over 17M cases and 9.5M deaths in 2018.

This is also supported by the advances in diagnostic techniques such as open MRI, visualization software and superconducting magnets; however, most of them are focused on the software, enabling faster contrast scans and simplifying imaging workflow as well.

In Figure 15 it is depicted each MRI application’s market share, and even though brain and neurological MRI hold almost 25% of the share (attributed to factors such as superior quality as compared to CT imaging), the spine and musculoskeletal one is rapidly increasing. This is because, unlike other imaging techniques, MRI’s spin-echo technique provides a detailed analysis on internal injuries on soft tissue and spinal cord, which is very much needed for emergency and trauma care units. [16]

Figure 15. Pie chart showing the global MRI market share, by application, 2020 (%)

- Brain and Neurological
- Spine and Musculoskeletal
- Vascular
- Abdominal
- Cardiac
- Breast
- Others
On the other hand, when it comes to the key companies in the MRI machine development sector, the market reflects a classical monopolistic scenario, where the established companies have a robust revenue position. We find the following key players leading the market:

- GE Healthcare
- Hitachi Medical Systems America AG.
- Siemens AG
- Toshiba Corporation
- Aurora Imaging Technologies, Inc.
- Koninklijke Philips N.V.
- Esaote SPA
- Sanrad Medical Systems Pvt. Ltd.

b) HISTORICAL EVOLUTION

In 1970, Raymond Damadian, discovered that abnormal cancer cells had longer relaxation time lengths than normal cells, this led him to realise the possibility of diagnosing diseases with a human-sized scanner. He called this imaging process ‘field-focused NMR’ and founded FONAR in 1978. [17] Other neuroimaging techniques like computerized tomography (CT) was first applied in 1961, and its discovery led to the invention of the X-ray CT scanner, which received the Nobel Prize for medicine in 1979.

MRI was first approved by the FDA in 1984, we are therefore talking about a field that has not been that long out in the market. Additionally, it did not grow until the early 1990s; until then there were less than 10 scanners per 1M Americans, since it was found not worth it and very expensive. This number spiked up to 27 by 2004 and 39 in 2015, where nearly $50 billion are spent per year [18].

All this data can be easily corroborated when searching ‘brain MRI’ in PubMed [19]. There are no articles until 1959, and it is not until 1982 when the exponential growth, and therefore the research starts. Figure 16 depicts all previous information.

![Figure 16. Histogram showing Brain MRI related articles per year. Source: PubMed](image)
As previously shortly stated, Covid-19 has had a negative impact in this market. The emergence of the pandemic forced all key players operating in the market to decrease their production of MRI equipment due to relocation of healthcare resources, declining patient visits, disruptions in supply chains among others. All those factors impacted highly on the sales, especially for the first half of 2020. However, MRI has been used as the first approach when it comes to detect Covid-19 infection, which triggered as well, an increase in their demand. For instance, GE Healthcare registered a decline of 13.8% in the total revenues generated in 2020 as compared to 2019. [15] Fortunately, as restrictions became more flexible (mid-end 2021) a significant increase in patient visits was seen, which resulted in a higher demand for equipment; this put the market back to the growth where it was right before the outbreak.

c) FUTURE PERSPECTIVES

It has clearly been seen all along the previous subsections that the MRI market is experiencing a growth that will only increase, and one of the main reasons why is because it is a young field in which loads of advances are still to be done. Among all the future perspectives that the market can experience we are going to go through the following ones:

- **High magnetic frequency MRI systems**: several universities are conducting research/studies on these systems. For instance, in University of Minnesota human MRI scanners have been performed using a 10.5 Tesla MRI. The MAGNETOM 10.5T was manufactured by Siemens and costed $15M but is expected to open new avenues in diagnosis of a wide range of diseases (e.g. Alzheimer, diabetes, cancer). Machines like that have more advantages associated than just the improvement in image clarity, such as integration with Artificial Intelligence. Further information can be found in the following article [20].

- **Open MRIs**: Loads of patients feel claustrophobic and disturbed inside an MRI scanner due to its loud noise and small space; these problems sometimes result in inaccurate results. That is why some of the market players are focusing on developing open MRI systems (which are also suitable for oversized patients). As an example, GE Healthcare’s Adventure MRI Series is specifically designed for pediatric imaging, this includes themed imaging rooms, lush visuals and hand-on activities to improve the experience. [21] Additionally, FONAR has developed the Upright MRI, which besides being suitable for claustrophobic and large patients it also allows to scan them in any position (sitting, standing, bending or lying down). This brings up few innovations, being able to image them in the positions in which they are experimenting their problems, and a more accurate diagnosis since the body is going to be imaged with their normal weight on the spine and other joints [22].

Despite close MRI still being the preferred one by radiologists, various clinical trials are being conducted to check the efficiency of the open one for the diagnosis of neonates’ diseases. In the following years, it is expected that those ongoing clinical trials and product approval of open MRI systems by organisms as the FDA increases.

However, this is all restrained by one big factor, the high cost associated with the acquisition, installation and proper care of these systems; which often requires complex infrastructure and associated costs that cannot be undertaken by many medical institutions.
4. CONCEPTION ENGINEERING

This section goes along the approaches that will be implemented in the TFG to achieve its objectives. All the solutions studied in order to compute the final data frames and posterior statistical analysis have been included in Table 1 (those in bold are the approaches finally taken). This will be described into further detail in the following sub-sections.

<table>
<thead>
<tr>
<th>SOLUTIONS STUDIED</th>
<th>T1W's brain extraction software</th>
<th>DWI registration software</th>
<th>dMRI correction software</th>
<th>Tractography software</th>
<th>Programming environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANT's BET2 FreeSurfer SPM12</td>
<td>ANT's FLIRT SPM12</td>
<td>Eddy MRtrix3 SMP12</td>
<td>MRtrix3 DTI Studio DSI Studio</td>
<td>Python Spyder MATLAB Octave RStudio</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Solutions studied per each step developed throughout the project

4.1. DATA POST-PROCESSING

Debating about why those acquisitions are done with one or another value regarding several parameters such as acquisition time, b-values, T1 or T2, goes beyond the scope of the TFG itself, therefore we are not going to take it into account for this Section. So, once images are downloaded from the MRI scanner, a post-process has to be done, the post-process done in question went through the following stages, for each stage a different software was used.

a) STUDY OF SOLUTIONS

4.2.1. T1-WMRI's BRAIN EXTRACTION:

Advanced Normalization Tools (ANTs) is a software package for normalizing data to a template. It includes a Neuroimaging Multimodality that computes high-dimensional mappings to capture the statistics of brain structure and function. Free and open source software. [23]

BET 2 (Brain Extraction Tool) is included inside the FSL software, a comprehensive library of analysis tools for MRI data. BET deletes non-brain tissue from a T1, T2 image of the whole head. Free and open source. [24]

FreeSurfer is a neuroimaging software for processing, analysing and visualisation of MR images. This includes skull stripping, B1 bias field correction, GM and WM segmentation, reconstruction of cortical surface models, labeling of regions on the cortical surface, nonlinear registration of the cortical surface and diffusion tractography toolboxes among many others. Free and open source software. [25]

SPM12 (Statistical Parametric Mapping) is a software package that has been designed for the analysis of brain imaging data sequences. It is designed to work with MATLAB. MATLAB's student license is of $99/year. [26]

4.2.2. DWI REGISTRATION:
FLIRT (fMRIB’s Linear Image Registration Tool) is a fully automated robust and
accurate tool for linear (affine) intra and inter modal brain image registration. Based around a multi-
start, multi-resolution global optimization method. [27] [28]

ANTS (explained in Section 4.2.1.)

SPM 12 (explained in Section 4.2.1.)

4.2.3. dMRI CORRECTION:
Eddy is a tool to correct for eddy current-induced distortions and signal dropouts. It also
performs outlier detection to identify slices where signal has been lost as a consequence of subject
movement during the diffusion encoding. It is included inside the FSL software; free and open
source software. [29]

SPM 12 (explained in Section 4.2.1.)

MRtrix 3 provides a large suite of tools for image processing, analysis and visualisation, with a
focus on the analysis of WM using DW MRI. Features include the estimation of fiber orientation
distributions using CSD, a probabilistic streamlines algorithm for fiber tractography of WM and a
comprehensive visualisation tools in mrview among many others. Free and open source software.
[30]

4.2.4. TRACTOGRAPHY:
DSI Studio is a tractography software tool that maps brain connections and correlates
findings with neuropsychological disorders. It is a collective implementation of several diffusion MRI
methods, including DTI, diffusion MRI connectometry, and generalized deterministic fiber tracking.
Free software. [31]

DTI Studio is a diffusion tensor image processing program running under Windows. It is suitable
for such tasks as tensor calculation, color mapping, fiber tracking, and 3D visualization. Free
software. [32]

MRtrix 3 (explained in Section 4.2.3)

b) PROPOSED SOLUTION

Once all software/toolboxes have been thoroughly reviewed by comparing all their
characteristics and applying them to the objectives of this study, the following solutions have been
selected.

For T1-WMRI’s brain extraction the software selected has been ANTs. The selection has been
partially based on the computational time, while BET2 is the fastest option, it is also quite imprecise
on several aspects; the opposite happens with FreeSurfer (it can take up to 7 hours). ANTs stands
in the middle, taking a reasonable time and giving us enough precision regarding our needs. SPM
12 is taken out of the options since a MATLAB license is required.

For DWI registration the case is quite similar as above, SMP12 requires a license and FLIRT
directly does not work as well as ANTs does, this latter one being a way more developed and
documented.

For dMRI correction, the toolbox selected has been MRtrix3 when working with the dMRI correction,
since it allows the removal of several aspects as motion, SNR or Gibbs ringing artefacts. For the
Eddy current-induced distortions Eddy FSL (note the redundancy) is the best tool. Once again SPM12 requires a license.

Ultimately, for tractography MRtrix3 is the software being selected, since it includes the tckgen command, this being the best way regarding our objectives to calculate each tractogram for different tractography algorithms.

4.2. PROGRAMMING ENVIRONMENT

In order to group all processed data, perform mathematical operations to obtain new data, extract relevant parameters, collect them into a data frame to later on perform an analysis different software can be used.

a) STUDY OF SOLUTIONS

MATLAB is a programming platform, with a matrix-based language designed specifically for engineers and scientists to analyse and design systems and products. It presents different toolboxes allowing the analysis of data, image processing, visualization and many others. The Student license is $99/year. [33]

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics for a general purpose and with a flexible environment, that is that makes it one of the most popular programming languages. Furthermore, it includes modules and packages such as NumPy, Matplotlib or SciPy. Free and open source software. [34]

Spyder is a scientific environment written in Python, for Python. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality allowing data exploration, interactive execution, deep inspection and humongous visualization capabilities. Free and open source software. [35]

RStudio uses the programming environment of R, which is specially focused on the data science environment, its statistical analysis and graphical representation. It includes a big number of R packages for data science, machine intelligence and interoperability between Python and R among others. Free and open source software. [36]

b) PROPOSED SOLUTION

MATLAB offers several image processing toolboxes that can be used for image data segmentation, extraction and analysis and are seen to work faster than Python ones, and that the price is not an impediment at all since UB owns a license. However, the packages/ tools needed to perform those operations (e.g. DIPY, pandas, nibabel) have been developed for Python, and even though DIPY allows to read/ save in MATLAB to later perform the statistical analysis, the quite reasonable approach to develop the entire script in the same programming environment was taken. The statistical and graphical representation pro that RStudio has, is covered by using the seaborn library. This being a Python data visualization library based on matplotlib yet more comfortable in handling Pandas data frames (which will be highly used throughout the project’s development). Finally, between Python and Spyder we are sticking to this latter one. Main reasons being the fact that you can work with several consoles at the same time, allowing code to running in one, different scripts without mixing variables among others.
5. DETAILED ENGINEERING

All along this section, as the name says by itself, a detailed explanation of the projects, the different steps, as well as their methodology will be conducted. As is foreseeable, this section will follow the course that the TFG’s development did by itself.

