Development of a 3D structural MRI preprocessing pipeline based on Advanced Normalization Tools (ANTs)

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Adrià Solana

In partial fulfillment

of the requirements for the degree in

TELECOMMUNICATIONS ENGINEERING

Advisors: Irene Cumplido

and Verónica Vilaplana

Barcelona, June 2022

Abstract

To tackle distinguishing normal brain aging from pathological aging, MRI scans are often used to study the brains' structure. Nevertheless, these scans frequently present several problems that harden extracting conclusions, so they have to undergo a series of image processing techniques to improve their usefulness.

This thesis studies the use of Advanced Normalization Tools (ANTs) as the core of the processing pipeline, complemented by a quality check system to assess its outputs and a method to monitor the computational resource usage of the whole process, in addition to extracting structural data to feed a Machine Learning brain age prediction system. The outcomes of this thesis include the optimal way to use ANTs in this context, a volume correlation system to evaluate its results, a simple Linux command based resource usage summarisation program and the results from feeding the extracted data to a XGBoost brain age prediction system.

Resum

Per distingir l'envelliment cerebral normal de l'envelliment patològic cal un estudi de l'estructura del cervell a través de ressonàncies magnètiques. Tot i això, aquestes imatges solen presentar diversos problemes que dificulten l'extracció de conclusions, per la qual cosa s'han de sotmetre a una sèrie de tècniques de processament de la imatge per fer-les més útils.

Aquesta tesi estudia l'ús d'Advanced Normalization Tools (ANTs) com a nucli del processat, complementat amb un sistema de control de qualitat per avaluar-ne els resultats i un mètode per monitoritzar l'ús de recursos computacionals de tot el procés, a més d'explorar com fer servir les dades extretes per alimentar un sistema Machine Learning de predicció de l’edat cerebral. Els resultats d'aquesta tesi inclouen: la forma òptima de fer servir ANTs en aquest context, un sistema de correlació de volums per avaluar els seus resultats, un sistema simple de registre de l'ús de recursos basat en comandes de Linux i els diversos resultats d'alimentar les dades extretes al sistema de predicció de l'edat cerebral XGBoost.

Resumen

Para distinguir el envejecimiento cerebral normal del envejecimiento patológico se requiere un estudio de la estructura del cerebro a través de resonancias magnéticas. No obstante, dichas imágenes suelen presentar varios problemas que dificultan la extracción de conclusiones, por lo que se deben someter a una serie de técnicas de procesado de la imagen para mejorar su utilidad.

Esta tesis estudia el uso de Advanced Normalization Tools (ANTs) como núcleo del procesado, complementado con un sistema de control de calidad para evaluar sus resultados y un método para monitorear el uso de recursos computacionales de todo el proceso, además de explorar cómo usar los datos extraídos para alimentar un sistema Machine Learning de predicción de la edad cerebral. Los resultados de esta tesis incluyen: la forma óptima de usar ANTs en este contexto, un sistema de correlación de volúmenes para evaluar sus resultados, un sistema simple de registro del uso de recursos basado en comandos de Linux y los resultados de alimentar diferentes los datos extraídos al sistema de predicción de la edad cerebral XGBoost.

Acknowledgements

Firstly, I would like to thank my advisors: Verónica Vilaplana -UPC professor- and Irene Cumplido -BBRC PhD student- for giving me valuable guidance and advice during the extension of this project. Furthermore, I would also like to express my gratitude to the UPC and BBRC institutions themselves for giving me the opportunity to work on this project.

Lastly, I wish to acknowledge the companionship displayed by my fellow BBRC PhD students and staff, in addition to the different insights and opinions provided by my friends and family.

Revision history and approval record

Revision	Date	Purpose
0	02/04/2022	Document creation
1	13/06/2022	Document revision
2	17/06/2022	Document revision
3	20/06/2022	Final revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Adrià Solana Carulla	adria.solana@estudiantat.upc.edu
Verónica Vilaplana Besler	veronica.vilaplana@upc.edu
Irene Cumplido Mayoral	icumplido@barcelonabeta.org

Written by:		Reviewed and approved by:	
Date	21/06/2022	Date	21/06/2022
Name	Adrià Solana	Name	Verónica Vilaplana
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables

1. Introduction

1.1. Statement of purpose

The main objective of this project is to develop a dataset of processed magnetic resonance imaging (MRI) scans along with various quality control systems associated with the processing pipeline. The goal of this dataset will be feeding a Machine Learning (ML) model capable of extracting conclusions about each subject's MRI scan. In particular, this dataset will be used to train a Machine Learning based brain age prediction system.

This project is a collaborative effort with the Barcelonabeta Brain Research Center (BBRC), and its results will be used by them in current and future research. In consequence, each goal is defined to accomplish a real world need regarding such research. In the light of the above, the goal breakdown remains as follows:

- Achieve a certain level of expertise on the tools that BBRC researchers usually work with -mainly Advanced Normalization Tools (ANTs)- and document the most important parts regarding the project's needs. For future investigation, this will be useful to avoid having to research the same subjects over again.
- Create and compute metrics to assess the computational resource consumption of the preprocessing pipeline. In current research performed by the BBRC, the main code containing the pipeline will be run by internal or external computing services. Such services require a summary of these resource usage metrics before actually running the associated code.
- Create and compute metrics to assess the processed dataset. Achieving this goal is necessary to discard any wrongly processed image that could adulterate the training stage of some Machine Learning model.
- Extract volumes and thicknesses for each cortical region contained in a defined brain atlas. These values will be the ones fed to the ML system.
- Process a subset of images with all the previous steps to make sure it works as expected, and therefore it is ready to process a greater image batch.
- Adapt a brain age prediction Machine Learning system to accept cortical regions volumes and thicknesses as input. Assess the outcome with the data extracted in the previous goal. In particular, an XGBoost system will be used to perform a direct comparison with previous BBRC research, where such a system was used to read volumes and thicknesses extracted from another pipeline to perform brain age prediction.

1.2. Gantt diagram

The Gantt diagram was made using the Tom's planner software and it can be seen online and interactively as in Appendices [A1][1] (sub-section A1.3.3.), along with other sections regarding the work planification and structure.

[1] This document uses ['x'] as Bibliography references and ['Ax'] as Appendix references.

2. Theoretic background

With age, the human brain undergoes several changes concerning both tissue composition and volume. Nevertheless, these natural aging traces overlap with the most common symptoms of neuro-degenerative diseases such as Alzheimer's Disease (AD) [1]. Therefore, the medical community finds it of great interest to develop systems to distinguish between natural and pathological aging.

The first indicator of pathological aging usually consists in abnormalities in sizes or volumes along various regions of the brain [2]. In this thesis, the most recurrent indicator is a subject's cortical thickness being unusually shrunk for its demographic segment (age, sex, studies, …). "Cortical" refers to the brain cortex, i.e. the most external layer of the brain mainly composed of gray matter (see Figure 1). Therefore, the gray matter's thickness should be a good indicator to infer if a subject suffers from a neuro-degenerative disease.

Nonetheless, extracting data such as gray matter regions' thicknesses or volumes without human intervention is not a trivial task: MRI scans have to undergo a series of image processing techniques to easen segmenting the brain into regions of interest or ROI (see Figure 2). These techniques fix common problems such as "homogenizing" groups of scans being obtained from different sites or machines, or correcting artifacts in scans' intensities.

To make sure that the resulting images can be compared with each other under the same conditions, all of these procedures have to use a common brain MRI scan known as the "template". In this context, a template is an image that represents the most common characteristics of an MRI dataset, therefore, it is usually obtained from "merging" all the components of a dataset into one single representative image. The main characteristics that the template holds are:

- Origin: a tridimensional point that defines where the center of the image is.
- Orientation: a tridimensional vector that indicates where the image contents are facing.
- Space: a particular set of origin and orientation.
- Volume: a "mean" value for all the dataset subjects.

Therefore, the goal of the template is anchoring all the processed images to the same spatial features, so each voxel (tridimensional pixel) of an image is analogous to the same point in every other scan, all this without removing each subject's particular information. This procedure is known as "normalizing" an image to a template, whilst the smaller task of shifting the image to match a space is commonly referred as "registration". Also, in addition to the main image, it is common to obtain brain segments and masks from the template creation process -for example, an image containing only the brain without the skull, various images containing corresponding brain segments, etc.-, all of them sharing the template's spatial metadata, that are also required for tasks such as brain extraction or segmentation.

All in all, the whole processing pipeline can be summarized as a series of image processing techniques that, holding onto a template, transform raw brain MRI scans into useful information -both images and numbers- that can be used, for instance, to estimate a person's brain age among other tasks. The technicalities underlying this process will be discussed further in the thesis.

Additionally, an illustration of the main brain tissues can be seen in Figure 1 to better envision the explanations regarding segmentations.

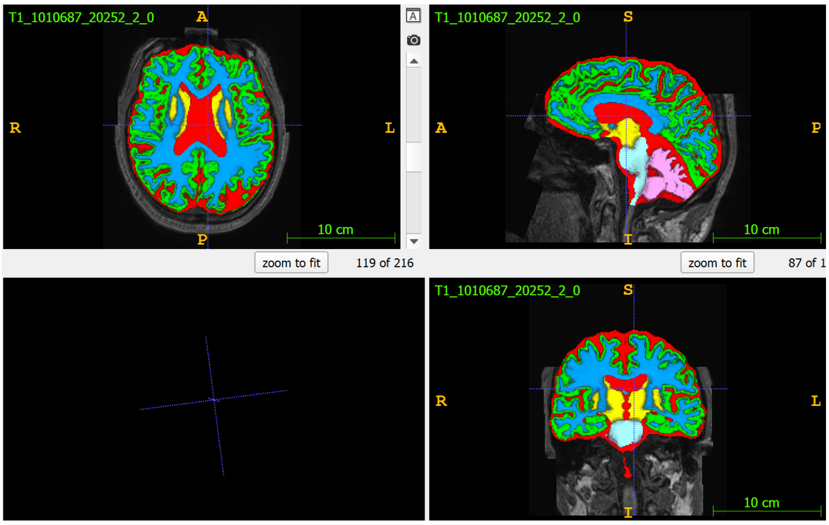


Figure 1: Basic brain tissues from three views: superior -from above-, sagittal -from the side- and anterior -from up front- (listed from left to right and up to down, not minding the empty box).

- Gray: tissues and structures that are not taken into account. In the superior view (box 1), the gray intensity enveloping the brain shows the subject's skull.
- Red: Cerebrospinal Fluid (CSF) - A protecting fluid filling the volume between the brain and the skull and inside the ventricles.
- Green: Gray Matter (GM) - The most external part of the brain from which the cortical thickness is extracted, containing neuron cell bodies.
- Blue: White Matter (WM) - Deeper within the brain, provides neuron connectivity.
- Yellow: Deep Gray Matter (DGM) - The internal GM, not taken into account when referring to the cortex.
- Cyan: Brain medulla.
- Pink: Cerebellum.

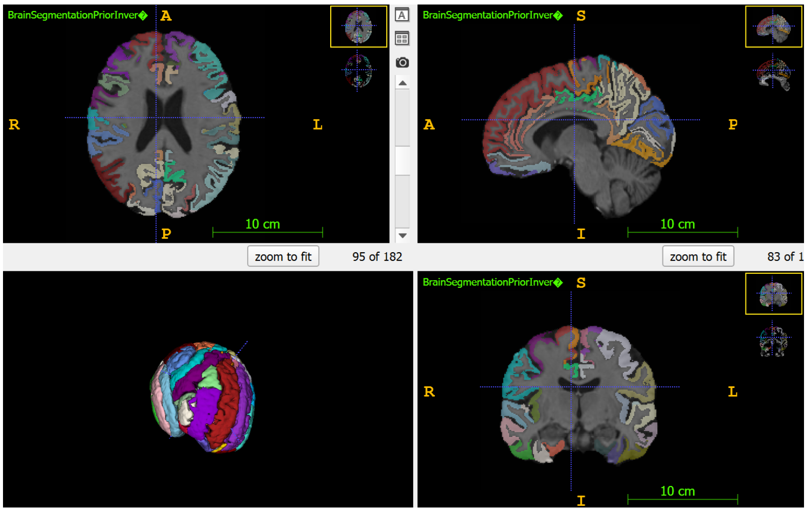

BrainSegmentationPriorInver◆
A
R
L
P
10 cm
zoom to fit
95 of 182
BrainSegmentationPriorInver◆
S
A
P
I
10 cm
zoom to fit
83 of 1
BrainSegmentationPriorInver◆
S
R
L
I
10 cm

Figure 2: Cortex segmentation into 31 ROI for each hemisphere. Each color represents a ROI.

To end with, some concepts will be developed to deepen into the most important ideas.

- MRI: these imaging techniques rely on magnetic fields to register 3-dimensional images that represent different kinds of tissues with different intensities. The images can be obtained using several modalities that exhibit the tissues with different intensities: for example, the modality known as T1 displays CSF as dark while T2 associates this tissue with lighter colors [A11]. As T1 images hold better information to measure structural volume, this project uses T1-weighted MRI scans.
- Coordinate systems and image space: in MRI scanning, coordinates can refer to world, anatomical or image coordinates. World coordinates are referenced to the machine that performs the scan, anatomical coordinates agree with the 3 perpendicular views from which a human body is typically studied, and image coordinates define the positions of each voxel in an image [A12]. This project employs anatomical coordinates when implying MRI image views, as in Figure 1, and it makes use of image coordinates when mentioning image origin, orientation, and space. Thus, an image's origin is a point in image coordinates where the image center is located, its orientation is a vector pointing to the direction that the image faces towards and its space is defined by the combination of these two [A15].
- Template: an anatomical template is an image that has to serve as a common reference to compare data from multiple subjects, either in terms of space or volume. Therefore, a useful template has to represent the "average" characteristics of the images under study. As creating a template is a computationally demanding process, it is common to use public templates [A13], which also favors the comparability across studies. Nevertheless, generating a custom template for a cohort (or, in this context, group of images under study) often results beneficial for tasks involving comparing processed images from said cohort.
- Image registration: in general, a registration aligns an image from one space to another using a second image as a reference [A15]. The registration is performed by maximizing a given similarity metric between both images under certain constraints. There are many types of registrations, but they can be divided into two subgroups: linear and nonlinear. The first group includes all transformations that translate, rotate and scale images (keeping parallel lines as such) so that their global features match the ones from the reference. The second group comprises transformations that freely deform images without any kind of constraint, and therefore are used to match local features or details to the reference image. Thus, at the end of a registration the image subject to it does not only align with a reference space (which could be done with only a translation to align the origin and a rotation to align the orientation), but also can do so in a way that matches the reference image itself to a certain extent.
- Parcellation and brain atlases: in this context, parcellation refers to subdividing brain tissues into ROI (Figure 2) for a more comprehensive analysis. This can be referred to as a segmentation, but it should not be confused with the priorly mentioned tissue segmentation. For instance, in this project the brain cortex is parcellated to differentiate several ROI for their study afterwards. An atlas -or brain map- [A14] is expected to perform this procedure, and while brain atlases often cover all the brain's volume, the tissue subject to parcellation can be a portion of the total volume.

3. State of the art of the technology used or applied in this thesis

3.1. MRI normalization and cortical thickness extraction

Even if ANTs is the normalization tool that BBRC suggested for this project, a more extensive research has been done in an effort to better understand the general processing pipeline, starting from raw images and ending up with the extracted cortical thickness.

3.1.1. Freesurfer

Freesurfer is an open source neuroimaging toolkit that offers human brain MRI scans' analysis and visualization tools [3]. The software approaches the cortical thickness extraction from a surface perspective, therefore it computes its surfaces rather than the volume. This pipeline comprises the following steps [4]:

1. Talairach Registration: this procedure registers the input image to a defined space using a large amount of previously registered images as reference.
2. Intensity Normalization: this step removes the magnetic field-induced artifacts to the possible extent.
3. Skull Stripping: starting from a tessellated ellipsoidal template, the figure is algorithmically deformed to match the brain. Then, the analogous volume in the registered, intensity normalized image is snipped out.
4. White Matter Labeling: this procedure identifies which points in the previous image correspond to white matter and which ones compose gray matter.
5. Cutting Planes: in this step, two planes are automatically defined to divide the hemispheres and separate the subcortical regions.
6. Connected Components: this stage generates one continuous mass for each hemisphere.
7. Surface Tessellation, Refinement, and Deformation: this final step computes a connected surface that starts from the white matter / gray matter separation and expands to the brain boundaries calculated earlier. As the volume between the white matter and these boundaries is mostly gray matter, this surface coats the brain cortex.

3.1.2. ANTs

ANTs is another open source toolkit that provides tools able to perform common neuroimaging techniques such as template creation, brain extraction and segmentation, image corrections, etc. One of its most broadly used tools is *antsCorticalThickness* [5], which consists of a 6-stage pipeline that ends up extracting the cortical thickness from a brain by linking various tools also developed by the ANTs' team. The steps to achieve this goal are:

1. Brain extraction: firstly, an intensity correction technique known as *N4 Bias Field Correction* is applied to smooth non-homogeneities. Then, the mask of the template's extracted brain is registered to optimally match the input's brain, and this is refined with an intensity distribution-based segmentation technique known

as *Atropos*. Finally, the warped mask snips out the brain from the whole head image.

2. Template registration: this step registers the template's brain to the input image's brain, and it provides the outcome of performing this registration and its inverse, i.e. the input's brain registered to the template space. ANTs implements 9 kinds of possible registrations, from only shifting the image to even deforming it to achieve the most exact match.
3. Brain segmentation: the template's tissue segments (also known as "priors") are registered to the input extracted brain, then, the segmentation is improved using *Atropos* and *N4* before separating the input's tissues in various images. Note that the resulting brain segments ("posteriors") are therefore not registered to the template.
4. Registration to a template: this step is optional, but if performed, the previous brain registration (stage 2) is slightly improved and the resulting cortical thickness image will be registered to the template space.
5. Cortical thickness: the cortical thickness is extracted using *Diffeomorphic registration based cortical thickness (DiReCT).* This technology takes advantage of the GM, WM and CSF segmentations to find the surfaces corresponding with the WM - GM and GM - CSF interfaces to estimate the thickness volume, bounded by both interfaces [6].
6. Quality control: generates mosaic-like images representing the cortical thickness and the segmentations along image slices.

Figure 3 shows an illustration of this process from the hands of the ANTs' developers, published in [16]. It reviews the pipeline on a deeper level, but the MRI scans portrayed should provide a good overview.

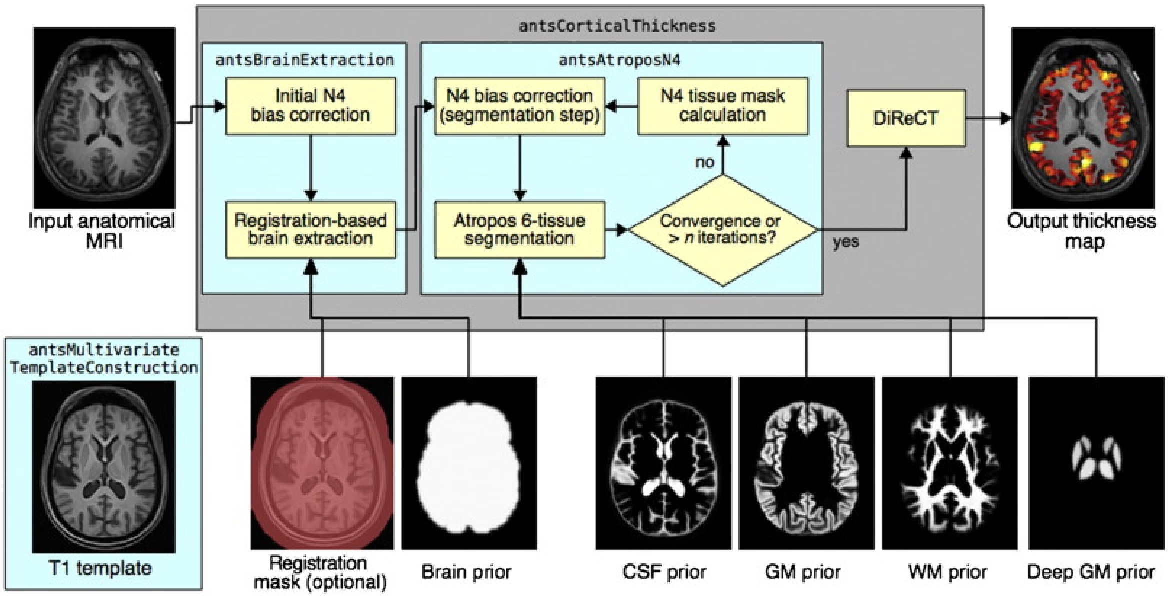

antsCorticalThickness
antsBrainExtraction
Initial N4 bias correction
Registration-based brain extraction
antsAtroposN4
N4 bias correction (segmentation step)
N4 tissue mask calculation
no
Atropos 6-tissue segmentation
Convergence or > n iterations?
yes
DiReCT
Input anatomical MRI
Output thickness map
antsMultivariate TemplateConstruction
T1 template
Registration mask (optional)
Brain prior
CSF prior
GM prior
WM prior
Deep GM prior

Figure 3: ANTs cortical thickness pipeline. Source: [16].

3.2. Quality control

The added numerical quality control system is based on outlier detection techniques. Whilst no universally accepted definition of "outlier" has been agreed upon, this thesis will understand an outlier as a data sample that appears to remarkably stand out from the group of samples that it belongs to [7]. The opposite of an outlier is commonly referred to as an "inlier". According to literature, the extension of outlier detection techniques can be grouped in 4 classes [8], discussed next.

3.2.1. Statistical methods

Statistical methods use the dataset distribution to find samples that noticeably deviate from it. This can be done, for example, by looking for low value bins in the dataset's histogram. According to literature, these techniques struggle with high amounts of data or high dimensional data, but otherwise are experimentally efficient when the probability distribution is given.

3.2.2. Distance-based methods

Distance-based methods rely on computing distance metrics to find the furthest data points and classify them as outliers. While independent from the data distribution, these methods still do not perform remarkably with high dimensional data.

3.2.3. Density-based methods

These kinds of methods calculate each point's local density (how crowded its surroundings are), so an outlier's density is expected to differ from its neighbors. A common strategy regarding density is Local Outlier Factor (LOF), that can detect outliers locally relative to clusters rather than relative to the whole set of data. Even if these methods are proven to perform better than distance-based methods in some cases, they still struggle with high dimensional data when computing density, and the number of nearest neighbors to compare densities to is a parameter to be defined.

3.2.4. Cluster-based methods

These techniques define data clusters iteratively, in a way that a cluster of a given size surrounds a relatively dense crowd of data points. The outliers are those points that do not fall in any cluster, i.e. they are not interpreted as a part of a crowd. A common technique is *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)*, that can generate clusters of arbitrary forms and identify noise (non desired values) in low density clusters. Whilst they do not require previous data distribution information and can identify local outliers, they have to be configured with a non negligible amount of parameters such as number of points in each cluster, number of clusters, target metric to improve each iteration, etc.

3.3. Brain region segmentation techniques

To extract thickness and volume data from particular sections of the brain cortex, a previous segmentation into regions of interest is required [A3]. To do that, at least a brain atlas (brain region map) is required to be referenced, but the most common approach is to jointly use multiple reference atlases [9].

Before computing the actual segmentation, the reference atlas images have to be registered to the input's space. Even if this is a necessary step to get the best segmentation, it represents the "computational bottleneck" of the whole process, so it is of great interest to avoid this step, if possible, by using a common space.

Another critical decision is the number of atlases to use. Although it is recommended to use a variety of references for better results, it is expected for the process' computational cost to be at least linear with respect to the number of used atlases, so detecting and putting aside the less significant ones is remarkably relevant when applied in time sensitive tasks. Literature [10] demonstrates various methods to programmatically calculate each atlas significance for a big atlas database.

The main process uses the selected, registered atlases to propagate their labels to the input image and join them to obtain the best possible segmentation. There are many methods to perform the joining step, the simplest ones being using only the best atlas and choosing the best fitting label from each atlas. Contrary to the "best atlas" method, the second strategy called "majority voting" does not disregard input from other images, but it does not use intensity information. Several variations of these methods along with more complex procedures have been implemented in the previous years, and many of them yield better results than the two first mentioned if correctly applied in suitable contexts, but they also imply a higher computational cost. However, recent research proves that simpler methods yield good results, especially if applied to brain MRI scans, at a lesser resource cost.

3.4. Linux process resource consumption

Linux offers a vast variety of ways of displaying process resource consumption information. Nevertheless, owing that many of these methods require root permissions (recall that said processes ran in external computing services) and that the required metrics are limited, a significant amount of options become unavailable or just excessive. Therefore, the literature review consists of exploring which are the most suitable Linux commands regarding the stated goals and limitations, which offers next to no discussion on account of the specificity of the task.

In the light of the above, the literature for this section consists of the manuals corresponding to the reviewed commands. Said manuals can be accordingly found in [A6].

3.5. Brain age prediction

The proposed Machine Learning system for this project to perform the brain age prediction is XGBoost or "eXtreme Gradient Boost". This algorithm is broadly used in ML tasks due to the fact that it can be remarkably parallelized, shortening the processing time that many other systems would take [13]. Recent research proves how ML can be relied on to extract biomarkers such as the brain age through MRI scans [14]. In particular, XGBoost has been previously used in similar studies in which gray matter is used to predict brain age on the UKB MRI dataset [15], and the results demonstrated the idea to be feasible under these and other circumstances.

The model is based on assembling various decision trees, optimizing them to minimize a target metric (objective function). A decision tree is a set of nested questions that split the input data into smaller groups. Each question asks for a particular feature of the data, and it divides the input set in two groups depending if their features satisfy or do not satisfy the question's statement [A4]. A single tree can be built on any set of data features and algorithms exist to optimize the questions to ask in order to improve the data classification. It can be used as a classifier by itself, but the model tends to overfit to a dataset, which leads to scarce extrapolation capabilities.

To tackle this, many different trees can be put in common to mitigate overfitting: this model is known as a random forest, and the final prediction is an aggregation of the individual trees' decisions. XGBoost makes an extra step and adds an objective function to perform gradient descent algorithms that iteratively improve the trees' structures. The objective function is usually the prediction residuals from a tree, i.e. the difference between the prediction and the ground truth label. In this case, the residuals of one tree are fed to the next one to minimize the final residuals, instead of being fed the data samples. Even though the trees cannot be built in parallel with these settings, each one of the trees' branches can be parallely computed to minimize the computation time.

To sum up, XGBoost uses the ensemble of multiple decision trees (boosting) whose branches can be built in parallel ("extreme") to perform joint classifications with the other trees and iteratively minimize the classification's residuals (gradient descent optimization).