As mentioned beforehand (Section 1), this project comprises the comparison of some of the most used MRI acquisition sentences by performing a statistical analysis regarding several parameters that we have considered to be relevant when it comes to image quality. Into this comparison two sequences (meant to be way superior both in image quality and acquisition time) that are being developed by researchers working for the MRI platform IDIBAPS are as well included.

The methodology followed has used a population of 114 controls (a total of 135 acquisitions), classified into 6 datasets. All of them have undergone an MRI scanner at the Hospital Clinic up to May 2022. Previously to the MRI scanner a protocol has to be designed in order to establish how the acquisitions will be done, how long each one will last etc. Once the images are downloaded into the IDIBAPS computers they have to undergo all the post-processing, this consisting on the brain extraction (brain, WM, GM, and CSF mask) atlas registration into the DWI, posterior correction on several aspects (e.g. subject motion, Eddy induced- currents, SNR, Gibbs or Gibbs ringing artifacts), and tractography. This being done, it will result in several files containing all corrected information.

Once all the files are ready, it is time to upload them into the programming environment, Spyder, by first creating a dataframe in order to group all files into several lists, by performing iterations on those lists all calculations (DIPY, MRtrix3…) will be done. Relevant parameters’ values will then be extracted and stored into another dataframe, which will be saved as a csv file. Finally, those csv will be uploaded into another script in which the statistical analysis and representations are performed.

5.1. DATA ACQUISITION

As previously stated, data has been separated into 6 different groups depending on which project they were acquired for, among them we find:

I. **EPILENG** (Epilèpsia i Llenguatge): comprises 21 subjects.

II. **EOP**: comprises 80 subjects.

III. **ALBUCAT**: comprises 2 subjects. For each subject two acquisitions were done, named AP (antero-posterior) and PA (postero-anterior), a total of 4 subjects. This is done to cancel out the distortion seen in Figure 17, which appears when the magnetic gradients first enter in contact with the skull. And by acquiring those images when emitting gradients both front-to-back and back-to-front, to later superpose them, we end up with a ‘neutral’ image.

IV. **BBHSA**: comprises 5 subjects. As for ALBUCAT two acquisitions are done for subject; therefore, we are going to consider it as if we had 10 subjects.

V. **BIOMARCADORES**: comprises 4 subjects.
VI. **LAB_IMATGE**: Comprises 2 subjects. This dataset is the last one that has been developed for IDIBAPS; this means two things. First, it is a brand new protocol proposal, designed to obtain as much information as possible without considering the acquisition time (which went up to 1.30h); and second, as this TFG had to be delivered by June 8th only two acquisitions were done and processed by then (Saül’s one and mine). As just said, for each subject 8 acquisitions were done; repeating each proposal twice (a total of 4) but changing the voxel size. Therefore, a total of 16 of ‘subjects’ were considered.

Furthermore, given the amount of data we will work with, it is highly needed to establish a consensus on how to organise and share data obtained. Lack of consensus leads to misunderstandings and wasted time on renaming data or rewriting scripts. Therefore, all files have been named in accordance with Brain Imaging Data Structure (BIDS) [37], each subject has their own folder; inside the folder two other ones are found `anat` and `dwi`. Being the latter one the one we are going to use throughout all project’s development, and where the corrected files are going to be saved into. For a better understanding, Figure 18 shows a screenshot of the files.

![Figure 17. Magnetic gradients distortion (dataset: ALBUCAT)](image)

The MRI scanner used for all acquisitions is the MAGNETOM Prisma fit 3T developed by Siemens Healthineers for clinical imaging and includes the following features: [38]

- XR gradient system: (80/200), with E11c software
- Head/neck coil 20 channels 440 mm × 330 mm × 370 mm
- Cap coil 32 channels 300 mm × 390 mm × 315 mm
- Head/neck coil 64 channels 435 mm × 395 mm × 350 mm
- Body coil 18 channels
- Column coil 32 channels
- VisuaStimDigital
- Presentation® licensed visual and auditory stimuli presentation system
- Screen to project visual stimuli for the 32-channel head coil
- Remote control to respond to fMRI studies

Each acquisition sequence has been done in accordance with the following parameters (Table 2)
5.2. DATA POST-PROCESSING

5.2.1. T1-WMRI's BRAIN EXTRACTION:

First step includes the skull-stripping of the anatomical image (i.e. the structural acquisition, T1-W) to extract the brain itself, this is done by applying extraction masks; those masks allow to segment the input data into CSF, WM and GM by using reference atlas. Among the outputs generated, we find the tissue segmented, and the brain extraction mask (excluding cerebellum and brainstem). The brain mask provides a reference for matching other modalities, in our case DWI (Figure 19). This has been done with the ANTs software [23].

<table>
<thead>
<tr>
<th>DATASET</th>
<th>Voxel size (isometric)</th>
<th>Field of View (FOV)</th>
<th>Diffusion directions</th>
<th>b-values (s/mm²)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIOMARCADORES</td>
<td>2 mm</td>
<td>122 x 122 x 60</td>
<td>31</td>
<td>0 (x3) 1000 (x6)</td>
</tr>
<tr>
<td>EOP</td>
<td>2 mm</td>
<td>128 x 128 x 60</td>
<td>31</td>
<td>0 (x7) 800 (x30)</td>
</tr>
<tr>
<td>EPILENG</td>
<td>1.5 mm</td>
<td>140 x 140 x 92</td>
<td>100</td>
<td>0 (x7) 1500 (x47) 3000 (x46)</td>
</tr>
<tr>
<td>ALBUCAT</td>
<td>1.5 mm</td>
<td>140 x 140 x 92</td>
<td>100</td>
<td>0 (x7) 1500 (x47) 3000 (x46)</td>
</tr>
<tr>
<td>BBHSA</td>
<td>1.5 mm</td>
<td>150 x 150 x100</td>
<td>47</td>
<td>0 (x3) 1000 (x6) 2000 (x12) 3000 (x26)</td>
</tr>
<tr>
<td>Proposal 1</td>
<td>1.5 mm</td>
<td>150 x 150 x100</td>
<td>78</td>
<td>500 (x6) 1000 (x64)</td>
</tr>
<tr>
<td>(fastnfurious)</td>
<td>2 mm</td>
<td>150 x 150 x100</td>
<td>78</td>
<td>500 (x6) 1000 (x64)</td>
</tr>
<tr>
<td>Proposal 2</td>
<td>1.5 mm</td>
<td>112 x 112 x 76</td>
<td>165</td>
<td>0 (x17) 500 (x6) 1000 (x32) 1500 (x48) 2000 (x88)</td>
</tr>
<tr>
<td>(mortalkombat)</td>
<td>2 mm</td>
<td>112 x 112 x 76</td>
<td>165</td>
<td>0 (x17) 500 (x6) 1000 (x32) 1500 (x48)</td>
</tr>
<tr>
<td>Proposal 3</td>
<td>1.5 mm</td>
<td>150 x 150 x100</td>
<td>149</td>
<td>500 (x6) 1000 (x64) 2000 (x64)</td>
</tr>
<tr>
<td>(diehard)</td>
<td>2 mm</td>
<td>150 x 150 x100</td>
<td>149</td>
<td>500 (x6) 1000 (x64) 2000 (x64)</td>
</tr>
<tr>
<td>Proposal 4</td>
<td>1.5 mm</td>
<td>150 x 150 x100</td>
<td>165</td>
<td>0 (x17) 500 (x6) 1000 (x32) 2000 (x48) 3000 (x64)</td>
</tr>
<tr>
<td>(sharknado)</td>
<td>2 mm</td>
<td>150 x 150 x100</td>
<td>165</td>
<td>0 (x17) 500 (x6) 1000 (x32) 2000 (x48) 3000 (x64)</td>
</tr>
</tbody>
</table>

Table 2. Acquisition sequence parameters per dataset
5.2.2. DWI REGISTRATION:

T1w and atlas are registered into the DWI in order to have both a structure and reference of where specific regions are located (this allows us to later extract particular regions). Exemplified in Figure 20. Done with the ANTs software [23].

5.2.3. dMRI CORRECTION:

This is done with several functions inside the MRtrix 3 package. \texttt{mrdegibbs} function, which results in an image where the Gibbs ringing artifacts are removed (Figure 21), and \texttt{dwidenoise}, which does a noise level estimation (caused by motion, SNR…) and returns a denoised image. Last one used is \texttt{dwibiascorrect}, and as the name says, it performs a field inhomogeneity correction for a DWI volume series and removes the estimated bias field. [30]
5.2.4. TRACTOGRAPHY

Last step inside the post-processing stage is to compute the tractograms with both algorithms, DTI and CSD. The software MRtrix 3 has the tckgen, where the number of tracts generated has to be specified (50,000) and the tracking algorithm that is going to be used. [30] Once tractograms are computed, the whole brain tractography can be visualized (an in Figure 11); moreover, by applying the atlas specific regions can be extracted, Figure 22 shows the language region with the CSD algorithm.

5.3. PARAMETERS' EXTRACTION

At this point of the project, all needed files have already been acquired by performing the post-processing. Three Spyder scripts were created for developing this subsection:

5.3.1. SNR:

Computing the Signal-to-Noise-Ratio (SNR) of DW images is still an open question, as it depends on the WM structure as well as the gradient direction corresponding to each DWI. SNR is defined as the ratio of the signal’s mean to the underlying Gaussian noise’s standard deviation. \[ SNR = \frac{\text{mean (signal)}}{\text{std (noise)}}, \text{Eq 10} \]. Where std (noise) is computed from the background of any DW image. To compute mean (signal), we are using the ‘worst-case’ approach following the DIPY documentation [39]. Where several WM structures such as the corpus callosum (CC) can be
easily identified from the colored-FA map; since they are mainly oriented in one direction, for the CC specifical case it would appear in red (left-right direction).

First step is to compute the DT model in the brain mask. This brain mask does not need to be post-processed, so we are going to directly upload the brain mask done on the T2-weighted images (T1W2). In a mathematical view, the brain mask is a matrix of 0s and 1s, 1 meaning that we are in an area where there is a brain. The DT model is computed by doing a gradient table containing both b-values and b-vectors (they give information about which way the diffusion gradients were pointing at). By using the tenmodel.fit function we end up with the DT fitted into the brain mask.

Next, a threshold is set to isolate the red voxels of the colored-FA map, meaning we are only obtaining information about the CC registered into the brain mask with the DT already fitted into it. The CC can be seen in Figure 23.

Once we have isolated the voxels in which the CC is, the mean (signal) in the region is estimated. Std (noise) is then calculated by inverting the brain mask and saving only the 0s (are outside the brain). Finally [Eq 10] is applied for the b-values intervals we had previously established.

The inputs of this first script have been:
1. DWI native b-values (file.bval)
2. DWI native b-vectors (file.bvec)
3. T1W2 native brain mask (file_T1w2diffnat_brain_mask.nii.gz)

The output of this script has been:
1. Dataframe containing the SNR ratio for each interval of b-values per each MRI acquisition.
   Moreover, the voxel size has also been added for posterior purposes.

Both script and dataframe can be found in Section 11 (Annex), named as 01_SNR.py and 01_SNRs.csv respectively.

5.3.2. FA:

As seen throughout Section 1 (point 3.3.3.1) the DT model is a simple way to characterize diffusion anisotropy. Nonetheless, regions near the cerebral ventricle and parenchyma can be underestimated by partial volume effects of the CSF causing a free water contamination; which can particularly affect DTI analysis for different subjects (and therefore a different brain morphology) [40]. If this were to happen, a statistical analysis of the FA for all 135 acquisitions grouped by datasets would not be accurate enough. This difference can be seen in Figure 24.
To remove this free water contamination, the DTI model can be expanded [Eq 11] to take into account an extra compartment representing the contributions of free water diffusion.

\[ S(g, b) = S_0(1 - f) e^{-bg \cdot D g} + S_0 f e^{-b D_{iso}} \quad [Eq \, 11] \]

Where \( g \) and \( b \) are diffusion gradient directed and weighted, \( S(g, b) \) is the DW signal measured, \( S_0 \) the signal in a measurement with no DW, \( D \) the diffusion tensor, \( f \) the volume fraction of the free water component, and \( D_{iso} \) the isotropic value of the free water diffusion.