4. Methodology / project development

The following figure represents what the outcome of this project aims to be, illustrating concepts developed in the previous sections:

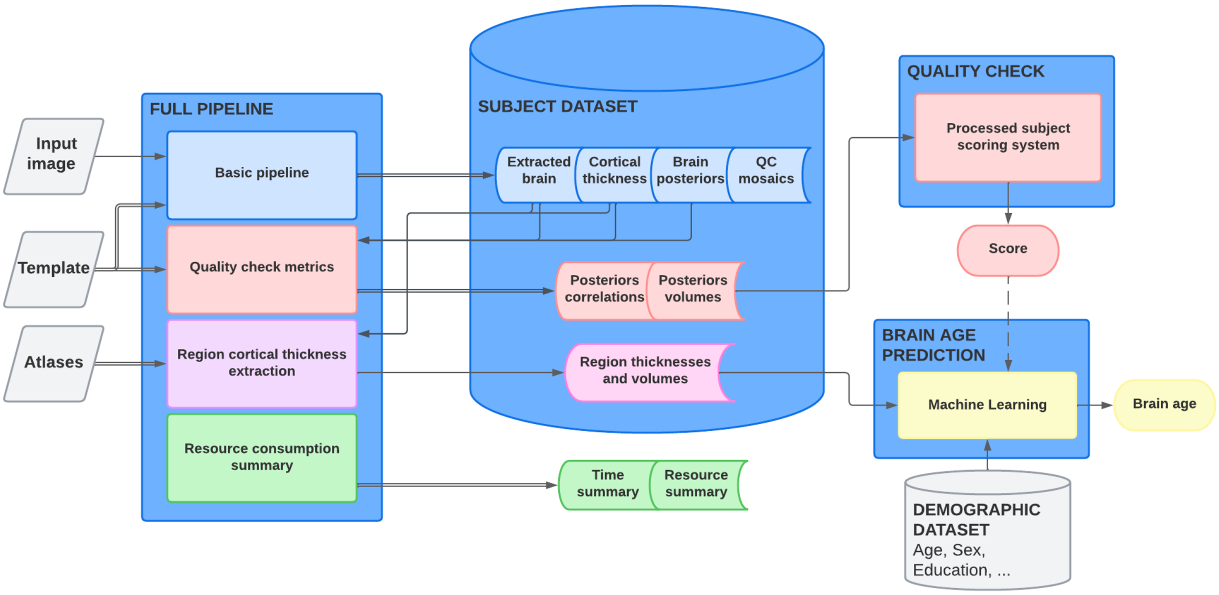


Figure 4: Project features breakdown.

In summary, a pipeline reads a raw MRI scan and, using the ANTs basic pipeline (light blue) relying on a template, processes the input to obtain useful images such as the cortical thickness and brain posteriors. This data is then used by the next stages to extract the quality check metrics (red), using the same template as reference, and to obtain the volume and thickness values for each brain region (magenta) using several brain atlases as a reference. This information is added to a subject dataset, and the computational performance of the pipeline is assessed by the resource consumption module (green), yielding the processing time and the resource usage summary.

After the dataset has been completed, the quality check metrics are used in a scoring system that points out the subjects that have been wrongly processed and hence might hinder the training of a ML system. Also, a ML model uses the regional cortical volumes and thicknesses along with the demographic information of the corresponding input to train a system able to predict the subject's brain age. Ideally, the scores should be used to put aside any doubtful subjects, but the interface to automatically do so has yet to be implemented.

Each one of these modules is not only strongly intertwined with each other, but also has a vast variety of parameters that can considerably alter its outcomes. In addition, performing the full pipeline is remarkably time consuming (logging a total of 7 hours for 3 parallel subjects in the optimum case) and the used systems are based mostly on novel technologies. As a consequence, the majority of resources have been put on performing multiple tests to obtain the best possible results. To optimize testing the pipeline, the following approach was followed:

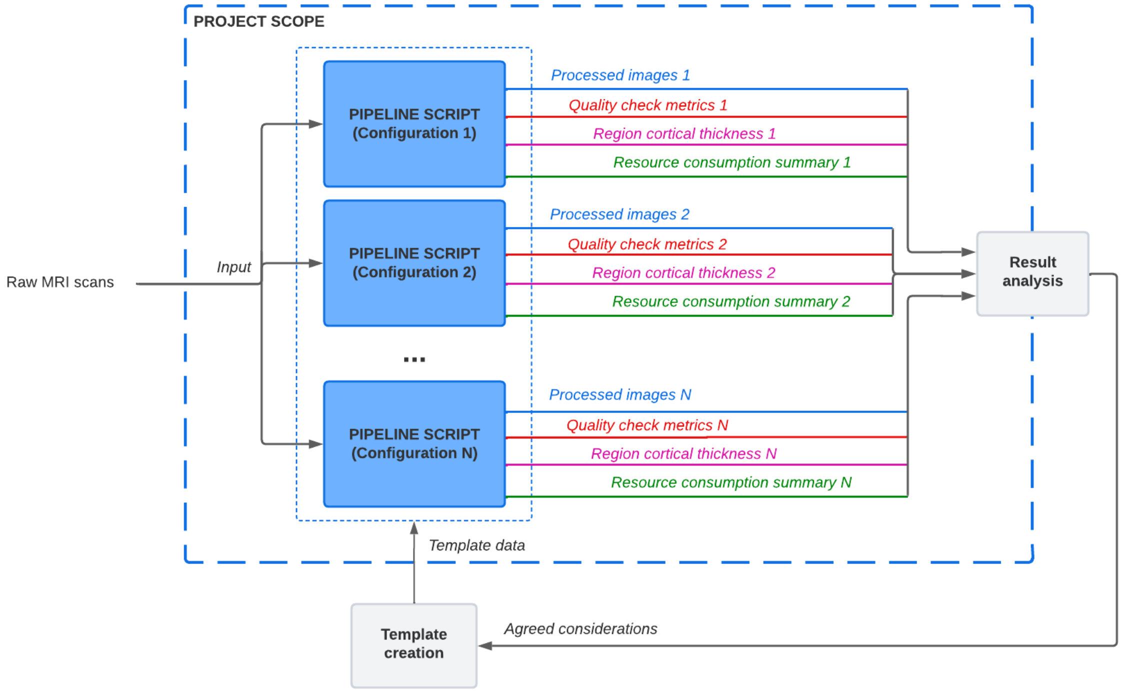

PROJECT SCOPE
Raw MRI scans
Input
PIPELINE SCRIPT (Configuration 1)
PIPELINE SCRIPT (Configuration 2)
...
PIPELINE SCRIPT (Configuration N)
Processed images 1
Quality check metrics 1
Region cortical thickness 1
Resource consumption summary 1
Processed images 2
Quality check metrics 2
Region cortical thickness 2
Resource consumption summary 2
Processed images N
Quality check metrics N
Region cortical thickness N
Resource consumption summary N
Result analysis
Template data
Template creation
Agreed considerations

Figure 5: Pipeline development methodology.

With this strategy, many versions of the pipeline (configurations) were put to test in parallel. Each configuration had its unique features: the pipeline could process the subjects sequentially or in parallel, it might run only a few of its modules or use a different set of parameters for each one of them, among many others.

Once the tests were finished, the various outcomes were jointly analyzed with the BBRC (note that the “Result analysis” block lies partially within the project scope). Then, a new template was created and the configurations changed satisfying the resulting considerations. At this point, it is important to note that selecting the input data and creating the templates is not encompassed by the project’s scope; this task is performed entirely by Irene Cumplido from the BBRC.

This approach unfolded several difficulties, for example, the amount of resources provided by the computing services limited how many configurations could run in parallel, in addition to setting a maximum storage boundary to save the various results. Also, testing the quality check and the brain age prediction required a good amount of processed subjects, which may quickly become obsolete if a new batch happened to be better. Therefore, improving these systems could not be done consistently over time: the best possible pipeline configuration had to be selected to process the biggest possible amount of data, taking into account the project’s time span.

The particularities of each one of the mentioned modules will be discussed in the next sections.

4.1. Dataset

Before proceeding with the modules' explanations, the dataset used to test the different systems will be briefly described. It was obtained from the UK Biobank and consisted of T1-weighted structural magnetic resonance images, all collected using a 3T Siemens Skyra scanner with a voxel size of 1 x 1 x 1 mm3, with the scans lasting 5 minutes each. The image sizes were 208x256x256 voxels and they were formatted as NIFTI. The whole dataset consisted of 44,183 participants ranging from ages 44 to 80, and the scans were taken in 2014. However, only subsets of these participants were used at various parts of the project due to storage limitations. The particularities of each subset are defined further on.

4.2. Basic pipeline

Recalling the previous explanations on the topic, the basic pipeline is performed by ANTs using a 6 stage process that, relying on a reference template, obtains useful images from a raw MRI scan such as intensity corrected, skull extracted brain, the brain's cortical thickness, 6 brain tissues' segmentations (Figure 1) and quality control images, among others. The following figure illustrates these images.

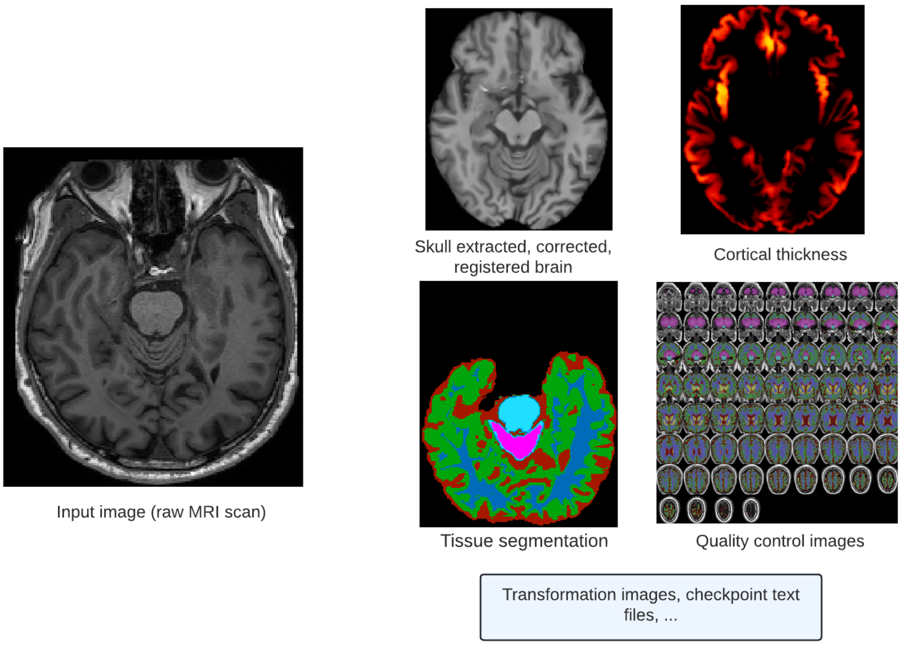

Skull extracted, corrected, registered brain
Cortical thickness
Input image (raw MRI scan)
Tissue segmentation
Quality control images
Transformation images, checkpoint text files, ...

Figure 6: Basic pipeline input (left) and output (right). Only a slice of the brain is shown in all the images except in the segmentation mosaic, which shows various slices from the same view.

This module's results depend mainly on the template selection, but other configurations can be added to accomplish different objectives. To list the most important ones:

- Unnecessary outcome discard: ANTs outputs a vast amount of images and other metadata that are unnecessary for this project, for example, an image representing the shifting that the brain has undergone to get registered to the template. As a matter of fact, the computing services storage filled up quickly if these images were kept. The solution was to remove this data as soon as it was not further needed, by identifying which files compounded the unnecessary archives and at which point they were no longer of use.

- Image normalization and binarization: it is usual for each sample in a dataset to range the same intensities, even in images, where the range is defined by the minimum and maximum pixel values. To do that, ImageMath's (part of the ANTs' toolkit) function *Normalize* can be used to limit the desired image intensities' span from 0 to 1. In addition, it can be useful to obtain a binarized version of an image, for example, to use it as a mask. ANTs provides the utility *ThresholdImage*, that can binarize an image if used as:

  ThresholdImage <dimension> <input> <output> 0 0 0 1

After several experiments, the best configurations to execute the pipeline were found. Firstly, the optimum way to run the pipeline was processing a number of subjects in parallel. This strategy decreased the total processing time in exchange of the computational resources used. As computing services are usually designed to perform computationally demanding operations, they should be able to meet the pipeline's resource requirements to a certain extent. Be as it may, the batch size -or number of subjects that are processed at the same time- can be regulated to comply with the limitations of any machine.

In terms of storage, the basic pipeline had to dispose of the files that were not strictly necessary to avoid flooding the machine's available memory. Those unneeded files include all text logs and all images representing transformations and warps. Therefore, the basic pipeline only saves the brain's extraction mask, the 6 posteriors, the extracted, intensity corrected, registered brain and the registered cortical thickness image. Numerically, if the pipeline did not implement this option, each subject's resulting folder would weigh about 830 MB, i.e. an output / input ratio equal to 44, but otherwise the folders weigh around 57 MB achieving a 3 output / input ratio.

Finally, some of these preserved images are modified to improve their usefulness in future steps. In particular, the extracted brain and the cortical thicknesses are normalized so their voxels fit the range [0, 1], and the cortical thickness is binarized into a mask that will be used to parcel the cortical thickness into ROI. The non processed brain and cortical thickness images are kept in case future operations are required.

This is the core of the project, but it does not provide all the results needed to fulfill the agreed requirements. Therefore, various features need to be added, and each one of these will configure the pipeline's behavior to achieve specific objectives.

4.3. Quality check

After the images undergo the basic pipeline, an automated quality control stage is required to detect wrongly processed images and prevent them from being fed to a ML system. This was approached by firstly developing a classifier that discarded outliers from a set of processed data, and then fine tuning the system to rate the segmentations according to how good they were performed.