Once explained the approach taken by the DIPY library, it is time to upload all needed files, as for SNR b-values and b-vectors are needed (native file, no post-process is done), to compute the gradient table and eventually the DT model. The case is not the same for the brain mask, since it is better to use the already corrected one in order to make all maps more comparable among themselves. As before the next step is to compute the tensor fit model with the `tenmodel.fit` function. Once done it is possible to extract the FA from it. FA is a matrix the same shape as the brain mask where numbers go from 0 to 1 in accordance with anisotropy.

Upcoming step is to multiply the FA by the WM, GM and CSF brain mask in order to obtain the specifical FA for each brain area. As for the normal brain mask the WM, GM and CSF ones are a matrix of same dimensions compounded by 0s and 1s. For each one of them the mean value and standard deviation (std, in order to measure the values’ spread) are computed.

Finally, the JHU WM labels atlas [41] is uploaded, in there 50 WM tract labels were created by hand segmentation of a standard-space average of diffusion MRI tensor maps. Figure 25 shows the atlas correlated into one of our acquisitions. All 50 labels have been entered into a Spyder dictionary (atlas_labels); by registering the file were each label is registered into a certain brain voxel into the FA matrix, we obtain the FA value for a certain number of voxels, which likewise correspond to a specific WM region defined in atlas_labels. As for the masks, both mean and std a computer and stored into the dataframe.

The inputs of the second script have been:

i. DWI native b-values (`file.bval`)
ii. DWI native b-vectors (`file.bvec`)
iii. T1W2 corrected brain mask (`file_MNI2diff_brainmask.nii.gz`)
iv. WM brain mask (`file_T1w2diff_wm_mask.nii.gz`)
v. GM brain mask (`file_T1w2diff_gm_mask.nii.gz`)
vi. CSF brain mask (`file_T1w2diff_csf_mask.nii.gz`)
vii. Atlas data (`file_JHU2diff_labels.nii.gz`)

The output of this script has been:

ii. Dataframe containing the FA value (mean and std) for each kind of brain mask (simple, GM, WM, and CSF), and atlas label per each MRI acquisition. Moreover, the voxel size has also been added for posterior purposes.

Both script and dataframe can be found in Section 11 (Annex), named as 02_FA.py and 02_FAmaps.csv respectively.
5.3.3. TRACTOGRAPHY:

Local fiber tracking is an approach used to model WM by creating streamlines from local directional information. Following the DIPY documentation [42], if the local directionality of a tract segment is known, one can integrate along those directions to build a complete representation of that structure. At this point, local fiber tracking has already been performed (subsection 5.2.4) using two of the algorithms explained in Section 1 (4.1 and 4.2), DTI and CSD; tractograms are then directly uploaded into the script. For each tractogram streamlines are going to be selected and stored into a new variable. Next step is to retrieve the length of each streamline by using the \texttt{dipy.tracking.streamline} function; this will be stored into two arrays depending on which algorithm is used (DTI or CSD) next to the total number of measures inside each of the arrays.

The inputs of the third script have been:

i. Tractogram using DTI algorithm (\texttt{file_tractogram_DTI_50kseeds.tck})
ii. Tractogram using CSD algorithm (\texttt{file_tractogram_CSD_50kseeds.tck})

The output of this script has been:

i. Dataframe containing two arrays giving the length of each streamline computed in the tractogram (one per algorithm) and the length of both arrays per acquisition (which corresponds to the number of streamlines).

Both script and dataframe can be found in Section 11 (Annex), named as 03\_tractogram\_py and 03\_tractogram\_data.pkl respectively.

5.4. STATISTICAL ANALYSIS

Once all the relevant parameters have already been extracted and stored into three different dataframes; the statistical analysis needs to be performed. This will be done by creating another Spyder script in which the dataframes are going to be uploaded and plotted as considered.

First step after uploading the dataframes is to group all acquisitions into the 6 datasets established in Section 5.1; so, all plots can be classified into them allowing a better analysis.

For the SNR dataframe we will extract the SNR values per each interval established under the voxel size condition. By simply looking at the dataframe it is seen that the voxel size (\texttt{xdim}, \texttt{ydim} and \texttt{zdim}) only takes to values, 1.5 and 2mm and is the same for all the dimensions; therefore, the analysis is going to be based on two boxplots, one per each voxel size.

When it comes to the FA dataframe three plots are going to be extracted. First and second one are 6 different sub-boxplots, representing both mean FA value and std FA value for all the masks already mentioned in the FA script: GM, WM and CSG. As for the SNR, the first plot is done for the 1.5mm voxel size and the second one for the 2mm. Each box represents a different dataframe. For a better analysis each pair (std and mean) of subplots have been set into equal y-axis scale, causing some boxes to look flattened.

Third plot represents the mean FA value for all the WM regions established in the atlas labels; as there are 50 regions the best way to visualise it was through a heatmap (i.e. map where values are turned into a color code). It has as many columns as WM regions (x-axis) and rows as datasets (y-axis). Voxel size has not been taken into consideration.
Finally, the **tractography streamlines length analysis** has been done by creating a set of histograms for each dataset (one color per dataset). Four histograms are displayed, one per algorithm approach followed (DTI and CSD) per each voxel size (1.5 and 2mm). In addition, a boxplot correlating the Number of Streamlines (NOS) per each algorithm, dataset and voxel size has also been created.

The inputs of the fourth and last script have been:
- 01_SNRs.csv
- 02_FAmaps.csv
- 03_tractogram_data.pkl

The outputs of this script have been:
- Fig 26 (SNR)
- Fig 27 (FA, 1.5mm)
- Fig 28 (FA, 2.5 mm)
- Fig 29 (FA, heatmap)
- Fig 31 (Histogram streamlines length)
- Fig 32 (Streamlines length zoomed in)
- Fig 30 (NOS)

Script can be found in Section 11 (Annex), named as 04_statistics.py and figures are deployed in Section 5.5.

### 5.5. RESULTS

The results accomplished are presented in this section, along with their corresponding discussion.

#### 5.5.1. SNR:

In Figure 26, it is seen the SNR value of the DWI per each b-value, left plot for a voxel size of 1.5mm and right plot for 2mm. The b-value interval has been chosen in accordance with the b-values used for each acquisition; in a way in which each dataset had as much representation as possible inside one interval. Main reason to do so, is because each dataset has different acquisition parameters (as seen in Section 5.1); therefore, there is no way to know for sure if the difference is due to the b-value or the parameters themselves.

For visualisation purposes each interval comprises a range of 400 values (the first one is not included). The obtained results suggested that as the b-value [s/mm$^2$] increases the SNR decreases, decreasing as well the power to obtain a better specification coming from the useful information (the signal). This decrease in the overall SNR is also seen when decreasing the voxel size (a smaller voxel size corresponds to a higher resolution).

At this point what is essential is to find a balance between the b-values and SNR; since taking only short b-values means that we are not letting enough time for the diffusion to happen, which at itself reduces quality. As mentioned in Section 1 the b-value that has been find to give best performance is around 1000 s/mm$^2$, however this should not always be generalized, but applied for each particular case; meaning that if we are in a situation where the patient is critical we will try to perform the shortest possible acquisition (increasing voxel size and considering short b-values).
Nonetheless, SNR does not hold that much value for itself as it does to consider other aspects as FA. So, we cannot just conclude that the bigger the voxel size the better because this will have a significant impact on the FA, as will be seen in the analysis performed below.

SNR per each b-value interval and voxel size

5.5.2. FA:

Figure 27 shows a heatmap, y axis represents each dataset and x axis specific WM brain areas. First view it can clearly be seen the difference among the old datasets and the new acquisition protocol designed by IDIBAPS; this latter one shows highest FA values, reflecting WM structures will also be better reconstructed, visualised and analysed since the fiber density will be higher.

On the other hand, voxel size does not appear to have a relevant role, this is in fact seen when looking at the slightest color variation among all new acquisitions paired in 1.5 mm and 2 mm voxel size. What appears to have a relevant role, is the number of directions taken in each acquisition (this parameter is displayed in Table 2), since those showing a higher number also have a greater FA value. So, it can be concluded that the best acquisition when it comes to specific WM structures reconstruction is the one used for DIEHARD (LAB_IMATGE dataset), and the worst one by far ALBUCAT.

However, this is a hypothesis and in order to establish a completely accurate conclusion several analyses have to be done for each individual structure.

Figure 29 deploys six sub-boxpots in which the FA value of each individual GM, WM and CSF brain mask is shown per dataset with a 2mm voxel size. Figure 28 shows the same information but for a 1.5 mm voxel size. It is easily seen how for the GM and CSF brain mask the FA values are quite low (i.e. there is almost no diffusivity in those particular regions).

When analysing Figure 28, it is clearly seen that the highest the number of diffusion directions the highest the FA value. Proposal 4 (Sharknado) has the greatest result.

As the voxel size increases, the FA decreases; we are analysing a larger region so it will be more difficult to detect the level of organization or anisotropy of certain structures. For this latter case (voxel size of 2 mm) all acquisitions show a similar FA value, since independently of the number of
directions or b-values the voxel size is enough to reduce FA by itself. A clear result to which discuss the hypothesis stated in Section 5.5.2, where saying that the SNR is higher is not enough to use a 2mm voxel.

Figure 27. WM atlas labels mean FA value per dataset

Figure 28. FA per each mask and dataset (1.5mm)
In Figure 30 it is seen the Number of Streamlines (NOS) that each algorithm (DTI and CSD) is capable to reconstruct; once again separated per voxel size and dataset. The NOS value decreases as the voxel size increases; meaning that it is best seen for a smaller voxel if a streamline located in a complex region actually corresponds to several of them. On the other hand,
when it is known that the DTI is the simplest tractography algorithm, makes sense that there is not much NOS difference among the worst and best dataset no matter what the voxel size is.

However, when moving towards a better algorithm as CSD is, results are way different. Once again, the smallest the voxel size the higher the NOS. Considering 1.5mm voxel size as the best approach, Proposal 4 (Sharknado) has the best performance.

Nonetheless, the NOS value is such a general concept that can also be affected by many other aspects, results above are just a hypothesis; in order to establish a completely accurate conclusion on why our 1.5 mm proposal works the best in combination with CSD, a whole brain tractography study has to be conducted so the streamlines can be individually analysed. This goes beyond the scope of this TFG.

Last figures (Figure 31) consist of a combination of superposed histograms (one per dataset) deploying on the x axis the length of the streamlines, and y axis the Number of Streamlines that accomplish the latter condition. Different histograms have been done per voxel size and tractography algorithm. By taking a superficial look it is seen that the NOS value per dataset highly differ for short streamlines, this being the main reason why Figure 32 has been done (zoomed in only for streamlines up to 100mm). Once again, the DTI algorithm shows a way more heterogeneous histogram, since its simplicity does not allow him to distinguish between datasets with better parameters.

Figure 31. Number of streamlines in accordance to length (CSD, DTI, 1.5mm and 2mm)
Figure 32. Number of streamlines in accordance to length (CSD, DTI, 1.5mm and 2mm). Zoomed in version
6. EXECUTION SCHEDULE

The following section, as referenced to in the previous ones, includes the phases and milestones that the project has gone through in the course of its development. It includes the timings and the structure followed to achieve the goals in the established timing; additionally, it defines as well the set of activities carried out and the time needed for their implementation.

a) WORK BREAKDOWN STRUCTURE (WBS)

The Work Breakdown Structure (WBS) is a visual, hierarchical and deliverable-oriented deconstruction of a project. It is a very helpful diagram since it allows you to break down the project into its scope and visualize all the tasks required to complete it. [43]

The WBS for this project (Figure 33) is structured into four main tasks: previous study and training, project development, parameters’ analysis and project wrap up. In addition, it consists of three levels, the first one enclosing the tasks previously mentioned, the second one

![WBS Diagram](Image)

Figure 33. WBS

b) DESCRIPTION OF THE TASKS

Tasks and subtasks from the WBS are briefly defined hereunder.

Task 1 – Previous study and training: firsts readings

1.1. Background study: Analyze previous works related to this field as well as its current situation.

1.2. Define objectives: Establish the aim of the project as well as the most important goals to be achieved by the end of the project.

1.2.1. Bibliographic review: gathering useful information for the project’s development.

1.3. Software training: research and practical training on the MRI field’s most used libraries for the programming environment used, in the projects’ case Spyder, and why it is the best one following our approach.
Task 2 – Project development: steps previous to the results’ acquisition.

2.1. Data post-processing: T1W brain extraction, DWI register, DWI correction and tractography.

2.2. Script elaboration: Once the images have been processed into data (e.g. matrix) using Spyder, a script needs to be built in order to first load the subject's data, extract the important parameters and values, and turn them into relevant information according to the project’s aim.

2.3. Market analysis: search for information on the sector to which our application is directed as well as its future potential taking into account legal limitations/ regulations when applied.

Task 3 – Parameters’ analysis: results’ acquisition and statistical analysis.

3.1. Data running and parameters’ acquisition: using the previous scripts we have to run the processed data in order to obtain the data frames we will later work with.

3.2. Data deployment and analysis: Those data frames need to be read by other scripts that will display the results in the desirable way (e.g. plots) in order to be able to do a statistical analysis.

Task 4 – Project wrap up: Last tasks, closure.

4.1. Drawing conclusions: Once results have been obtained and analysed we can draw conclusions and determine whether those objectives have been accomplished as initially thought or not.

4.2. Memory elaboration: State in a document all knowledge acquired during the process as well as procedure, approaches, results and the previous conclusion.

4.3. Oral presentation preparation: Determine the key points and highlights about the project and consequent bibliographic context to prepare a clear, short and concise presentation. Visual support material was prepared for this task.

c) PROGRAM EVOLUTION AND REVIEW TECHNIQUES (PERT)

A PERT chart is a visual management tool used to map out and track the tasks and timelines. The main advantage it holds over GANTT charts (studied in the next subsection) is that it indicates dependencies, which gives knowledge of how one missed deadline could affect other tasks. [44]

<table>
<thead>
<tr>
<th>Task</th>
<th>Activity</th>
<th>Description</th>
<th>Predecessor activities</th>
<th>Consequent activities</th>
<th>Expected time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A</td>
<td>Background study</td>
<td>-</td>
<td>B, C</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>B</td>
<td>Define objectives</td>
<td>A</td>
<td>F</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>C</td>
<td>Software training</td>
<td>A</td>
<td>D, E</td>
<td>10</td>
</tr>
<tr>
<td>2.1</td>
<td>D</td>
<td>Data post-processing</td>
<td>C</td>
<td>G</td>
<td>50</td>
</tr>
<tr>
<td>2.2</td>
<td>E</td>
<td>Script elaboration</td>
<td>C</td>
<td>G, H</td>
<td>50</td>
</tr>
<tr>
<td>2.3</td>
<td>F</td>
<td>Market analysis</td>
<td>B</td>
<td>H, J</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>G</td>
<td>Data running and parameters’ extraction</td>
<td>D, E</td>
<td>H</td>
<td>50</td>
</tr>
<tr>
<td>3.2</td>
<td>H</td>
<td>Data deployment and analysis</td>
<td>E, F, G</td>
<td>I, J</td>
<td>50</td>
</tr>
<tr>
<td>4.1</td>
<td>I</td>
<td>Drawing conclusions</td>
<td>H</td>
<td>J</td>
<td>10</td>
</tr>
<tr>
<td>4.2</td>
<td>J</td>
<td>Memory elaboration</td>
<td>F, H, I</td>
<td>K</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>K</td>
<td>Oral presentation preparation</td>
<td>I</td>
<td>-</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 3. PERT matrix
The projects’ PERT chart has been elaborated from a task matrix (Table 3) where the activities defined in the WBS are assigned a letter, duration and dependencies with previous ones are established. It has to be taken into account, that according to the UB Teaching Plan, the TFG is equivalent to approximately 300h.

From this last table, the PERT chart was built (as Figure 34 shows). In each node, two main values can be found, in the bottom-left the Early time (i.e. the minimum time needed to reach a node) and in the bottom-right the Late Time (i.e. the maximum time the task needs to be completed without delaying the project). In addition, green arrows indicate the Critical Path, this being the minimum necessary time to carry out the project.

![PERT Chart](image)

**d) GANTT CHART**

And last but not least, the Generalized Activity Normalization Time Table (GANTT) displays the tasks against time. On the left of the chart the tasks are shown, and along the top there is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the tasks (Table 4).

![GANTT Chart](image)

<table>
<thead>
<tr>
<th>TASK</th>
<th>March (in weeks)</th>
<th>April</th>
<th>May</th>
<th>June</th>
</tr>
</thead>
<tbody>
<tr>
<td>Background study</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Define objectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Software training</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data post-processing</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Script elaboration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Market analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data running and parameters’ extraction</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data deployment and analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drawing conclusions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Memory elaboration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oral presentation preparation</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Table 4. GANTT chart*
7. TECHNICAL AND ECONOMICAL FEASIBILITY

a) TECHNICAL

The technical feasibility corresponds to a SWOT (strengths, weaknesses, opportunities and threats) analysis. A strategic planning technique that studies and detailing both internal (SW) and external (OT) factors that might affect the realization of the project, both in a positive or negative way. Mentioned information is depicted in Table 5.

In order to take as much advantage as possible of the opportunities as well as avoid those threats, the confluence of strengths and opportunities and of weaknesses and threats must be studied. The internal strengths have to be focused into fitting the opportunities.

In between the strengths that have been analysed, we find that the data that has been used corresponds to a temporal period going up to 2022; heterogenous data since new developments have been done and protocols designed. Additionally, it has all been acquired and post-processed in the IDIBAPS’ MRI in Hospital Clinic. There are a variety of opportunities that this project could benefit from, mainly focused on possible improvements in the MRI acquisitions, perfect examples are the new protocols that have been used as proposals. Even more opportunities can come with the introduction of the 7T MRI scanner, and the fact that the market is experiencing an exponential growth causing the same growth on related articles published.

Threats are external and therefore unavoidable. However, this does not mean that strategies can be thought to control them and actuate as soon as possible so their consequences have the smallest impact possible.

Even though the project presents a wide number of strengths, it also has important weaknesses to consider. First of all, computation times related to post-processing take an incredibly high value, limiting even more the available time to perform it. Moreover, as each dataset has their own acquisition parameters it is impossible to know if some variations are due to only those parameters that we are considering, or due to the impact of secondary ones. Finally, the new acquisition proposals only have two samples per each one, making hard to distinguish personal variation form the one caused by acquisition parameters.

Last but not least, the principal threat this project is facing is that with so many technological advances, and the emergence of new technologies, this study could end up being neglected. Two main reasons are, the elevated cost of performing not only the MRI scanner but also the post-processing, and the implementation of the 7T MRI.
### Positive factors

**STRENGTHS**

- Data are controls acquired in Hospital Clinic by IDIBAPS
- High motivation and dynamism
- MRI scanner in which to perform acquisitions (3T).
- Guidance and support
- Possibility to access to data from previous studies done by IDIBAPS

### Negative factors

**WEAKNESSES**

- Slow computation time
- Limited time
- Only two samples per each acquisition proposal
- Different acquisition parameters make it hard to compare them

### Opportunites

- 7T MRI imaging
- Improvements on MRI acquisitions
- Exponential growth on both market size and related articles

### Threats

- Cost MRI
- Implementation of 7T can leave this study neglected

Table 5. SWOT

#### b) ECONOMICAL

The following subsections comprises all theoretical costs of this project and consequent study. The cost of the project has been divided into two main sections, while the first one comprises the cost of what has been the TFG by itself; the second one comprises several studies performed by the platform from which I was able to take the data and apply it into my project. This division is somehow crucial since the prices’ range is so much different. For both cases, IDIBAPS has provided the funds for covering all expenses, this includes human resources, facilities and required equipment (software/ hardware). Overall, it does not take a rocket scientist to realise that the project’s entirety was technically feasible by virtue of IDIBAPS. All expenses, together with further details have been listed in Table 6.

Starting with **human resources**, the salary which has been taken into account when it comes to the student is 15 €/hour, an average salary in Spain for junior biomedical engineers. Since the total amount of hours needed to complete the project has been stated to be 300h (Section 6) the cost of that person is 4,500€. This total cost has been specifically divided taking into account two main stages: **educational and developing stage** (e.g. bibliographic research, background’s analysis, conduction of the study) which took about 160 h. And **writing and editing stage** (memory elaboration and oral presentation preparation), counting approx. 140h.

Keeping on with the **software**, the ones used only when considering the TFG has been **Spyder**, an open-source cross-platform integrated development environment for scientific programming in the Python language. Additionally, **FSLeyes** was also used for complementary visualization, this
consisted on an open-source image viewer for 3D and 4D data. As both names say by themselves, the cost was of 0 €. Detailed information was given in Section 4.

The hardware only consisted of a regular desktop computer located in the IDIBAPS’ MRI lab, which was estimated to cost approx. 800 € but it is hard to know since it is as well quite old.

Moving on towards the part where the substantial cost weight is, the studies conducted by IDIBAPS and from which all data to develop the project was taken from consisted of 114 subjects (as stated in Section 5). Each of those subjects underwent an MRI scanner, the price for each MRI acquisition is stipulated to 153.5 € per hour (including preparation, acquisition, medical report and technical support if needed) since each subject was in there 1h, the total cost goes up to 17,499 €*. Later on, those images need to be post-processed, taking into account that this is a Diffusion MRI study in which human brain structural connectomics have to be analysed the price just for setting up the equipment is of 1,485 €*. The post-processing itself is 39.38 € per image analysed, considering all 114 images, the cost is of 4,490.46 €*.

All those prices are of public knowledge in IDIBAPS Scientific Platform Fees 2022 document [45]; taken into account considering that we are internal workers in IDIBAPS (fees for workers in Campus Clinic or external personal are also included) and with the IVA yet to be applied (*).

The total cost of the study is 28,774,46 €. The cost divided by specific activity is depicted in Figure 16, where it can clearly be seen that the second section is the one that holds the majority of the weight. This being the main reason why Figure 35 shows the costs only taking into account the TFG section. One last consideration has to be taken into account, as the TFG is mandatory and therefore all human resources are tasks that have to be done by the student, no actual cost is associated.

![Figure 35. Pie chart, cost divided per specific activity.](image)
<table>
<thead>
<tr>
<th>ITEM</th>
<th>COST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Human Resources</strong></td>
<td></td>
</tr>
<tr>
<td>Educational and Developmental Stage</td>
<td>2.400 €</td>
</tr>
<tr>
<td>Writing Stage</td>
<td>2.100 €</td>
</tr>
<tr>
<td>Total Cost Student</td>
<td>4.500 €</td>
</tr>
<tr>
<td><strong>Software</strong></td>
<td></td>
</tr>
<tr>
<td>Spyder</td>
<td>open source 0€</td>
</tr>
<tr>
<td>FSLeys</td>
<td>open source 0€</td>
</tr>
<tr>
<td><strong>Hardware</strong></td>
<td></td>
</tr>
<tr>
<td>Desktop computer</td>
<td>800 €</td>
</tr>
<tr>
<td><strong>MRI acquisition</strong></td>
<td></td>
</tr>
<tr>
<td>Acquisition 3T: preparation,</td>
<td></td>
</tr>
<tr>
<td>acquisition, medical report and</td>
<td></td>
</tr>
<tr>
<td>technical support</td>
<td>153.50 €/ hour</td>
</tr>
<tr>
<td>**Image post-processing and</td>
<td></td>
</tr>
<tr>
<td>statistical analysis**</td>
<td></td>
</tr>
<tr>
<td>Diffusion MRI - Set up</td>
<td>1,485€ *</td>
</tr>
<tr>
<td>Diffusion MRI - Structural</td>
<td></td>
</tr>
<tr>
<td>Connectomics (human brain fee)</td>
<td>39.38 € (fee per image)</td>
</tr>
<tr>
<td></td>
<td>4,490.46 € *</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL COST</strong></td>
<td>28,774,46 €</td>
</tr>
</tbody>
</table>

* IVA not applied

Table 6. Economic analysis
8. CONCLUSIONS AND FUTURE PERSPECTIVES

To conclude, this project had the objective of comparing several popular MRI acquisitions designed in the last 5 years by means of different acquisition parameters, among them, and with a new acquisition that has just been designed by IDIBAPS. To achieve the previous goal a list of previous steps needed to be done, since in order to extract those parameters, both a post-processing stage and several data frames creation was essential. For each post-processing stage, a description of tasks and software/toolboxes used has been done. Same way, for each data frame created, a description of the Spyder libraries and toolboxes has also been performed, as well as the main steps followed to compute/extract those parameters.