To tackle this, it was chosen to assess a variety of outlier detection methods using a selected set of metrics, to finally end up using the better performing ones. This approach required labeling the output dataset based on individual visual examination, deciding whether a segmentation was “correct” or an “outlier”. Considering this was a really subjective approach, many of these decisions were corroborated by the BBRC to avoid mislabeling, but there were some clear cases of bad segmentations such as the following:

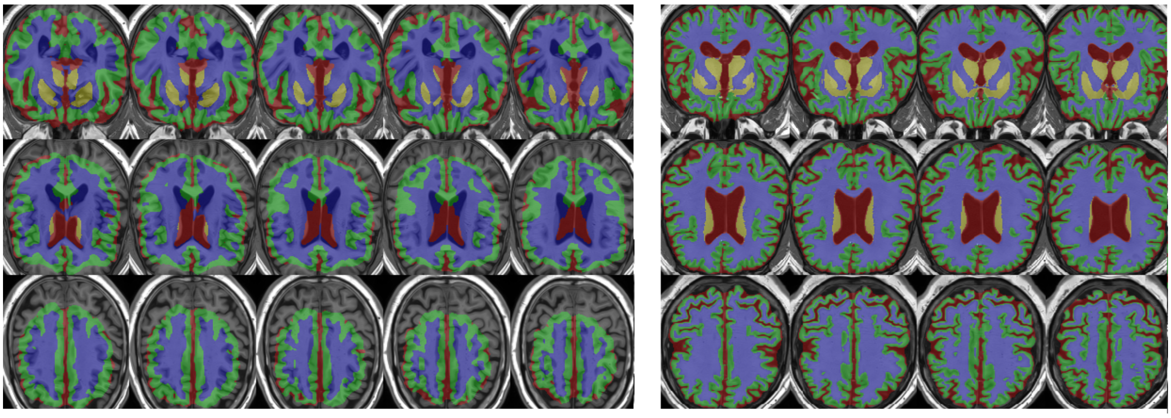


Figure 7: “Outlier” segmentation (left) vs “correct” segmentation (right). Both images represent the same slices of two subjects, with the grays representing the original raw image and the overlaid colors representing different kinds of tissues (Figure 1).

In the first image, a remarkable amount of tissue is left unclassified. Not only that, but the ventricles (that appear as a dark ‘X’ in the middle of the image) are partly misclassified as WM when they should be CSF. Therefore, the left image can be considered an outlier.

Other types of outliers exist, typical examples being images where the skull has been classified as WM and the background is labeled as CSF (see Figure 8). In any case, the data extracted from the GM (that makes up the brain cortex) would be wrong and therefore the subject should be discarded.

Once the problem was defined, two questions remained: how to transform the images to data suitable for outlier detection methods and which methods to use.

4.3.1. Metric gathering and selection

Under a purely statistical point of view, an outlier is an abnormal value that stands out from a set of data. Therefore, extracting the adequate type of data from the images (i.e. obtaining a dataset that truly makes the outliers stand out) is crucial for any detection system to work properly.

Considering the number of possibilities that sprung up from that idea, a few considerations and constraints had to be made to assess only the best strategies, using information gathered by visually analyzing many segmentation results:

- The first indicator of a bad segmentation is a segment having an unusually small or large volume (as commented after Figure 7).
- Even with the above, inter-subject comparison through corresponding segment volumes may not be accurate owing that the subjects’ brain sizes might be different, or that some of the subjects can show a noticeable level of atrophy. Thus, a correctly segmented brain with large or small proportions should be identified as a correct segmentation: outlier classification should be independent of the brains’ particular shapes and volumes.
- The segments (or tissues) that usually get segmented the worst are GM and WM. Also, GM is the most critical segment when extracting the regional cortical thickness and volumes.

- On the other hand, many cases present good labeling of some segments such as the brain stem whilst wrongly identifying GM or WM. This sustains the idea of an intra-subject segmentation assessment, i.e. comparing a subject's potentially correctly segmented tissue to a mistakenly segmented one.

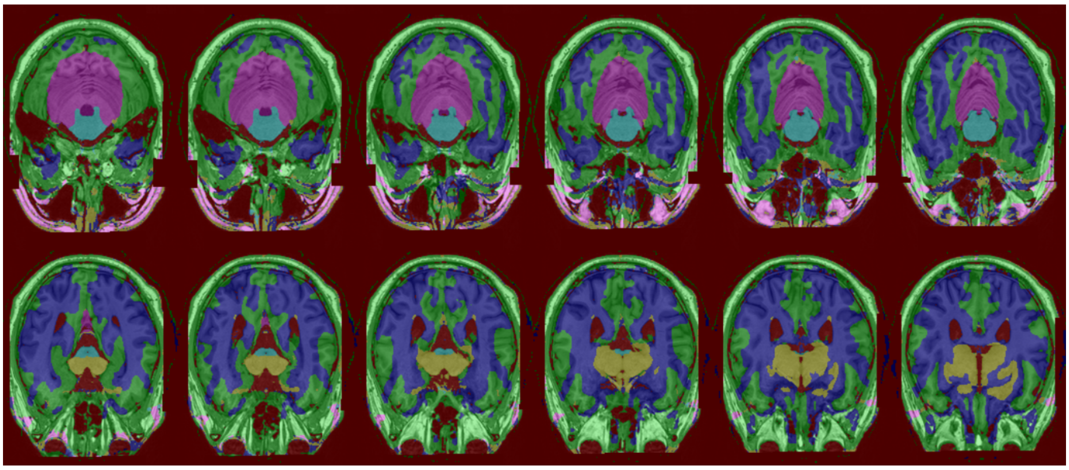


Figure 8: Segmentation slices where the brain stem (cyan) has been correctly identified but the background has been labeled as CSF, among many other mistakes.

- Many subjects are mistakenly segmented due to the computed brain extraction mask being too shrunk (the segmentation is only computed over said mask - see Figure 7), therefore, an unusually small mask could be a good indicator for an outlier.
- Not only a tissue's inaccurate segmentation is characterized by having an unusual volume, but also its shape should not correlate with its analogue template segment. Nevertheless, the subjects' segments are not registered to the template space, though it is necessary to extract faithful correlation values. This unfolds questions regarding which registration method to use. Intuitively, the registration that most truthfully portrays the segments correlation should be one that does not escalate nor deform the volumes, but only translates and rotates the section to match the template's space.
- Brain templates are usually blurry (low pass filtered) to give more importance to shapes (low level information) rather than details (high level information). Mistakenly segmented regions, especially GM, do not contain details such as the brain sulcus, so the shape information predominates in the segmentation. This way, an outlier segment may better correlate with its analogous in the template. To avoid this bias, it might be a good idea to filter only the low level information for a given segment in each subject.

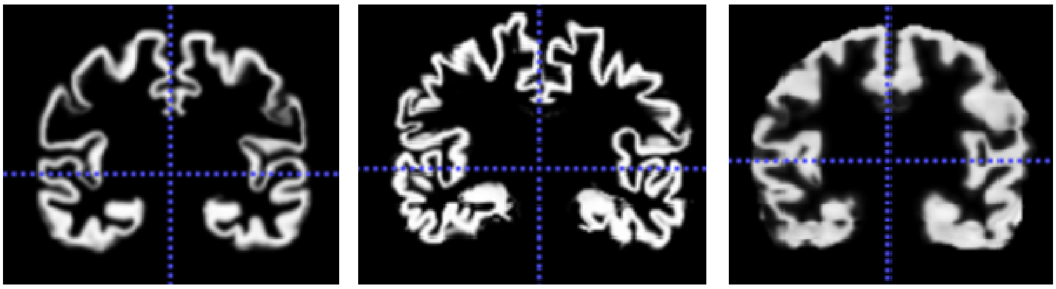


Figure 9: Same slice from the same view of a template's GM (left), correctly segmented subject's GM (middle), and outlier's GM (right). The correct subject presents more high-level details than the template and the outlier, but in terms of shape the outlier could correlate more with the template.

- ANTs provides a toolkit to obtain image volumes and correlation metrics which should be easily added to the pipeline. Also, the computational cost and time of each data extraction method should be taken into account.

The following metrics and data extraction methods were found to address the hypothesis and fit the constraints.

4.3.1.1. Posterior correlations

The rationale of these metrics is assessing a segmentation comparing a subject segment's shape to its analogous in the template, therefore using the latter as a "ground truth". ANTs offers two main correlation functions: *PearsonCorrelation* and *Mattes*. The first one calculates the Pearson's correlation and the second one computes the mutual information (MI) between images. It is important to note that while Pearson's correlation ranges from 0 to 1 and identifies high similarity with values near and to 1, the mutual information computed by ANTs is a negative number that reflects high correlations with lower values [A2]. Taking into consideration that the GM and WM segments are the most affected by a bad segmentation, these correlation metrics were computed comparing GM and WM against the template's analogous regions. Nonetheless, a registration between subject and template segments is required to do that, and even if the hypothetical best registrations to do that have been discussed -i.e. those that do not deform the volumes-, all of the 9 registrations in the ANTs toolkit were used at first instance to avoid mistakenly discarding options right away.

Moreover, ImageMath's operator *G* was used to apply a (Gaussian) low pass filter to the subject's segments in order to extract the low level information, minimizing the previously mentioned bias. This operator allows configuring the filter with a certain sigma that increasingly makes the segment blurrier, and sigmas ranging from 0 (meaning not filtered) to 4 were used for each registration method.

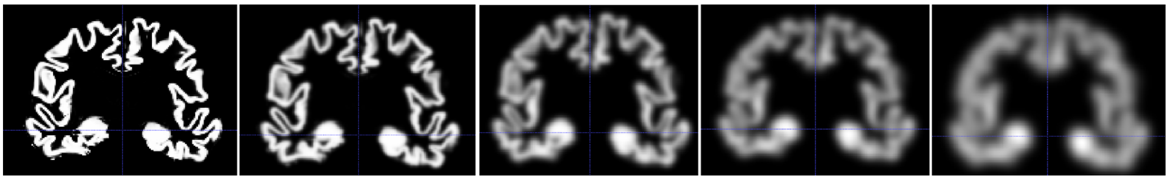
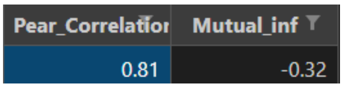


Figure 10: Same slice of a template's GM filtered with increasing sigma from 0 (left) to 4 (right).

To sum up, each segment (GM/WM) was low pass filtered with a sigma ranging from 0 to 4 before being registered in one of the 9 available ways to the template's analogous. This resulted in a total of *2*5*9 = 90* Pearson's correlations and mutual information pairs for each subject. Hence, a subject's result file for a given segment, registration and sigma ended up looking like the following figure:



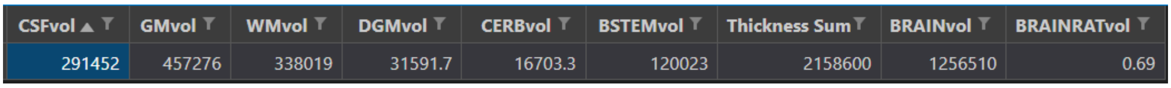
Pear_Correlatio	Mutual_inf
0.81	-0.32

Figure 11: An example of a subject's extracted correlation metrics.

4.3.1.2. Posterior volumes

Correlation information quantified a segmentation based on the shape of its segments, but it required steps such as registering and filtering that added up to the total processing time. As this could not be done extensively, it was found adequate to find a faster, more straight forward metric such as segment volumes.

Volumes can be computed for each segmented tissue by using ImageMath's *total*, which sums up the values of all voxels in an image. Bearing in mind that, for this project, the voxel intensities in a segment all were equal to one and that each voxel had a size of 1 x 1 x 1mm3, the outcome of this function was the volume of said region. Besides this, other metrics were computed such as the whole brain volume and the extracted cortical thickness' volume, in addition to the ratio between the subject's and the template's brain volumes. The latter was useful to identify atypically small brain masks, resulting from a bad brain extraction and therefore contributing to a bad segmentation. A subject's file containing this information looked as follows:



CSFvol	GMvol	WMvol	DGMvol	CERBvol	BSTEMvol	Thickness Sum	BRAINvol	BRAINRATvol
291452	457276	338019	31591.7	16703.3	120023	2158600	1256510	0.69

Figure 12: An example of a subject's extracted volume metrics. The units are cubic millimetres.

4.3.2. Quality check dataset creation and definition

Quality control systems are often tested in a complete, priorly made dataset. As this project consists of processing the best dataset possible over time, obtaining the best processed subject pool in time to test these systems was not an easy task. To mitigate this, the best datasets obtained at different points in the project were kept, so they were not processed under the same pipeline configurations. Even if this had an impact on the quality check module's robustness, the most useful subset were selected time limitations considered.

In the light of the above, the subsets of subjects that were used to test the quality check were:

- A subset of 35 participants obtained with an early version of the pipeline, containing 30 correct subjects and 5 outliers. This subset was used to extract all the possible metrics, and the results of that were useful to discard any outstandingly bad strategy for metric computation. The rationale behind trying all the metric obtention approaches on this batch was based upon processing time: the registrations required a vast amount of time depending on their complexity. As a note, extracting all the metrics in this dataset took 4 days running in the optimal conditions.
- A subset of 106 participants obtained with an improved version of the template, containing 100 correct subjects and 2 outliers. Only the metrics that achieved the best results on the previous subset (alluded further on) were extracted from it due to the mentioned time constraints.

Note that these datasets are far from optimal for a classification problem: the ratio between the two classes -correct and outlier processed subjects- is not too even (class imbalance), not to mention both are obtained through different templates. However, they proved to be useful to perform the following experiments.

4.3.3. Outlier detection methods

Once retrieved the most significant metrics, the next challenge was creating classifiers that took the most advantage of them. Different approaches were followed to tackle this issue, based on state of the art techniques on similar problems.

4.3.3.1. Distance-based methods

This strategy groups up a common metric from all subjects and finds the most distant ones. As the obtained metrics are 1-dimensional, the most distant metrics are the ones with the most extreme values, either high or low. To detect outliers in this scenario, a threshold that splits outliers from correct subjects can be defined, dividing the range of values in two zones. This classifier works ideally if all the values in a zone correspond to outliers and vice versa.