Taking into account the whole population of the project, there have been 114 control subjects (i.e. healthy), which turned out into a total of 135 acquisitions who had undergone an MRI scanner up to May 2022. All 135 acquisitions have been grouped into 6 datasets according to the acquisition protocol.

Just by taking a shallow look into the plots it can easily be seen that for each parameter analysed the values undergo such a big variety when split into datasets. Therefore, it can be said that yes, it is possible to not only compare those acquisitions but also to determine which ones have the best performance. And as it has already been done with the new acquisition, use those results to design new proposals which will combine as much of the best acquisition parameters as possible; but most importantly to make further evidence in why those new proposals work way better and should be implemented into the MRI scanners in Hospital Clinic. However, also taking into account up to which step we want to take the diffusion study, we will not use the same protocol with someone who is suffering a stroke or a healthy person undergoing a whole brain tractography study as a control.

With regard to the limitations that have appeared along the development of the project, the major drawback has been time, emphasizing on the fact that this a TFG which has to be handed in before a deadline. Moreover, the computational time limitation could partially be solved by running scripts in a different computer with a bigger capacity; the same approach, was taken when doing the data post-process.

This being said, this project does not end here, several future perspectives have been taken into consideration. In order to obtain substantial evidences on why our protocol proposal is quite a good candidate to be implemented, further control subjects have to undergo it until we reach a substantial number. Scripts should then be re-runed to obtain updated dataframes to perform further statistical analyses. To be even way more accurate this new analysis should also include a deepest sight into the FA values for the WM specific structures (i.e. further analysis on Figure 27), and a whole-brain tractography to specifically evaluate the relationship between the CSD and DTI algorithms, voxel size and other acquisition parameters conforming a dataset integrity.
9. REFERENCES


[40] DIPY : Docs 1.5.0 - Using the free water elimination model to remove DTI free water contamination. (n.d.). Retrieved June 8, 2022, from https://dipy.org/documentation/1.5.0/examples_built/reconst_fwdti/#example-reconst-fwdti

10. ANNEXES
### Loading packages

```python
import os
import pandas as pd
import numpy as np
import nibabel as nib
from dipy.core.gradients import gradient_table
from dipy.io.gradients import read_bvals_bvecs
from scipy.ndimage import binary_dilation
from dipy.segment.mask import bounding_box
```

```python
warnings.simplefilter(action='ignore', category=FutureWarning)
```

### main_dir

```python
main_dir = '/home/marina/TFG/DATA'
output_file = '/home/marina/TFG/outputs/01_SNRs.csv'
```

```python
try:
df = pd.read_csv(output_file)
except:
df = pd.DataFrame(columns=['SubjID'])
```

```python
s_list = []
for s in sorted(os.listdir(main_dir)):
s_wd = f'{main_dir}/{s}'
if os.path.isfile(f'{s_wd}/dwi/{s}_DWI_native.nii.gz'):
s_list.append(f'{s_wd}/dwi/{s}_DWI_native.nii.gz')
elif os.path.isfile(f'{s_wd}/dwi/{s}_DWI_AP_native.nii.gz'):
s_list.append(f'{s_wd}/dwi/{s}_DWI_AP_native.nii.gz')
s_list.append(f'{s_wd}/dwi/{s}_DWI_PA_native.nii.gz')
```

```python
error = []
for f_path in s_list:
s = os.path.basename(f_path).split('.')[0].split('DWI')[0][:-1]
f_name = os.path.basename(f_path).split('.')[0]
s_wd = f'{main_dir}/{s}'
if f_name in df['SubjID'].values: continue
print(f'{s},{f_name}')
```

```python
raw_data = nib.load(f_path)
data = raw_data.get_fdata()
affine = raw_data.affine
```

```python
bvals, bvecs = read_bvals_bvecs(f'{f_path[:-7]}.bval', f'{f_path[:-7]}.bvec')
gtab = gradient_table(bvals, bvecs)
```

```python
if '_AP_' in f_name: mask_path = f'{s_wd}/dwi/{s}_T1w2diffnat_AP_brain_mask.nii.gz'
elif '_PA_' in f_name: mask_path = f'{s_wd}/dwi/{s}_T1w2diffnat_PA_brain_mask.nii.gz'
```

```python
...```
else: mask_path = f'{s_wd}/dwi/{s}_T1w2diffnat_brain_mask.nii.gz'

mask = np.squeeze(nib.load(mask_path).get_fdata())

threshold = (0.6, 1, 0, 0.1, 0, 0.1)
CC_box = np.zeros_like(data[..., 0])

mins, maxs = bounding_box(mask)
mins = np.array(mins)
maxs = np.array(maxs)
diff = (maxs - mins) // 4
bounds_min = mins + diff
bounds_max = maxs - diff

CC_box[bounds_min[0]:bounds_max[0],
       bounds_min[1]:bounds_max[1],
       bounds_min[2]:bounds_max[2]] = 1

mean_signal = np.mean(data[mask > 0], axis=0)
mask_noise = binary_dilation(mask, iterations=10)
mask_noise[..., :mask_noise.shape[-1]//2] = 1
mask_noise = ~mask_noise

noise_std = np.std(data[mask_noise, :])

#bvals
bvals400=mean_signal[np.where(bvals<=400)]/noise_std
bvals800=mean_signal[np.where((bvals>400)&(bvals<=800))]/noise_std
bvals1200=mean_signal[np.where((bvals>800)&(bvals<=1200))]/noise_std
bvals1600=mean_signal[np.where((bvals>1200)&(bvals<=1600))]/noise_std
bvals2000=mean_signal[np.where((bvals>1600)&(bvals<=2000))]/noise_std
bvals2400=mean_signal[np.where((bvals>2000)&(bvals<=2400))]/noise_std
bvals2800=mean_signal[np.where((bvals>2400)&(bvals<=2800))]/noise_std
bvals3200=mean_signal[np.where((bvals>2800)&(bvals<=3200))]/noise_std

s_dict = { 'SubjID': f_name, 
            'bvals [0,400]': np.mean(bvals400) ,
            'bvals (400,800]': np.mean(bvals800),
            'bvals (800,1200]': np.mean(bvals1200),
            'bvals (1200,1600]': np.mean(bvals1600),
            'bvals (1600,2000]': np.mean(bvals2000),
            'bvals (2000,2400]': np.mean(bvals2400),
            'bvals (2400,2800]': np.mean(bvals2800),
            'bvals (2800,3200]': np.mean(bvals3200),
            'xdim': raw_data.header.get_zooms()[0],
            'ydim': raw_data.header.get_zooms()[1],
            'zdim': raw_data.header.get_zooms()[2]}

df = df.append(s_dict, ignore_index=True)
df.set_index('SubjID', inplace = True)
df.to_csv(output_file)
<table>
<thead>
<tr>
<th>SubjID</th>
<th>bvals (1600,2000)</th>
<th>bvals (2000,2400)</th>
<th>bvals (2800,3200)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.610130167689274</td>
<td>4.959810994427331</td>
<td>3.500000000000000</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>4.931137033363760</td>
<td>5.272099832493482</td>
<td>0.909090909090909</td>
<td>8.012500000000000</td>
</tr>
<tr>
<td>20.427065682702374</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>0.17624785432618995</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
<tr>
<td>2.0</td>
<td>2.0</td>
<td>2.0</td>
<td>0.17624785432618995</td>
</tr>
</tbody>
</table>
# 02_FA

```python
# Loading packages
import os
import pandas as pd
import numpy as np
import nibabel as nib
from dipy.core.gradients import gradient_table
from dipy.io.gradients import read_bvals_bvecs
from dipy.reconst.dti import TensorModel
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