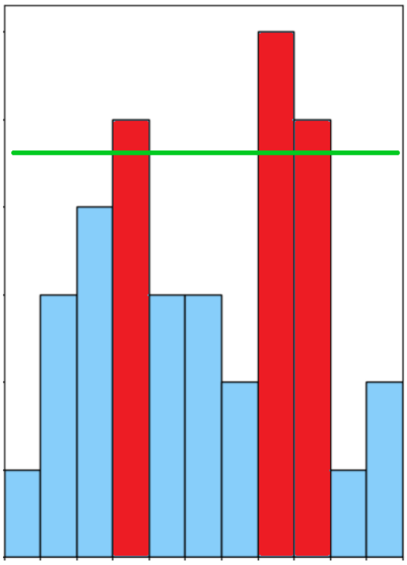


Figure 13: Representation of an ideal distance classifier. Blue values correspond to correct subjects, red values correspond to outliers and the green line illustrates the threshold. All the outliers values belong to the upper zone and all the correct subjects belong to the one underneath.

4.3.3.1.1. Correlations

Grouping each subject correlation's information, a set of data for each triplet of "compared segment, applied registration and low pass filter sigma" was obtained for both Pearson's correlation and mutual information correlation metrics. For example, the data group obtained from registering GM using an affine registration and a low pass filter with sigma = 4 can be illustrated as follows:

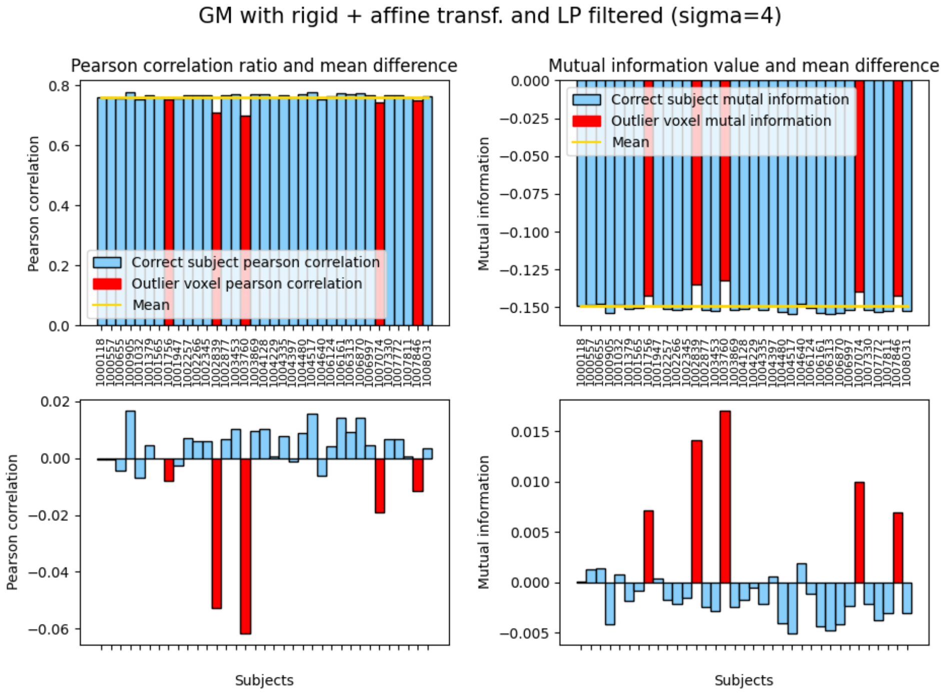


Figure 14: Pearson's correlation and mutual information computed on the gray matter of each one of the 35 batch dataset's subjects, using an affine registration and a Gaussian filter with sigma = 4. The top left plot shows the Pearson correlation for each subject in addition to a gold line representing the mean value. The plot underneath illustrates the difference between each value and the mean, to have a better view on the differences. The plots in the right column represent the same data but with mutual information.

In this case, the parameter triplet is "[posterior = gray matter, registration type = affine registration, sigma = 4]". From now on, a parameter triplet will refer to a certain combination of these 3 elements (therefore, there can exist up to 2*9*5 = 90 triplets).

Also, in this context, the "distance" is inferred from a metrics' value relative to the values of all the others in the batch. To illustrate this, the bottom plots represent each subject's correlation metric distance to the mean of the set. However, this distance was not the one used to classify outliers: the classifier worked directly on the metric's numerical values.

4.3.3.1.2. Volumes

For volumes, a similar examination was made. Nevertheless, contrary to the correlations, volumes did not show any clear tendencies when compared to each other: the distribution seemed to be random. Figure 15 shows the results that better suited the classifier under examination, but neither a distance-based method or statistical method seemed to work on these metrics, so this approach was discarded.

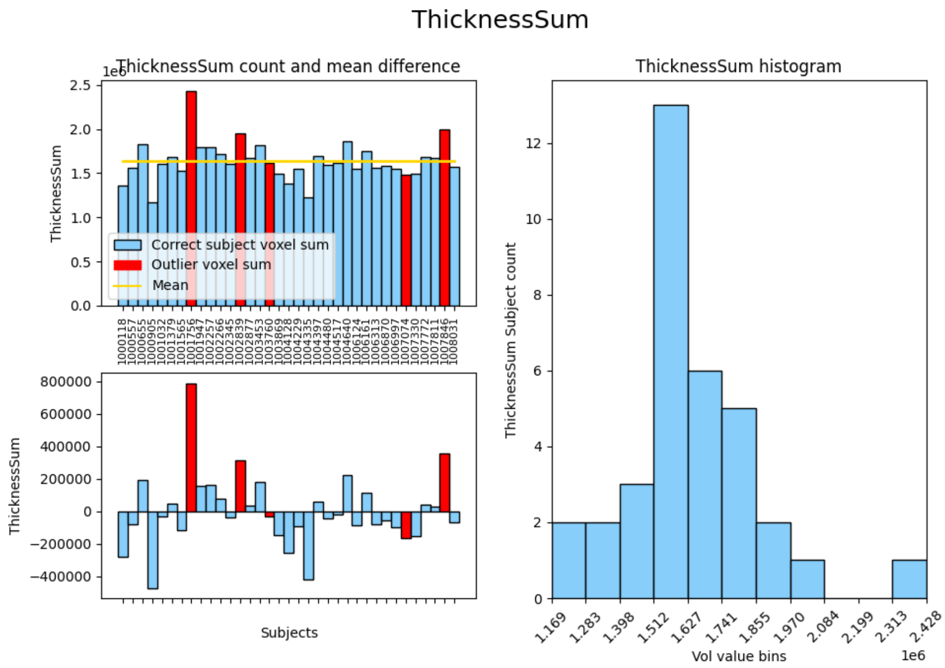


Figure 15: Cortical thickness volumes from the small batch of 35 subjects. The left side displays the same information as in the correlation case. On the right side, an histogram of the values is shown.

However, to take advantage of the volume information, other strategies were explored by pairing the volumes into 2-dimensional metrics and creating a point cloud where each point was a subject's volume tuple. This approach contributes to robustness by taking into account more information about the subject rather than relying on just one tissue's volume. For instance, a subject can have a correctly segmented brain stem but a wrongly identified GM. While the brain stem volume should not stand out by itself and the GM volume could be interpreted to be part of an unusually large or small brain, neither of these metrics would identify the subject as an outlier. Nevertheless, pairing the volumes, future classifiers should be able to see that such GM volume is uncommon for that brain stem volume, potentially improving the chance for the classifier to correctly label this subject.

Also, It is important to point out these kinds of systems perform better with large datasets, so the 35 subject batch was discarded to test this strategy. Consequently, all the initial experiments regarding 2-dimensional metrics were made with the 106 subject batch.

In the light of the above, the following 2-dimensional distance-based method was tested:

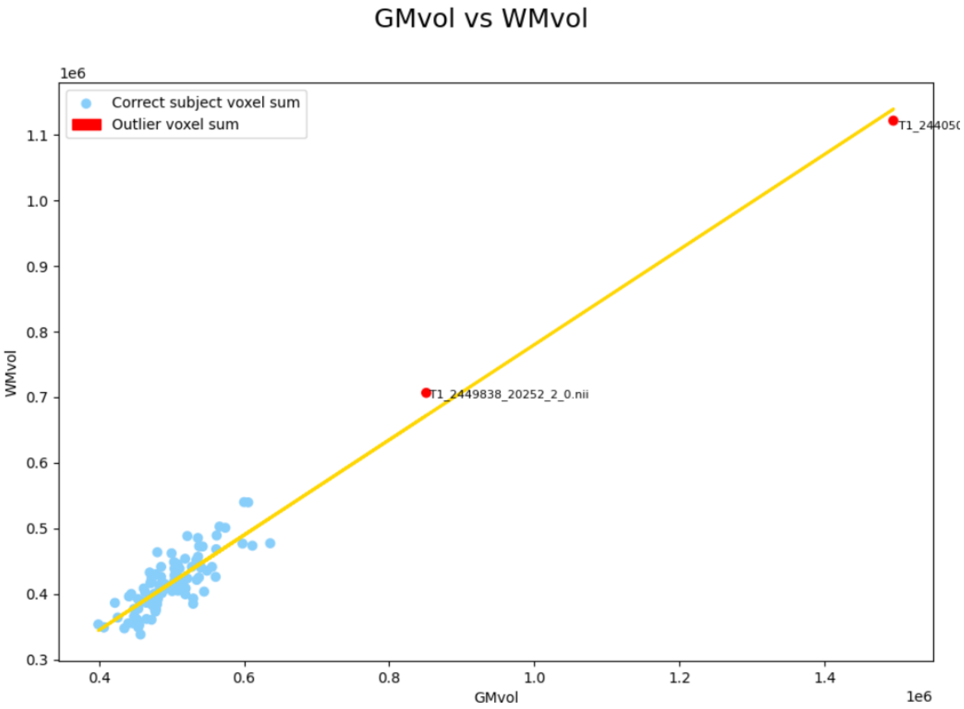


Figure 16: GM volume vs WM volume scatter plot with the computed regression line (gold). Outliers are highlighted in red and labeled.

The method consisted on identifying an outlier by computing its distance to the groups' regression line. Although it performed well in some cases, the same outliers contributed to the regression, so some cases as Figure 16 were given where the outliers' distance to the line were shorter than the correct subjects' ones. Thus, this system was discarded and other more robust methods were explored in order to take advantage of the volume information.

4.3.3.2. Density-based method: LOF

This strategy requires a point cloud or N-dimensional scatter plot. Based on how crowded the surroundings of a data point are, a score (known as Local Outlier Factor) is given. If the point's neighborhood is really crowded, it obtains a result near to 1 and, in the opposite case, it obtains a higher value, higher the score the more isolated it is. However, this system requires comparing each point with a fixed number of data samples. This fixed number is known as “k-neighbors” and it is a hyperparameter to be considered when implementing these kinds of systems. Also, the threshold at which a LOF score is classified as an outlier is hard-coded into the module implementing the algorithm, so this is not a hyperparameter to be configured. Finally, it is important to point out that this algorithm returns negative scores, but their absolute values hold the adequate meaning.

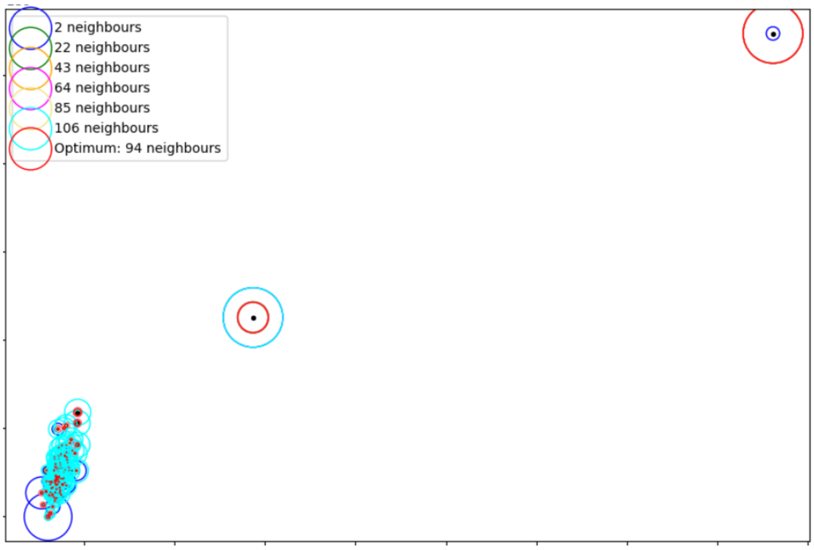


Figure 17: Representation of the LOF system. The circumferences surrounding each data point (black) represent the LOF score, so higher radiuses mean higher scores. Each circumference color represents scores from different k-neighbors, ranging from 2 (minimum for this system to work) to 106 (total number of points). The optimum value for k-neighbors (red) gives the higher scores to the 2 furthest points, assigning the higher score to the most extreme one.

At first instance, this approach proved to work better than the 2-dimensional distance-based one, so more experiments were conducted on it using multiple volume pairs.

4.3.4. Metrics and methods overview

To summarize the ultimately used outlier detection metrics and methods, the following illustration has been added to provide an overview of the whole process:

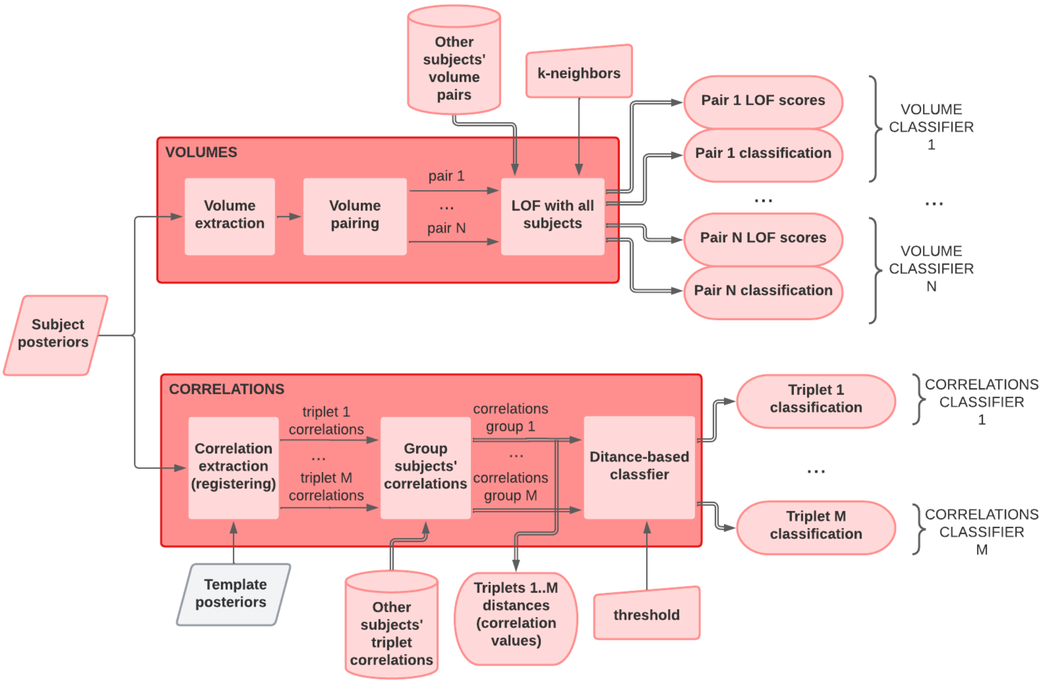


Figure 18: Quality check metrics and methods overview.

The volumes and correlations extraction modules process the subjects' posteriors to convert them into useful metrics. The volumes module uses LOF to rate all the subject's pairs of volumes for each metric tuple, and the correlations block uses the Pearson correlations and mutual information to check the most extreme volumes for each parameter triplet. Hence, each subject's initial metrics are combined or processed to obtain the following possible classifiers:

- Volumes: starting from 9 volumes, each subject can end up with up to 36 unrepeated volume pairs to classify using LOF.
- Correlations: starting from 2 correlations (Pearson and MI), each subject can end up with up to 2 possible tissues * 9 types of registrations * 5 sigmas = 90 possible triplets for each type of correlation to classify using distance-based methods.

That adds to a total of 126 classifiers that can be used to identify outliers. Nevertheless, a way of putting the outcomes of these classifiers together has not been defined yet.

4.3.5. Outlier detection to result scoring

The final goal of this module was to rate the processed subjects and obtain a score based on the results. At this point, the explored methods outputted a classification and a numerical value: the Pearson correlation and mutual information for the correlations, and a LOF score for the volumes. Note that the correlations method does not need any classification to provide numerical values, whilst the volumes method needs to undergo the LOF classifier to obtain numerical values (scores) and a classification. Therefore, the k-neighbors hyperparameter is needed to translate volume pairs into scores, while no translation is needed for correlation metrics.

Therefore, scores representing the outlierness of a subject were available for each possible outlier detection method. However, not all of the methods could be relied on: it has been shown that some of them perform a better job at classifying than others. Fortunately, metrics for classifier evaluation such as F-score [A5] can be computed. Now, not only every method outputted a subject's outlierness score, but there was a number able to quantify the reliability of the method when classifying.

As the F-score is a metric obtained after training a classifier with labeled data, the final scoring system had to implement a "training" stage. Under this perspective, the scoring system can be seen as analogous to a typical ML system. It was approached as such:

1. Create a "train" stage with a number of labeled subjects. Sweep through all the possible classifier hyperparameters and find the ones (thresholds for the distance-based method and k-neighbors for density-based one) that maximize the classifiers performance, keeping the associated maximum F-score. At this point, the system holds the best F-scores and hyperparameters for all the possible classifiers.
2. Start the "test" stage and compute each subject's numerical scores using the previously computed hyperparameters for the LOF system. In addition to the F-scores for each classifier, the system has obtained the numerical outlierness score for each subject.
3. Normalize the scores' range. This is necessary to translate the numerical values to a common system that can assess all of them equally. To do this, Pearson

correlation (usually ranging from 0.6 to 0.85) and mutual information (ranging from -0.1 to -0.15) are linearly expanded to fit a [0.1, 1] range.

$$\mathit{norm\_vals} = (((\mathit{vals} - \min(\mathit{vals})) * (1 - 0.1)) / (\max(\mathit{vals}) - \min(\mathit{vals}))) + 0.1$$

Note that this normalization assumes that the best processed subjects always have high correlations, and worst correct ones -not outliers- should have lower values even if not identified as outliers. By the other hand, LOF score, logging values around -1 for correct subjects and equal or less than -30 for outliers is translated into [0, 1] by dividing 1 by the absolute value of the score.

$$\mathit{abs\_score} = 1/\mathit{absolute\_vale}(\mathit{score})$$

This way, each classifier yields a numeric score ranging from 0, meaning probably outlier, to 1, meaning most likely a correct subject. All the intermedium values should be higher the better the processing has performed.

4. For each subject, sum their individual scores weighted with their respective best F-scores obtained in the training phase. Add the volumes and scale the range from 0 to 100. Therefore, the final score has been computed using each system's scores weighted by their performance as classifiers.

This process can be illustrated in the following image:

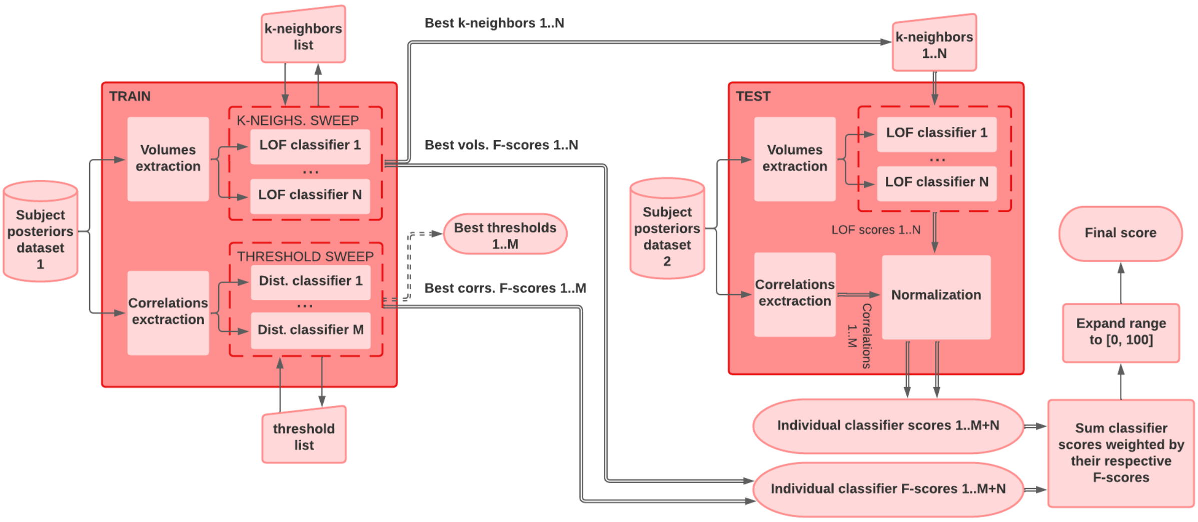


Figure 19: Quality check scoring.

4.4. Region volume and cortical thickness extraction

A system that scores the segmentation results has been defined, but there is yet no way of transforming the basic pipeline's results into numbers suitable to feed the ML system: the various cortical volumes and thicknesses have to be extracted from each cortical region. To do this, a brain cortex segmentation into ROI is required. Then, the volume and thickness of each ROI can be estimated using the most suitable tools.

4.4.1. Brain cortex segmentation

The first thing needed when segmenting a brain is a brain atlas, or a map that limits and labels each ROI. In this project, it was a requirement to use the *Desikan-Killiany-Tourville atlas* (or DKT), consisting of 31 ROI for each brain hemisphere [A3].

To perform the segmentation, ANTs offers the script *antsJointLabelFusion*, that creates a segmentation from multiple previously segmented brain cortices. This complies with the literature since most state of the art researches claim that MRI cortical segmentations into ROI perform better when using multiple segmented images as reference, also known as "Multi Atlas Label Fusion" (MALF). The reference images were obtained from the broadly used MindBoggle dataset [12]. The images taken from this dataset were 20 manually segmented cortical volumes, all of them in the MNI space.

There are two important settings to consider when optimizing the results yielded by *antsJointLabelFusion*: how many reference images to use and which mask to apply to distinguish background (in which the segments are not computed) and foreground. However, the most critical decision was choosing the input image to be segmented: while running the algorithm with the cortical thickness image seemed like the obvious thing to do, using the extracted brain as the input demonstrated to work significantly better, probably because the system wasn't propagating any mistakes that could have happened when extracting the cortical thickness. Regarding the mask selection, ANT's script offers ways to automatically compute it as well as options to add a mask image. If the last is chosen, it is important to consider that the mask has to be in the same space as the image to be segmented.

Finally, it should be taken into account that MALF's output consists of an image containing the various ROIs, but the voxels bounded by each region are all equal to ROI's label ID, so each one of the 62 regions has a unique representative value. These numbers can then be mapped to their corresponding label names using MindBoggle's data.

4.4.2. ROI volume and thickness extraction

Once the segmentations have been performed, the next step is to find a way to estimate their volumes and thicknesses. Substantial research on the ANTs' toolkit did not provide any straightforward method of doing that, but a workaround could be implemented by linking a few of its tools:

1. Use ImageMath's *LabelThickness* on the resulting segmented image to obtain another image where the voxel values (intensities) belonging to a region are all equal to the estimated region's thickness, instead of being equal to the label's ID.
2. Use ANT's *ImageIntensityStatistics* on the last image to find the mean intensity value of each ROI, which equals the ROI's thickness value since all the intensities in a region equal the thickness in the input image. This script requires an additional input to map each region's thickness to its label ID, so the image resulting from the segmentation can be used for this. Running this will output a text matrix where each row is a ROI's label and each column is a feature including mean intensity (i.e. thickness). This text has to be saved in a generic file.

3. Use ANT's *LabelGeometryMeasures* on the image resulting from the segmentation to obtain a csv where each row is a region's label and each column is a geometrical feature, including the ROI's volume.
4. Using a parsing script that reads from the text file obtained in step 2, the csv obtained in step 3 and the brain volume csv obtained in the quality check module, join all the relevant values into a final csv and map the label's ID to its name. Therefore, each subject should end up having one csv where each row holds a ROI and the columns are the region's ID, the region's name, the region's volume, the region's thickness and the subject's brain volume (or Total Intracranial Volume - TIV).

4.5. Linux process resource consumption

The resource consumption log is used both to find the optimum configuration of the pipeline and to fulfill one of the requirements imposed by various computing services, in which an estimation of the cost of running a code is required before truly running it. Some basic metrics are commonly expected: execution time, RAM usage, CPU usage, GPU usage and memory occupation. To do this, Linux offers a variety of options, but few of them are compatible with the computing services that this project is built on. On that account, the most suitable way to obtain these and other metrics is designing a solution from scratch that fits inside said boundaries. This design can be summarized as follows:

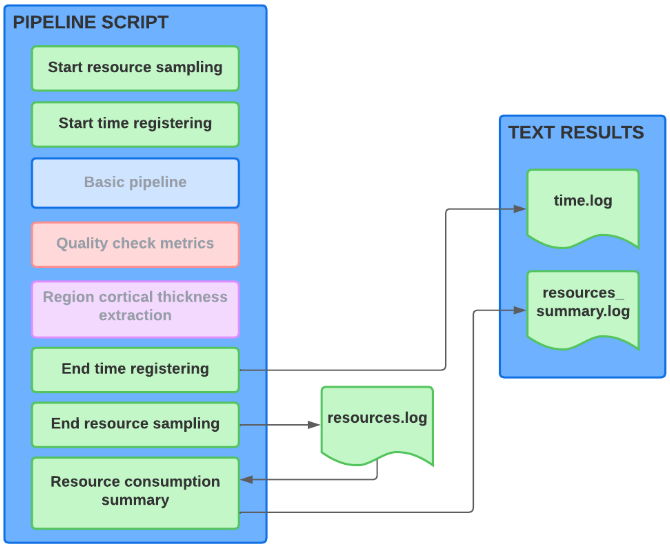


Figure 20: Resource consumption extraction diagram.

The left box represents the pipeline script in a sequential breakdown, unfolding the various interventions of the resource usage module. The right box holds the results of the extraction: a time log file that registers how long the various parts of the pipeline last and a resource log that contains a summary of various consumed resources.

The resource usage module uses Linux commands to register both the time and the resource usage. The most complex part is registering the usage, which is done by using Linux's *top* command periodically in time. Each period, this command samples the process that runs the pipeline script and saves the resources that it is consuming on a text file (*resources.log*), appending the sample each time. Please see [A6] to check the list of the metrics that can be obtained through *top*. At the end, when the other modules are finished, Linux's command *ps* [A6] is called to obtain a new CPU metric that is not included in the *top* command. Next, a parser registers and counts all of the *top* samples, identifies the various fields, saves their values into their according metric categories and finally performs an average over each metric. The parser formats the metrics with their corresponding magnitudes (size [MiB], time [s], ratio [%], …) and writes down their averages in addition to a brief explanation of their meaning into a new text file (*resources_summary.log*). The parser also adds the metric obtained with the *ps* command in its corresponding category. Finally, the program calculates the input (raw images) and the output (folders with several images and text files) size along with the output / input ratio and adds it to the final file.

In addition, information about the resources consumed by the process is displayed by the computing services' terminal once the process (in this case, "job") has finished. Even if there was not a way to add this information programmatically into the resources file, it was manually added because it holds new important data. Finally, a summary of the CPU characteristics is appended to the final resources file using Linuxs' *lscpu*, so the computing services that receive this information can have a deeper insight on the CPU metrics.

Time-wise, the module registers the time that each other module lasts using the *SECONDS* Linux variable at the start and end of each part, computing the elapsed time as the difference of the two. Then, the program saves this data into the *time.log* file, in addition to the total processing time.