# Atlas labels
atlas_labels = {1: 'Middle cerebellar ped.'}
atlas_labels[2] = 'Pontine crossing tract'
atlas_labels[3] = 'Genu of CC'
atlas_labels[4] = 'Body of CC'
atlas_labels[5] = 'Splenium of CC'
atlas_labels[6] = 'Fornix'
atlas_labels[7] = 'Corticospinal tract R'
atlas_labels[8] = 'Corticospinal tract L'
atlas_labels[9] = 'Med. lemniscus R'
atlas_labels[10] = 'Med. lemniscus L'
atlas_labels[12] = 'Inf. cerebellar ped. L'
atlas_labels[13] = 'Sup. cerebellar ped. R'
atlas_labels[14] = 'Sup. cerebellar ped. L'
atlas_labels[15] = 'Cerebral ped. R'
atlas_labels[16] = 'Cerebral ped. L'
atlas_labels[17] = 'Ant. limb of int. capsule R'
atlas_labels[18] = 'Ant. limb of int. capsule L'
atlas_labels[19] = 'Posterior limb of int. capsule R'
atlas_labels[20] = 'Posterior limb of int. capsule L'
atlas_labels[21] = 'Retrolenticular part of int. capsule R'
atlas_labels[22] = 'Retrolenticular part of int. capsule L'
atlas_labels[23] = 'Ant. corona radiata R'
atlas_labels[24] = 'Ant. corona radiata L'
atlas_labels[25] = 'Sup. corona radiata R'
atlas_labels[26] = 'Sup. corona radiata L'
atlas_labels[27] = 'Posterior corona radiata R'
atlas_labels[28] = 'Posterior corona radiata L'
atlas_labels[29] = 'Posterior thalamic radiation R'
atlas_labels[30] = 'Posterior thalamic radiation L'
atlas_labels[31] = 'Sag. stratum R'
atlas_labels[32] = 'Sag. stratum L'
atlas_labels[33] = 'Ext. capsule R'
atlas_labels[34] = 'Ext. capsule L'
atlas_labels[35] = 'Cingulate gyrus R'
atlas_labels[36] = 'Cingulate gyrus L'
```

atlas_labels[37] = 'Cingulum (hippocampus) R'
atlas_labels[38] = 'Cingulum (hippocampus) L'
atlas_labels[39] = 'Fornix (cres) / Stria terminalis R'
atlas_labels[40] = 'Fornix (cres) / Stria terminalis L'
atlas_labels[41] = 'Sup. longitudinal fasc. R'
atlas_labels[42] = 'Sup. longitudinal fasc. L'
atlas_labels[43] = 'Sup. fronto-occipital fasc. R'
atlas_labels[44] = 'Sup. fronto-occipital fasc. L'
atlas_labels[45] = 'Inf. fronto-occipital fasc. R'
atlas_labels[46] = 'Inf. fronto-occipital fasc. L'
atlas_labels[47] = 'Uncinate fasc. R'
atlas_labels[48] = 'Uncinate fasc. L'
atlas_labels[49] = 'Tapetum R'
atlas_labels[50] = 'Tapetum L'

# main_dir="/home/marina/TFG/DATA"
output_file = '/home/marina/TFG/outputs/02_FAmaps.csv'

try:
    df = pd.read_csv(output_file)
except:
    df = pd.DataFrame(columns = ['SubjID'])

s_list = []
for s in sorted(os.listdir(main_dir)):
    s_wd = f'{main_dir}/{s}'
    if os.path.isfile(f'{s_wd}/dwi/{s}_DWI_corr.nii.gz'):
        s_list.append(f'{s_wd}/dwi/{s}_DWI_corr.nii.gz')

for f_path in s_list:
    s = os.path.basename(f_path).split('.\')[0].split('DWI\')[0][-1]
    f_name = os.path.basename(f_path).split('.\')[0]
    s_wd = f'{main_dir}/{s}';
    s_error=[]
    if s in df['SubjID'].values:
        continue
    print(f'Adding subject {s}...')

    try:
        raw_data = nib.load(f_path)
        data = raw_data.get_fdata()
        affine = raw_data.affine
        bvals, bvecs = read_bvals_bvecs(f'{f_path[:-7]}.bval', f'{f_path[:-7]}.bvec')
        gtab = gradient_table(bvals, bvecs)
        mask = nib.load(f'{s_wd}/dwi/{s}_MNI2diff_brainmask.nii.gz').get_fdata()
        mask = np.squeeze(mask)
        wm_mask = nib.load(f'{s_wd}/dwi/{s}_T1w2diff_wm_mask.nii.gz').get_fdata()
        wm_mask = np.squeeze (wm_mask)
        gm_mask = nib.load(f'{s_wd}/dwi/{s}_T1w2diff_gm_mask.nii.gz').get_fdata()
        gm_mask = np.squeeze (gm_mask)
        csf_mask = nib.load(f'{s_wd}/dwi/{s}_T1w2diff_csf_mask.nii.gz').get_fdata()
        csf_mask= np.squeeze (csf_mask)
tenmodel = TensorModel(gtab)
tensofit = tenmodel.fit(data, mask=mask)

fa = tensorfit.fa
fa_wm = fa[fa * wm_mask > 0]
fa_gm = fa[fa * gm_mask > 0]
fa_csf = fa[fa * csf_mask > 0]
fa_m = fa[fa * mask > 0]

s_dict = {'SubjID': s,
          'FA WM': np.mean(fa_wm),
          'FA GM': np.mean(fa_gm),
          'FA CSF': np.mean(fa_csf),
          'FA mask': np.mean(fa_m),
          'std FA WM': np.std(fa_wm),
          'std FA GM': np.std(fa_gm),
          'std FA CSF': np.std(fa_csf),
          'std FA mask': np.std(fa_m)}

atlas_data = nib.load(f'{s_wd}/dwi/{s}_JHU2diff_labels.nii.gz')
atlas_matrix = np.squeeze(atlas_data.get_fdata())

for e in np.unique(atlas_matrix)[np.unique(atlas_matrix) > 0]:
    e_mask = fa[atlas_matrix == e]
    s_dict[f'{atlas_labels[e]} mean'] = np.mean(e_mask)
    s_dict[f'{atlas_labels[e]} std'] = np.std(e_mask)

df = df.append(s_dict, ignore_index=True)
except:
    s_error.append(s)
df.set_index('SubjID', inplace = True)
df.to_csv(output_file)
| SubjID | Cerebral ped. L mean | Cerebral ped. L std | Temporal L mean | Temporal L std | Angular L mean | Angular L std | Interhemispheric mean | Interhemispheric std | Cerebral ped. L mean | Cerebral ped. L std | Temporal L mean | Temporal L std | Angular L mean | Angular L std | Interhemispheric mean | Interhemispheric std |
|--------|---------------------|-------------------|----------------|--------------|---------------|--------------|----------------------|---------------------|---------------------|-------------------|----------------|--------------|--------------|---------------|--------------|----------------------|---------------------|
| 0.21515098536484717 | 0.1590500105539045 | 0.1517238888723687 | 0.1463943916643388 | 0.1413204800990784 | 0.1904500869019419 | 0.1873522178063691 | 0.2301319928656143 | 0.2239085411690844 | 0.21515098536484717 | 0.1590500105539045 | 0.1463943916643388 | 0.1413204800990784 | 0.1904500869019419 | 0.1873522178063691 | 0.2301319928656143 | 0.2239085411690844 |
| 0.162564652099326 | 0.1590500105539045 | 0.1517238888723687 | 0.1463943916643388 | 0.1413204800990784 | 0.1904500869019419 | 0.1873522178063691 | 0.2301319928656143 | 0.2239085411690844 | 0.162564652099326 | 0.1590500105539045 | 0.1463943916643388 | 0.1413204800990784 | 0.1904500869019419 | 0.1873522178063691 | 0.2301319928656143 | 0.2239085411690844 |
| 0.574690773176503 | 0.5410667267708713 | 0.4193517898315078 | 0.5529843040988727 | 0.5581045047314992 | 0.5172392270074219 | 0.5037910522598442 | 0.5397218225154337 | 0.5655237090113543 | 0.574690773176503 | 0.5410667267708713 | 0.5529843040988727 | 0.5581045047314992 | 0.5172392270074219 | 0.5037910522598442 | 0.5397218225154337 | 0.5655237090113543 |
|----------------------------------------|----------------|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| Corticospinal tract R mean             |                |            |          |          |          |          |          |          |          |          |          |           |
| Fornix (cres) / Stria terminalis R mean|                |            |          |          |          |          |          |          |          |          |          |           |

*Note: Lateralization details are not clearly visible in the image.*
<table>
<thead>
<tr>
<th>Measure</th>
<th>Side</th>
<th>Hemisphere</th>
<th>Location</th>
<th>Pathology</th>
<th>Value</th>
<th>Side</th>
<th>Hemisphere</th>
<th>Location</th>
<th>Pathology</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fornix (cres)</td>
<td>R std</td>
<td>Frontal</td>
<td>Fornix</td>
<td>Normal</td>
<td>0.1946</td>
<td>R std</td>
<td>Frontal</td>
<td>Fornix</td>
<td>Normal</td>
<td>0.1946</td>
</tr>
<tr>
<td>Fornix mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inf. fronto-occipital fasc.</td>
<td>L std</td>
<td>Occipital</td>
<td>Inf. fronto-occipital fascicle</td>
<td>Normal</td>
<td>0.1859</td>
<td>L std</td>
<td>Occipital</td>
<td>Inf. fronto-occipital fascicle</td>
<td>Normal</td>
<td>0.1859</td>
</tr>
<tr>
<td>Inf. fronto-occipital fasc.</td>
<td>R std</td>
<td>Occipital</td>
<td>Inf. fronto-occipital fascicle</td>
<td>Normal</td>
<td>0.1782</td>
<td>R std</td>
<td>Occipital</td>
<td>Inf. fronto-occipital fascicle</td>
<td>Normal</td>
<td>0.1782</td>
</tr>
<tr>
<td>Fornix (cres)</td>
<td>R std</td>
<td>Frontal</td>
<td>Fornix</td>
<td>Normal</td>
<td>0.1946</td>
<td>R std</td>
<td>Frontal</td>
<td>Fornix</td>
<td>Normal</td>
<td>0.1946</td>
</tr>
<tr>
<td>Fornix mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inf. fronto-occipital fasc.</td>
<td>L std</td>
<td>Occipital</td>
<td>Inf. fronto-occipital fascicle</td>
<td>Normal</td>
<td>0.1859</td>
<td>L std</td>
<td>Occipital</td>
<td>Inf. fronto-occipital fascicle</td>
<td>Normal</td>
<td>0.1859</td>
</tr>
<tr>
<td>Inf. fronto-occipital fasc.</td>
<td>R std</td>
<td>Occipital</td>
<td>Inf. fronto-occipital fascicle</td>
<td>Normal</td>
<td>0.1782</td>
<td>R std</td>
<td>Occipital</td>
<td>Inf. fronto-occipital fascicle</td>
<td>Normal</td>
<td>0.1782</td>
</tr>
<tr>
<td>Med. lemniscus L mean</td>
<td>0.481808754418581</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Posterior corona radiata R std</td>
<td>0.19855372912878064</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Posterior limb of int. capsule L mean</td>
<td>Retrolenticular part of int. capsule L mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model</td>
<td>Description</td>
<td>Volumetric Unit</td>
<td>Mean</td>
<td>Standard Deviation</td>
<td>Maximum</td>
<td>Minimum</td>
<td>Overall Mean</td>
<td>Overall Standard Deviation</td>
<td>Overall Maximum</td>
<td>Overall Minimum</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
<td>----------------</td>
<td>------</td>
<td>-------------------</td>
<td>---------</td>
<td>---------</td>
<td>--------------</td>
<td>---------------------------</td>
<td>----------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Sup. corona radiata L mean</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sup. longitudinal fasc. R std</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Subject</td>
<td>FA</td>
<td>CSF</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>----</td>
<td>-----</td>
<td>-----</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.175630451819481</td>
<td>0.5455659492338581</td>
<td>0.175630451819481</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1800172176673931</td>
<td>0.5455659492338581</td>
<td>0.1800172176673931</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1867587002918976</td>
<td>0.5455659492338581</td>
<td>0.1867587002918976</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2297437755132596</td>
<td>0.5455659492338581</td>
<td>0.2297437755132596</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2322263693102929</td>
<td>0.5455659492338581</td>
<td>0.2322263693102929</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2998094670970895</td>
<td>0.5455659492338581</td>
<td>0.2998094670970895</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1980119328388701</td>
<td>0.5455659492338581</td>
<td>0.1980119328388701</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1904939104182373</td>
<td>0.5455659492338581</td>
<td>0.1904939104182373</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4604078054992679</td>
<td>0.5455659492338581</td>
<td>0.4604078054992679</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4618937580483078</td>
<td>0.5455659492338581</td>
<td>0.4618937580483078</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4326066456024404</td>
<td>0.5455659492338581</td>
<td>0.4326066456024404</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4593740024011473</td>
<td>0.5455659492338581</td>
<td>0.4593740024011473</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.433641419240715</td>
<td>0.5455659492338581</td>
<td>0.433641419240715</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
main_dir = '/home/marina/TFG/DATA'
output_file = '/home/marina/TFG/outputs/03_tractograms.csv'

# Libraries
import os
import pandas as pd
import nibabel as nib
from dipy.tracking.utils import length
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# DF creation
f_dir=[]
s_list=[]
for s in sorted(os.listdir(main_dir)):
    if os.path.isfile(f'{main_dir}/{s}/dwi/{s}_tractogram_CSD_50kseeds.tck'):
        s_list.append(s)
        f_dir.append(f'{main_dir}/{s}/dwi/{s}_tractogram')

for n in ['MARINA', 'SAUL']:
    for p in ['FASTnFURIOUS', 'DIEHARD', 'MORTALKOMBAT', 'SHARKNADO']:
        for v in [15, 20]:
            if os.path.isfile(f'{main_dir}/LABIMATGE_{n}/dwi/{v}voxsize_{p}/LABIMATGE_{n}_DWI_tractogram_CSD_50kseeds.tck'):
                f_dir.append(f'{main_dir}/LABIMATGE_{n}/dwi/{v}voxsize_{p}/LABIMATGE_{n}_DWI_tractogram')
                s_list.append(f'LABIMATGE_{n}_{v}voxsize_{p}')

df = pd.DataFrame()
df['SubjID'] = s_list
df['CSD_lengths'] = None
df['DTI_lengths'] = None
for i,f in enumerate(f_dir):
    for m in ['DTI', 'CSD']:
        f_name=f'{f}_{m}_50kseeds.tck'
        print(f'Working on file {f_name}')
        tractogram = nib.streamlines.load(f_name).tractogram.streamlines
        df.at[i, f'{m}_NOS'] = len(tractogram)
        df.at[i, f'{m}_lengths'] = length(tractogram)
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EPILENG_C_0021</td>
<td>37</td>
<td>33</td>
<td>27</td>
<td>21</td>
<td>15</td>
<td>9</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>EPILENG_S14_R209</td>
<td>20</td>
<td>19</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>ALBUCAT_S15_R106</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>LABIMATGE_MARINA_20voxsize_FASTnFURIOUS</td>
<td>39</td>
<td>36</td>
<td>33</td>
<td>30</td>
<td>27</td>
<td>24</td>
<td>21</td>
<td>18</td>
<td>15</td>
<td>12</td>
<td>9</td>
</tr>
<tr>
<td>LABIMATGE_MARINA_15voxsize_DIEHARD</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>LABIMATGE_MARINA_20voxsize_DIEHARD</td>
<td>77</td>
<td>73</td>
<td>69</td>
<td>65</td>
<td>61</td>
<td>57</td>
<td>53</td>
<td>49</td>
<td>45</td>
<td>41</td>
<td>37</td>
</tr>
<tr>
<td>LABIMATGE_MARINA_15voxsize_MORTALKOMBAT</td>
<td>130</td>
<td>126</td>
<td>122</td>
<td>118</td>
<td>114</td>
<td>110</td>
<td>106</td>
<td>102</td>
<td>98</td>
<td>94</td>
<td>90</td>
</tr>
<tr>
<td>LABIMATGE_MARINA_20voxsize_MORTALKOMBAT</td>
<td>429</td>
<td>425</td>
<td>421</td>
<td>417</td>
<td>413</td>
<td>409</td>
<td>405</td>
<td>401</td>
<td>397</td>
<td>393</td>
<td>389</td>
</tr>
<tr>
<td>LABIMATGE_MARINA_15voxsize_SHARKNADO</td>
<td>139</td>
<td>135</td>
<td>131</td>
<td>127</td>
<td>123</td>
<td>119</td>
<td>115</td>
<td>111</td>
<td>107</td>
<td>103</td>
<td>99</td>
</tr>
<tr>
<td>LABIMATGE_MARINA_20voxsize_SHARKNADO</td>
<td>405</td>
<td>400</td>
<td>395</td>
<td>390</td>
<td>385</td>
<td>380</td>
<td>376</td>
<td>371</td>
<td>366</td>
<td>361</td>
<td>356</td>
</tr>
<tr>
<td>LABIMATGE_SAUL_15voxsize_FASTnFURIOUS</td>
<td>18</td>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>LABIMATGE_SAUL_20voxsize_FASTnFURIOUS</td>
<td>248</td>
<td>243</td>
<td>238</td>
<td>233</td>
<td>228</td>
<td>223</td>
<td>218</td>
<td>213</td>
<td>208</td>
<td>203</td>
<td>198</td>
</tr>
<tr>
<td>LABIMATGE_SAUL_15voxsize_DIEHARD</td>
<td>35</td>
<td>34</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
<td>28</td>
<td>27</td>
<td>26</td>
<td>25</td>
</tr>
<tr>
<td>LABIMATGE_SAUL_20voxsize_DIEHARD</td>
<td>203</td>
<td>202</td>
<td>201</td>
<td>200</td>
<td>199</td>
<td>198</td>
<td>197</td>
<td>196</td>
<td>195</td>
<td>194</td>
<td>193</td>
</tr>
<tr>
<td>LABIMATGE_SAUL_15voxsize_MORTALKOMBAT</td>
<td>39</td>
<td>38</td>
<td>37</td>
<td>36</td>
<td>35</td>
<td>34</td>
<td>33</td>
<td>32</td>
<td>31</td>
<td>30</td>
<td>29</td>
</tr>
<tr>
<td>LABIMATGE_SAUL_20voxsize_MORTALKOMBAT</td>
<td>221</td>
<td>220</td>
<td>219</td>
<td>218</td>
<td>217</td>
<td>216</td>
<td>215</td>
<td>214</td>
<td>213</td>
<td>212</td>
<td>211</td>
</tr>
<tr>
<td>LABIMATGE_SAUL_15voxsize_SHARKNADO</td>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>LABIMATGE_SAUL_20voxsize_SHARKNADO</td>
<td>120</td>
<td>119</td>
<td>118</td>
<td>117</td>
<td>116</td>
<td>115</td>
<td>114</td>
<td>113</td>
<td>112</td>
<td>111</td>
<td>110</td>
</tr>
</tbody>
</table>

#!/usr/bin/env python3
# -*- coding: utf-8 -*-

"""
04_STATISTICAL ANALYSIS
"""

#% Libraries

main_dir = '/home/marina/TFG'
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np

#% Upload dataframes

SNR_df = pd.read_csv(f'{main_dir}/outputs/01_SNRs.csv')
SNR_df.set_index('SubjID', inplace = True)

FA_df = pd.read_csv(f'{main_dir}/outputs/02_FAmaps.csv')
FA_df.set_index('SubjID', inplace = True)

dataset_FA = []
for s in FA_df.index:
    if 'CTR_HSA' in s: dataset_FA.append('BBHSA')
    elif 'EOT' in s: dataset_FA.append('EOP')
    elif 'EPILENG' in s: dataset_FA.append('EPILENG')
    elif 'BIOMARCADORES' in s: dataset_FA.append('BIOMARCADORES')
    elif '20voxsize_FASTnFURIOUS' in s: dataset_FA.append('20voxsize_FASTnFURIOUS')
    elif '15voxsize_FASTnFURIOUS' in s: dataset_FA.append('15voxsize_FASTnFURIOUS')
    elif '20voxsize_DIEHARD' in s: dataset_FA.append('20voxsize_DIEHARD')
    elif '15voxsize_DIEHARD' in s: dataset_FA.append('15voxsize_DIEHARD')
    elif '20voxsize_MORTALKOMBAT' in s: dataset_FA.append('20voxsize_MORTALKOMBAT')
    elif '15voxsize_MORTALKOMBAT' in s: dataset_FA.append('15voxsize_MORTALKOMBAT')
    elif '20voxsize_SHARKNADO' in s: dataset_FA.append('20voxsize_SHARKNADO')
    elif '15voxsize_SHARKNADO' in s: dataset_FA.append('15voxsize_SHARKNADO')
    else: dataset_FA.append('ALBUCAT')

FA_df['Dataset'] = dataset_FA
FA_df['xdim']=SNR_df['xdim']

#% Figure 1: SNR  (for 1.5 mm i 2 mm)

xlabels = ['bvals [0,400]', 'bvals (400,800]', 'bvals (800,1200]',
'bvals (1200,1600)', 'bvals (1600,2000]', 'bvals (2000,2400]',
'bvals (2400,2800]', 'bvals (2800,3200]'

sns.set(font_scale=1.2)
fig, axis = plt.subplots(1,2, figsize=(12,5))
fig.suptitle('SNR per each b-value interval and voxel size', fontsize = 20)
a = sns.boxplot(data=SNR_df[SNR_df.xdim==1.5][xlabels], palette='pastel', ax=ax)
a.set(ylim=(0,100))
plt.ylabel('SNR')
ax[0].set_title('voxel size = 1.5mm', fontsize = 20)
a.set_xticklabels(labels= xlabels, rotation=45)

b = sns.boxplot(data=SNR_df[SNR_df.xdim==2][xlabels], palette='pastel', ax=axis
b.set(ylim=(0,100))
axis[1].set_title('voxel size = 2mm', fontsize = 20)
sns.set(rc={'figure.dpi':300, 'savefig.dpi':300})
plt.xticks(rotation=45)
plt.ylabel('SNR')
plt.tight_layout()
plt.show()

#%% Fig 2: FA (1.5 mm)
sns.set(font_scale=1.2)
labels1= ['BBHSA', 'EPILENG','ALBUCAT', '15v_FAST', '15v_DIE','15v_MORT', '15v_SHAR']

fig, axes = plt.subplots(3, 2, figsize=(18, 10))
fig.suptitle('FA per each mask and dataset (voxel size = 1.5mm)', fontsize = 30)
a = sns.boxplot(ax=axes[0, 0], data=FA_df[FA_df.xdim==1.5], y='FA GM', x = 'Dataset'
a.set(ylim=(0.08,0.188))
a.set_xticklabels(labels = labels1, rotation = 0)
b = sns.boxplot(ax=axes[0, 1], data=FA_df[FA_df.xdim==1.5], y='std FA GM', x = 'Dataset'
b.set(ylim=(0.08,0.188))
b.set_xticklabels(labels = labels1, rotation = 0)
c = sns.boxplot(ax=axes[1, 0], data=FA_df[FA_df.xdim==1.5], y='FA WM', x = 'Dataset'
c.set(ylim=(0.17,0.45))
c.set_xticklabels(labels = labels1, rotation = 0)
d = sns.boxplot(ax=axes[1, 1], data=FA_df[FA_df.xdim==1.5], y='std FA WM', x = 'Dataset'
d.set(ylim=(0.17,0.45))
d.set_xticklabels(labels = labels1, rotation = 0)
e = sns.boxplot(ax=axes[2, 0], data=FA_df[FA_df.xdim==1.5], y='FA CSF', x = 'Dataset'
e.set(ylim=(0.09,0.25))
e.set_xticklabels(labels = labels1, rotation = 0)
f = sns.boxplot(ax=axes[2, 1], data=FA_df[FA_df.xdim==1.5], y='std FA CSF', x = 'Dataset'
f.set(ylim=(0.09,0.25))
f.set_xticklabels(labels = labels1, rotation = 0)

plt.tight_layout()

#%% Fig 3: FA (2 mm)
sns.set(font_scale=1.2)

labels2= ['BIOMARCADORES', 'EOP', '20v_FAST', '20_DIE','20v_MORT', '20v_SHAR']

fig, axes = plt.subplots(3, 2, figsize=(18, 10))
fig.suptitle('FA per each mask and dataset (voxel size = 2mm)', fontsize = 30)
a = sns.boxplot(ax=axes[0, 0], data=FA_df[FA_df.xdim==2], y='FA GM', x = 'Dataset'
a.set_ylim((0,100))
a.set_xticklabels(labels= xlabels, rotation=45)

b = sns.boxplot(data=SNR_df[SNR_df.xdim==2][xlabels], palette='pastel', ax=axis
b.set(ylim=(0,100))
axis[1].set_title('voxel size = 2mm', fontsize = 20)
sns.set(rc={'figure.dpi':300, 'savefig.dpi':300})
plt.xticks(rotation=45)
plt.ylabel('SNR')
plt.tight_layout()
plt.show()
a.set(ylim=(0.08, 0.21))
a.set_xticklabels(labels = labels2, rotation = 0)

b = sns.boxplot(ax=axes[0, 1], data=FA_df[FA_df.xdim==2], y='std FA GM', x='Dataset')
b.set(ylim=(0.08, 0.21))
b.set_xticklabels(labels = labels2, rotation = 0)

c = sns.boxplot(ax=axes[1, 0], data=FA_df[FA_df.xdim==2], y='FA WM', x='Dataset')
c.set(ylim=(0.165, 0.422))
c.set_xticklabels(labels = labels2, rotation = 0)

d = sns.boxplot(ax=axes[1, 1], data=FA_df[FA_df.xdim==2], y='std FA WM', x='Dataset')
d.set(ylim=(0.165, 0.422))
d.set_xticklabels(labels = labels2, rotation = 0)

e = sns.boxplot(ax=axes[2, 0], data=FA_df[FA_df.xdim==2], y='FA CSF', x='Dataset')
e.set(ylim=(0.115, 0.248))
e.set_xticklabels(labels = labels2, rotation = 0)

f = sns.boxplot(ax=axes[2, 1], data=FA_df[FA_df.xdim==2], y='std FA CSF', x='Dataset')
f.set(ylim=(0.115, 0.248))
f.set_xticklabels(labels = labels2, rotation = 0)

plt.tight_layout()

# Fig 4: Heatmap FA


regional_fa = FA_df[labels2].groupby('Dataset').mean()

plt.figure(figsize=(15, 8))

h = sns.heatmap(regional_fa, cmap='jet', xticklabels=True, yticklabels=False)
plt.title('WM atlas labels mean FA value per each dataset')
plt.xlabel('Atlas labels')
sns.set(rc={'figure.dpi':300, 'savefig.dpi':300})

# Read Tractogram df

main_dir = '/home/marina/TFG'
df = pd.read_pickle(f'{main_dir}/outputs/03_tractogram_data.pkl')
df.set_index('SubjID', inplace=True)
dataset = []
for s in df.index:
    if 'CTR_HSA' in s: dataset.append('BBHSA')
    elif 'EOT' in s: dataset.append('EOP')
    elif 'EPILENG' in s: dataset.append('EPILEN')
    elif 'BIOMARCADORES' in s: dataset.append('BIOMARCADORES')
    elif '20voxsize_FASTnFURIOUS' in s: dataset.append('20voxsize_FASTnFURIOUS')
    elif '15voxsize_FASTnFURIOUS' in s: dataset.append('15voxsize_FASTnFURIOUS')
    elif '20voxsize_DIEHARD' in s: dataset.append('20voxsize_DIEHARD')
    elif '15voxsize_DIEHARD' in s: dataset.append('15voxsize_DIEHARD')
    elif '20voxsize_MORTALKOMBAT' in s: dataset.append('20voxsize_MORTALKOMBAT')
    elif '15voxsize_MORTALKOMBAT' in s: dataset.append('15voxsize_MORTALKOMBAT')
    elif '20voxsize_SHARKNADO' in s: dataset.append('20voxsize_SHARKNADO')
    elif '15voxsize_SHARKNADO' in s: dataset.append('15voxsize_SHARKNADO')
else: dataset.append('ALBUCAT')

df['Dataset'] = dataset
df['xdim'] = FA_df['xdim']

df.to_csv('/Users/rin/Desktop/spyder/outputs/03_tractogram.csv')

#% Extract CSD_lengths per acquisition
for s in df.index:
    print(f'df.loc["{s}"]["CSD_lengths"}')

BIOMARCADORES_s_list = df.loc['BIOMARCADORES_01']['CSD_lengths'], df.loc['BIOMAR
EOP_s_list = [df.loc['EOT_FARAO23048']]["CSD_lengths"], df.loc['EOT_FARAO25000']['
EPILEN_s_list = [df.loc['EPILEN_C_0000']["CSD_lengths"], df.loc['EPILEN_C_000
BBBHSA_s_list = [df.loc['CTR_HSA_01']["CSD_lengths"], df.loc['CTR_HSA_02']["CS
ALBUCAT_s_list = [df.loc['S14_R209']["CSD_lengths"], df.loc['S15_R106']["CSD
LABF2_s_list = [df.loc['LABIMATGE_MARINA_20voxsize_FASTnFURIOUS']["CSD_lengths"
LABF1_s_list = [df.loc['LABIMATGE_SAUL_15voxsize_FASTnFURIOUS']["CSD_lengths"
LABD1_s_list = [df.loc['LABIMATGE_MARINA_15voxsize_DIEHARD']["CSD_lengths"], d
LABD2_s_list = [df.loc['LABIMATGE_MARINA_20voxsize_DIEHARD']["CSD_lengths"], d
LABM1_s_list = [df.loc['LABIMATGE_MARINA_15voxsize_MORTALKOMBAT']["CSD_lengths"
LABM2_s_list = [df.loc['LABIMATGE_MARINA_20voxsize_MORTALKOMBAT']["CSD_lengths"
LABS1_s_list = [df.loc['LABIMATGE_MARINA_15voxsize_SHARKNADO']["CSD_lengths"],
LABS2_s_list = [df.loc['LABIMATGE_MARINA_20voxsize_SHARKNADO']["CSD_lengths"]

sample_1_csd = np.concatenate((BIOMARCADORES_s_list), axis = 0)
sample_2_csd = np.concatenate((EOP_s_list), axis = 0)
sample_3_csd = np.concatenate((EPILEN_s_list), axis = 0)
sample_4_csd = np.concatenate((BBHSA_s_list), axis = 0)
sample_5_csd = np.concatenate((LABF2_s_list), axis = 0)
sample_6_csd = np.concatenate((LABF1_s_list), axis = 0)
sample_7_csd = np.concatenate((LABD1_s_list), axis = 0)
sample_8_csd = np.concatenate((LABD2_s_list), axis = 0)
sample_9_csd = np.concatenate((LABM1_s_list), axis = 0)
sample_10_csd = np.concatenate((LABM2_s_list), axis = 0)
sample_11_csd = np.concatenate((LABS1_s_list), axis = 0)
sample_12_csd = np.concatenate((LABS2_s_list), axis = 0)
sample_13_csd = np.concatenate((ALBUCAT_s_list), axis = 0)
%% Fig 5,6: 1.5mm CSD

```python
c = sns.boxplot(data = df[df.xdim==1.5], y = 'Dataset', palette='Set3', ax=axis[0,1])
a = sns.histplot(x = sample_12_csd, weights = 1/len(LABF1_s_list), bins = 100, color='red', fill=False, label='BIOM', element='step', ax=axis[0,0])
fig.suptitle('CSD streamlines length per each dataset (voxel size 1.5mm)')
```

%% Fig 7,8 2mm CSD

```python
c = sns.boxplot(data = df[df.xdim==2], y = 'Dataset', palette='Set3', ax=axis[0,0])
a = sns.histplot(x = sample_11_csd, weights = 1/len(LABS1_s_list), bins = 100, color='green', fill=False, label='BIOM', element='step', ax=axis[0,1])
fig.suptitle('CSD streamlines length per each dataset (voxel size 2mm)')
```

%% Fig 9. NOS

```python
fig, axis = plt.subplots(2,2, figsize=(12,8))
fig.suptitle('Number of streamlines (NOS) per each algorithm, dataset and voxel size')
a = sns.boxplot(data = df[df.xdim==1.5], y = 'NOS', x = 'Dataset', palette='pastel', ax=axis[0,0].set_title('NOS CSD v=1.5mm'))
b = sns.boxplot(data = df[df.xdim==2], y = 'NOS', x = 'Dataset', palette='pastel', ax=axis[0,1].set_title('NOS CSD v=2mm'))
c = sns.boxplot(data = df[df.xdim==1.5], y = 'NOS', x = 'Algorithm', ax=axis[1,0].set_title('NOS CSD v=1.5mm'))
d = sns.boxplot(data = df[df.xdim==2], y = 'NOS', x = 'Algorithm', ax=axis[1,1].set_title('NOS CSD v=2mm'))
```
```python
axis[1,0].set_title('NOS DTI v=2mm')
c.set_xticklabels(labels = labels2, rotation = 90)
c.set(ylim=(38000,48000))

plt.legend()

# Extract DTI_lengths per acquisition

BIOMARCADORES_s_list = df.loc[['BIOMARCADORES_01'], ['DTI_lengths'], df.loc[['BIOMBARDO_DATS_02'], ['DTI_lengths'], df.loc[['EOT_FARA023048'], ['DTI_lengths'], df.loc[['EOT_FARA025000'], ['DTI_lengths'],
EPILENG_s_list = [df.loc[['EOT_FARA023048'], ['DTI_lengths'], df.loc[['EOT_FARA025000'], ['DTI_lengths'], df.loc[['EPILENG_C_0001'], ['DTI_lengths'], df.loc[['EPILENG_C_0002'], ['DTI_lengths'],
BBBHSA_s_list = [df.loc[['CTR_HSA_01'], ['DTI_lengths'], df.loc[['CTR_HSA_02'], ['DTI_lengths'], df.loc[['CTR_HSA_03'], ['DTI_lengths'],
ALBUCA_T_s_list = [df.loc [['S14_R209'], ['DTI_lengths'], df.loc ['S15_R106'], ['DTI_lengths]
LABF2_s_list = [df.loc ['LABIMATGE_MARINA_20voxsize_FASTnFURIOUS'], ['DTI_lengths'],
LABF1_s_list = [df.loc ['LABIMATGE_SAUL_15voxsize_FASTnFURIOUS'], ['DTI_lengths'],
LABD1_s_list = [df.loc ['LABIMATGE_MARINA_20voxsize_MORTALKOMBAT'], ['DTI_lengths'],
LABD2_s_list = [df.loc ['LABIMATGE_MARINA_20voxsize_MORTALKOMBAT'], ['DTI_lengths'],
LABM1_s_list = [df.loc ['LABIMATGE_MARINA_15voxsize_MORTALKOMBAT'], ['DTI_lengths'],
LABM2_s_list = [df.loc ['LABIMATGE_MARINA_20voxsize_MORTALKOMBAT'], ['DTI_lengths'],
LABS1_s_list = [df.loc ['LABIMATGE_MARINA_15voxsize_SHARKNADO'], ['DTI_lengths'],
LABS2_s_list = [df.loc ['LABIMATGE_MARINA_15voxsize_SHARKNADO'], ['DTI_lengths'],
sample_1_dti = np.concatenate((BIOMARCADORES_s_list), axis = 0)
sample_2_dti = np.concatenate((EOP_s_list), axis = 0)
sample_3_dti = np.concatenate((EPILENG_s_list), axis = 0)
sample_4_dti = np.concatenate((BBBHSA_s_list), axis = 0)
sample_5_dti = np.concatenate((LABF2_s_list), axis = 0)
sample_6_dti = np.concatenate((LABF1_s_list), axis = 0)
sample_7_dti = np.concatenate((LABD1_s_list), axis = 0)
sample_8_dti = np.concatenate((LABD2_s_list), axis = 0)
sample_9_dti = np.concatenate((LABM1_s_list), axis = 0)
sample_10_dti = np.concatenate((LABM2_s_list), axis = 0)
sample_11_dti = np.concatenate((LABS1_s_list), axis = 0)
sample_12_dti = np.concatenate((LABS2_s_list), axis = 0)
sample_13_dti = np.concatenate((ALBUCA_T_s_list), axis = 0)

# Fig 10,11 1.5mm DTI

plt.figure()
sns.histplot(x = sample_13_dti, weights = 1/len(ALBUCA_T_s_list), bins = 100, color)
sns.histplot(x = sample_4_dti, weights = 1/len(BBBHSA_s_list), bins = 100, color)
sns.histplot(x = sample_3_dti, weights = 1/len(EPILENG_s_list), bins = 100, color)
sns.histplot(x = sample_6_dti, weights = 1/len(LABF1_s_list), bins = 100, color)
sns.histplot(x = sample_7_dti, weights = 1/len(LABD1_s_list), bins = 100, color)
sns.histplot(x = sample_9_dti, weights = 1/len(LABM1_s_list), bins = 100, color)
sns.histplot(x = sample_11_dti, weights = 1/len(LABS1_s_list), bins = 100, color)
plt.xlim([0,1000])
plt.set(rc= {"figure.dpi":300, 'savefig.dpi':300})
plt.title('DTI streamlines length per each dataset (voxel size 1.5mm)')
plt.tight_layout()
plt.legend()
```
plt.xlabel('Length')
plt.ylabel('# of streamlines')

# % Fig 12,13 2mm DTI
plt.figure()
sns.histplot(x = sample_1_dti, weights = 1/len(BIOMARCADORES_s_list), bins = 100, color = 'green', element='step', fill = False, label = 'BIOMARCADORES')
sns.histplot(x = sample_2_dti, weights = 1/len(EOP_s_list), bins = 100, color = 'blue', element='step', fill = False, label = 'EOP')
sns.histplot(x = sample_5_dti, weights = 1/len(LABF2_s_list), bins = 100, color = 'pink', element='step', fill = False, label = 'LABF2')
sns.histplot(x = sample_8_dti, weights = 1/len(LABD2_s_list), bins = 100, color = 'red', element='step', fill = False, label = 'LABD2')
sns.histplot(x = sample_10_dti, weights = 1/len(LABM2_s_list), bins = 100, color = 'orange', element='step', fill = False, label = 'LABM2')
sns.histplot(x = sample_12_dti, weights = 1/len(LABS2_s_list), bins = 100, color = 'purple', element='step', fill = False, label = 'LABS2')
plt.xlim([0,100])
sns.set(rc={'figure.dpi':300, 'savefig.dpi':300})
plt.title('DTI streamlines length per each dataset (voxel size 2mm) ')
plt.tight_layout()
plt.legend()
plt.xlabel('Length')
plt.ylabel('# of streamlines')