This is a general explanation of the resource usage module, but another version was implemented in which the resources could be found at a subject level rather than a general one. In this scenario, the resource samples are grouped by subjects so the averages show how individual subjects performed, which can provide another potentially useful insight of the pipeline's resource usage.

4.6. Brain age prediction

Although the XGBoost script was provided by the BBRC due to the fact that it had been used before to test similar metrics on other pipelines, some modifications had to be done to interface the current outputs of the pipeline and the inputs of the model. To read and adapt the pipeline's output, a data integrity function was implemented.

This function first reads the subject's files containing their ROI metrics, checking if the data is correct and otherwise discarding the subject. Then, it transforms the data to match a matrix where each row is a subject and each column features each region's volumes and thicknesses. As the model also profits from demographical data, the demographics file (provided by the BBRC) has to be read and saved as a matrix where each row is a subject and each column holds a demographic characteristic such as age and sex. Nevertheless, the list of subjects with available demographics may not match the list of subjects with regions' information, so an inner merge has to be done between the demographics and the ROIs matrix to end up with only the subjects whose entire data is available.

This data suits what the XGBoost model expects as input. With these samples, the model can be assessed using a cross validation system. This system divides the data into a number of equal parts (or number of “folds”) and splits each partition into train and validation subsets. The model uses typical values for this, with 10 folds and a 90%-10% train-test ratio. Each one of these partitions trains and validates the model using as a target each subject’s real age to obtain their own validation metrics. At the end the mean and variance of the desired evaluation metrics are returned. In this case, the evaluation metrics are Pearson r coefficient (“r”), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) [A7].

5. Results and discussion

Many experiments were conducted to test the systems described in the Methodology. The results of these experiments will be discussed along this section.

5.1. Basic pipeline

After the optimum way of running the pipeline was established (as stated in the methodology), many experiments were conducted trying different templates to obtain the best possible processed images.

Ultimately, the main template used to execute the pipeline was the one created from 413 UKB subjects in the MNI space, 206 women and 207 men ranging from 45 to 80 years. However, this template did not have the usual MNI volume and, as a consequence, a second registration with the MNI template was used to escalate the previously registered subjects to the MNI space.

Other experiments were conducted where only a single template was used, but this was created using UKB subjects and then registered to the MNI space in various ways to correct its volume. This strategy was discarded for two main reasons: first, even the registrations that do not deform the image made the UKB template lose relevant data. Second, the optional registration was required anyways to obtain a cortical thickness image registered to the template's space, since this was necessary for the brain cortex parcellation step that occurred later on, so using just one template would not imply any improvements regarding resource usage.

5.2. Quality check

5.2.1. Posterior correlation through a distance-based classifier

In the batch containing 35 subjects, in which the correlation metrics were extracted using all 90 methods (parameter triplets), the best performing specifications were:

- Posterior: GM.
- Registration type: affine and rigid.
- Sigma: 3.

In particular, the methods that used combinations of these three parameters performed perfectly in the classification stage, unequivocally splitting outliers and correct subjects. This is coherent with the previously mentioned hypothesis that sprung up from visual examination: GM is the most affected tissue in bad segmentations, the best registrations should be the ones that do not deform the volumes (affine translates, rotates and linearly escalates; rigid translates and rotates) and low pass filter contributes to best classification removing possible biases. Also, results were better overall as sigma increased, although at sigma = 4 some classifications started to worsen, as can be seen in the next example:

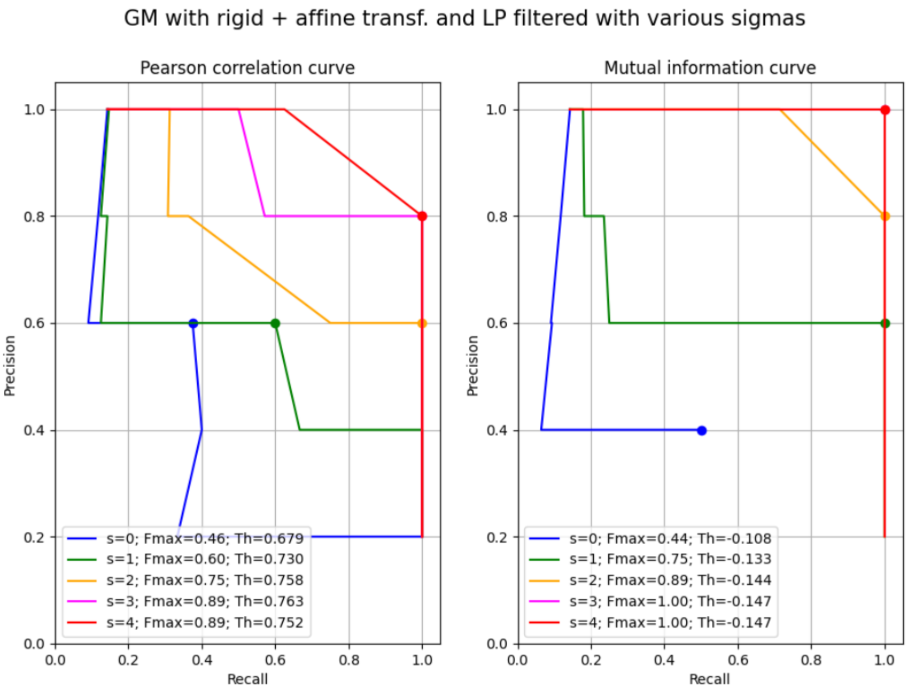


Figure 21: Precision-recall [A5] curves of the proposed classifier applied on Pearson correlation and mutual information with one tissue, one kind of registration and various sigmas. Each point shows the precision and recall value for a given threshold, and the marked ones highlight the threshold that provided the best F-score (Fmax) for every sigma. Bottom left, these point's F-score and associated threshold can be seen. Note that the plots do not quite resemble typical precision-recall curves due to the fact that there is an important class imbalance.

In consequence, more tests were performed using rigid and affine registrations with GM and sigma = 3. In particular, experiments were conducted using these methods to extract metrics from the 106 subjects. Results showed that, with this batch, using the affine registration yielded ideal results (perfect classification) while using the rigid one only classified 1 outlier as such (see [A8]). As this batch only includes 2 outliers, the outcome of the previous batch was taken into more consideration and therefore both registrations were kept for experiments yet to come.

5.2.2. Posterior volume through a LOF classifier

As previously mentioned, all the experiments regarding this density based classifier were performed on the 106 subject batch. The results showed that all the pairs portrayed the outliers as isolated points:

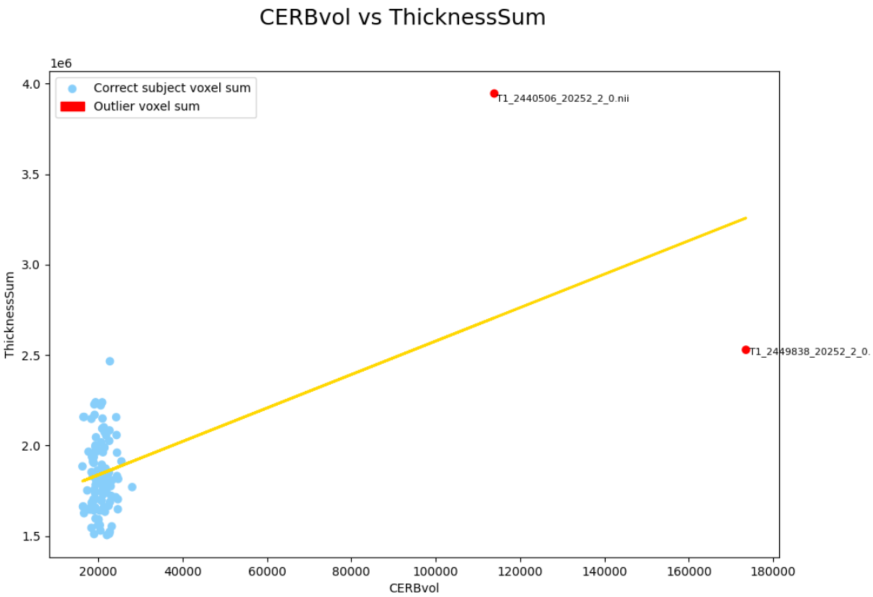


Figure 22: Cerebellum (axis X) vs cortical thickness (axis Y) volumes. Highlighted samples point out outliers. Regression line drawn as a reference.

Under these circumstances, the LOF algorithm was always capable of detecting all the outliers with a given k-neighbors hyperparameter, therefore, each assessed pair in this batch had a F-score equal to 1.

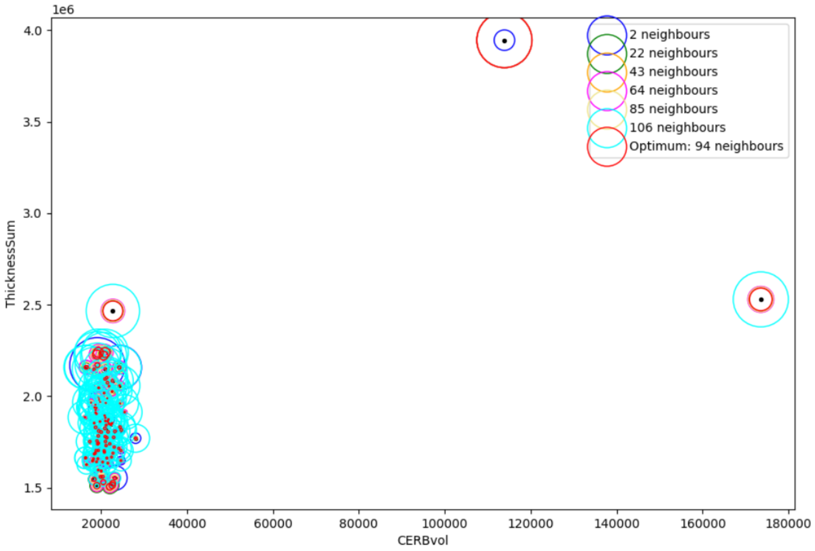


Figure 23: Cerebellum (axis X) vs cortical thickness (axis Y) volumes with their LOF score represented with circumferences, greater the score the bigger the radius is. Each color plots the results for a different k-neighbors hyperparameters.

5.2.3. Outlier detection to result scoring

To conduct experiments on this system, the 106 subject batch was used as a train dataset and a new 297 subject batch, consisting of 12 outliers and 285 correctly segmented subjects, was used to test the system. The 12 outliers had scores ranging from 7.64 to 27.37 out of 100, while correctly segmented subjects scored over 54.83 points and up to 95.6. However, lowest and highest scoring correctly segmented subjects were examined and they did not seem to present many differences regarding quality, which leads to believe that a more complex segmentation involving more regions could be more precise when identifying anomalies.

On the other hand, this experiment demonstrated the extrapolation capacity of the model. With a labeled dataset of 106 subjects, the QC system was capable of successfully detecting outliers from a new 297 subject batch by using hyperparameters extracted from the labeled batch. Therefore, this system should correctly identify outliers coming from new batches, provided that these subjects are all processed in the same way.

5.3. Region volume and cortical thickness extraction

Tests on the previously mentioned variables influencing this module were conducted. Regarding the number of atlases to use as reference in the cortical parcellation, experiments pointed out that the results were better with an increasing volume of references, although the computing time rose with it (logging about 15 minutes for a single atlas and 3 hours for 20). However, after about 20 atlases the results stopped improving.

Examining the different masking options allowed by the ANTs' script, the best experimental results were yielded by using the binarized cortical thickness as a mask, even if this was not the optimal solution timewise. The processing time of all these options tested under the same conditions are listed below:

- *majorityvoting*: 35 minutes
- *otsu*: 1 h
- *or*: 3 h 40 minutes
- Cortical thickness: 1 h
- Binarized cortical thickness: 50 mins

In conclusion, ANTs' *antsJointLabelFusion* script was configured using the extracted brain as an input, 20 DKT atlases in the MNI space and the binarized cortical thickness image as the mask. The whole module lasted around 3 hours when used on a single subject, and the posterior stages where the thicknesses and volumes are extracted were computationally negligible. Nonetheless, this can be optimized by taking advantage of a script`s feature that allows to process each atlass in parallel, inevitably in exchange for an increased resource usage. This feature was not used in the pipeline because it surpassed the used computing services' limitations.

5.4. Linux process resource consumption

As portrayed in Figure 20, this system was able to correctly summarize all the available resource usage metrics into two text files. The final tests were performed on a 3 subject batch so the resource consumption could be extrapolated for bigger batches. Some files resulting from these tests can be seen in [A9]. As expected, parallel processing significantly reduced the total processing time in exchange for an increase on the consumed computational resources, that should be affordable by a majority of computing services.

5.5. Brain age prediction

A few experiments were made to test the brain age prediction system on the extracted data. The used batch consisted of 297 subjects where 12 of them were outliers. After using the data integrity function in this batch, there were a total of 202 subjects with all the information (region volumes and thicknesses + demographics) available to feed the XGBoost, but 6 of them remained outliers still. The 95 subjects that were discarded was due to the fact that there were no demographics associated with them.

To check the usefulness of discarding these subjects and, therefore, evaluating how segmentations affected the performance of the model, two tests were made: one in which all the 202 subjects were used and another where the 6 outliers were discarded, leaving 196 samples to train the model. The dataset descriptions can be found in [A10]. The following figures illustrate scatter plots of the real ages against the predicted ones, including evaluation metrics, for the two experiments.

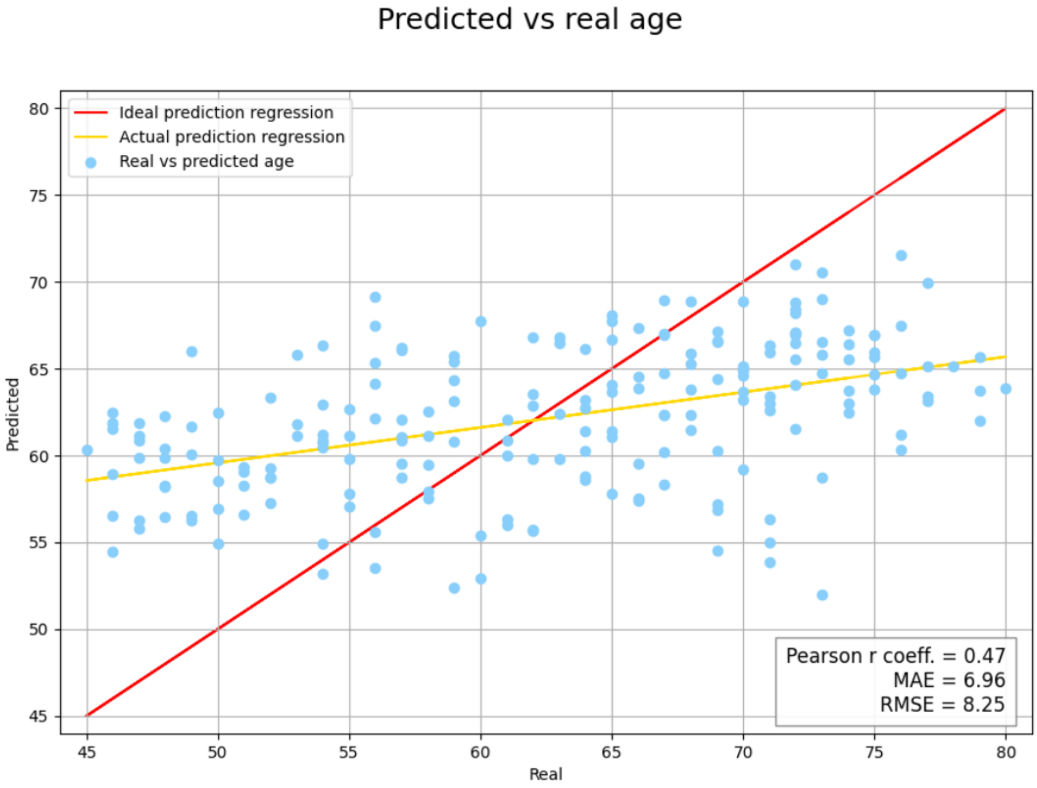


Figure 24: Prediction with 202 subjects: 196 correctly segmented and 6 wrongly segmented ones. The blue dots represent the real age (axis X) - predicted age (axis Y) pairs. The yellow line illustrates the regression computed on these points and the red line shows the direction in which the points should have clustered to obtain an almost ideal prediction. In the bottom right corner, a patch shows the calculated evaluation metrics for these points.

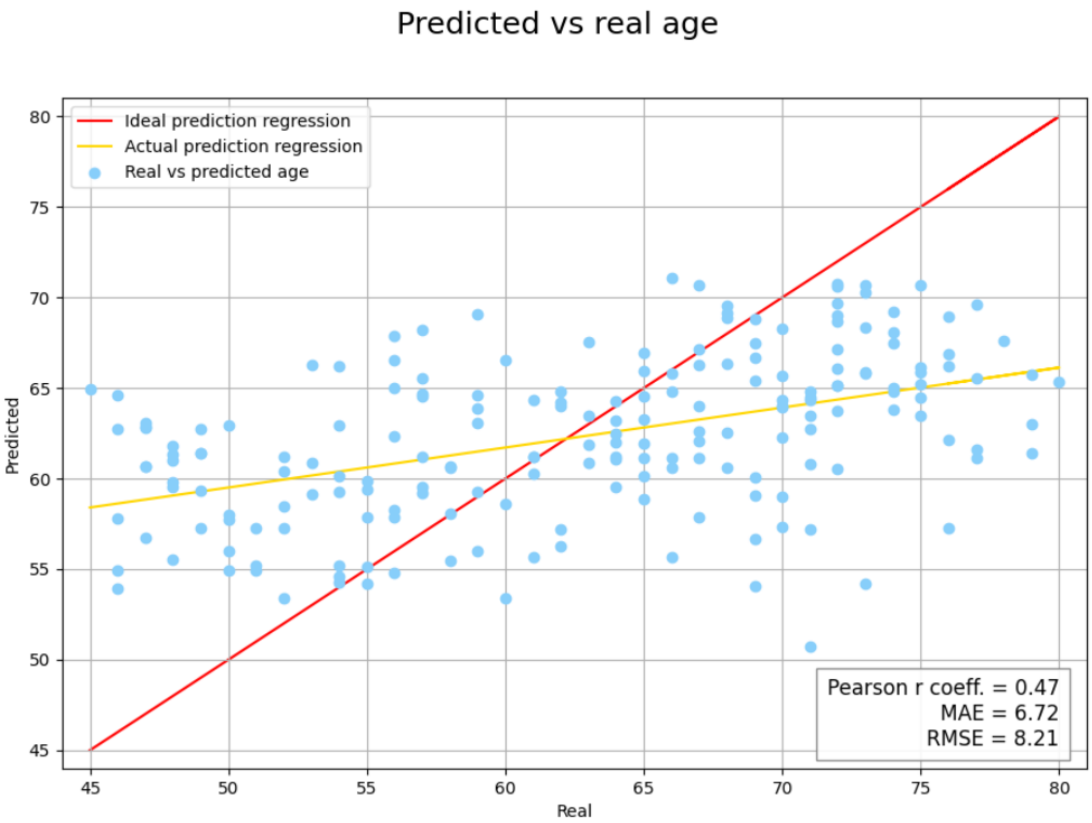


Figure 25: Prediction with 196 correctly segmented subjects.

The model trained with the 196 correctly segmented subjects dataset seemed to perform better as indicated by the MAE and RMSE, however this improvement shows that a good previous segmentation into tissues does not have great influence in the brain age prediction. As mentioned in the methodology, the ROI’s thicknesses and volumes are extracted from the skull-stripped brain image using the binarized cortical thickness image as a mask. In this process, the only thing linked to the previous tissue segmentation is the cortical thickness image, as it relies on the GM and WM tissues. As a consequence, it can be inferred that, whilst the cortical thickness ended up being the best mask to perform the brain parcellation, it seems to have little influence in the overall performance.

To finish with, the same experiment was performed on analogous data extracted with Freesurfer from the same 196 subjects, as one of this model’s goals is to assess different structural data extraction methods. The results can be seen in Figure 26.

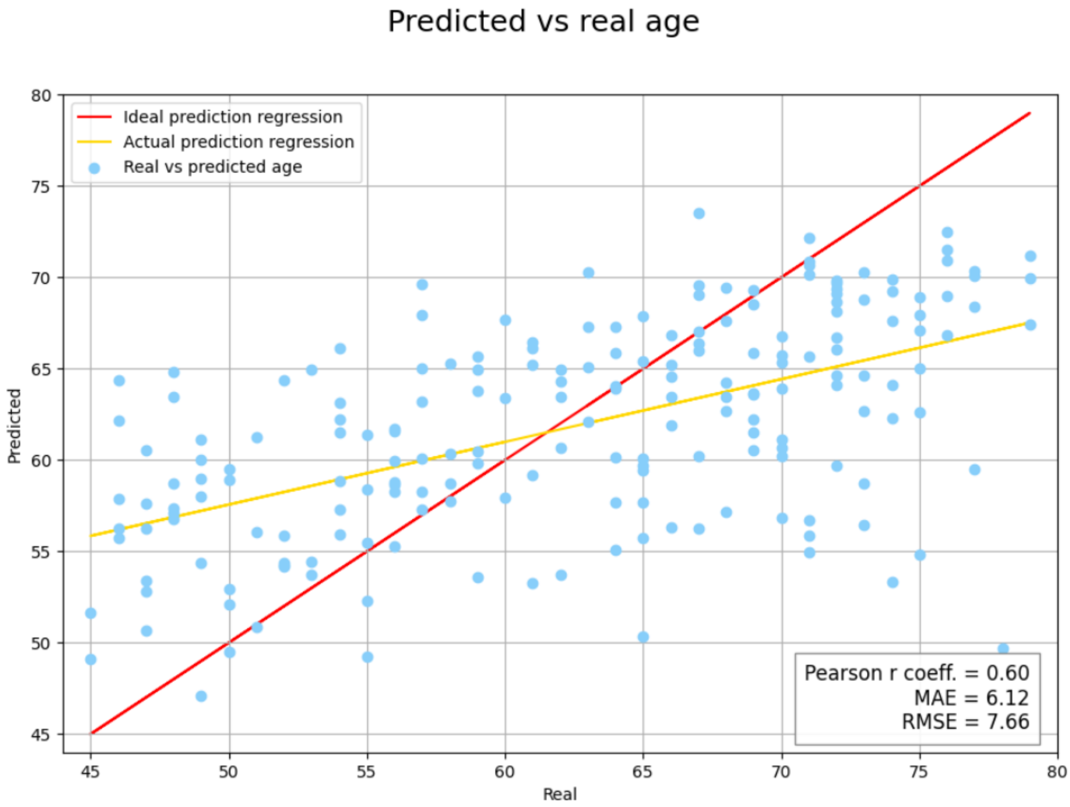


Figure 26: Prediction with the same batch of 196 subjects but processed with Freesurfer.

With this number of subjects, the Freesurfer processing pipeline seemed to perform better. Research [16] shows how ANTs yields better results under a similar context: using a random forest classifier with DKT regions' thicknesses and 40 subjects from different datasets, the age prediction's RMSE registered lower values with ANTs data. Nevertheless, it has to be taken into account that, for this project, the XGBoots model's hyperparameters were optimized to fit the Freesurfer data, so this presumably biases the results. As the goals of this thesis include using a model to compare ANTs and Fressurfer's results under the same conditions, no tests were made optimizing the model to fit the ANT's data. More research has to be done to develop a model that yields the most truthful, unbiased results.

6. Budget

The budget has been computed using direct costing and considering that the project's time extension equals 19 weeks. In the human costs, it has been assumed that the dedication of this project has been equal to the ECTS hour conversion (18 credits * 25 hours/credit), and that the advisors have been consulted approximately one hour a week along the project's duration plus one introductory meeting previous to that. The resource costs have been based on Microsoft's Azure computing services pricing[2]. In the light of the above, the cost breakdown has been estimated as follows:

HUMAN COSTS				
Position	**Occupied positions**	**Salary/hour**	**Total Worked Hours**	**Position cost**
Student	1	10.00€	450	4,500.00€
Advisors	2	50.00€	20	2,000.00€
TOTAL	**6,500.00€**			

Table 1: Budget human costs.

RESOURCE COSTS				
Asset	**Number of existences**	**Cost/hour**	**Used hours**	**TOTAL**
Calcula computing services	1	0.794€	2928	**2,324.83€**

Table 2: Budget resource costs.

DEPRECIATION						
Asset	**Cost**	**Useful life (years)**	**Residual value**	**Year deprecation**	**Week deprecation**	**TOTAL**
Laptop	800.00€	5	100.00€	14.00€	2.69€	**51.15€**

Table 3: Budget depreciation costs.

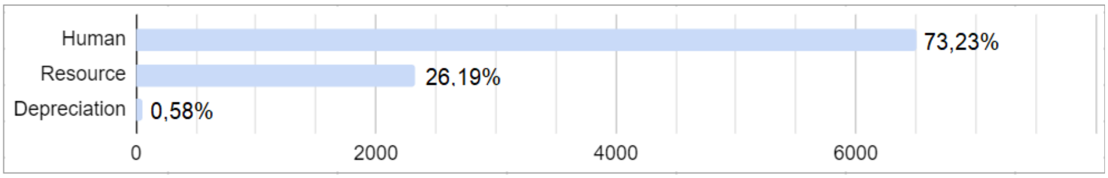


Figure 27: Budget distribution: category (Y) vs cost (X).

Adding these three categories, the project's budget estimation adds up to **8,875.98 €**.

[2] https://azure.microsoft.com/en-us/pricing/calculator/

7. Environment Impact

Environmental sustainability has been recognized as one of the challenges that systems using big dimension datasets (such as Machine Learning and related) have to face to comply with the modern world's needs and contribute to a healthy technological development. Therefore, a sustainability report should be added to projects using these technologies to gain awareness, explore sustainable alternatives and divulge good practices to the scientific community. This section will focus on environmental sustainability, whether the reported impacts are positive or negative.

Environmentally speaking, the main difficulty when working with Machine Learning comes from the fact that these kinds of systems usually require a vast amount of computing resources, either required to train them or to preprocess the training data. As the hardware running these processes require energy proportional to their computing capabilities, it is inferable that their electricity consumption translates into carbon footprint when bearing in mind that obtaining the energy implies an environmental impact. Taking into account that this project relies on external computing services, not many choices were available to minimize the energy consumption. Nevertheless, for other related projects that might be able to choose where to run their experiments, it is advised to choose computing services minding their sustainability reports, minimize their use and shut them down when not being used in order to mitigate the environmental impact.

In addition, working in teams also leaves a carbon footprint: sending data -especially audiovisual content- consumes energy resources from the server hosting the interchange. To mitigate this during the development of this project, it has been prioritized to exchange only the necessary data, avoiding redundancy and, when possible, transferring information physically without relying on any cloud. For projects with a bigger scope either in content, duration or team size, it is recommended to perform a previous research on data transferring methods and even establish a protocol to avoid unnecessary exchanges, for example, by working on a shared, remote server rather than having each team member download the data individually.

By the other hand, this project has a positive impact by creating a dataset suitable for many state of the art experiments, since researchers won't need to process all the images once again. Furthermore, the created dataset is scalable due to the fact that it consists of 3 dimensional images, but also numerical values that can be used with system training purposes. This is relevant because the size of the numerical data is significantly lesser than the size of the images, so its download asks for less energy usage and therefore leaves a smaller carbon footprint.

In conclusion, while projects revolving around Machine Learning topics are usually not environmentally friendly, this thesis tries to minimize its carbon footprint by being conscious about the common struggles of dataset and, in particular, image processing. It also shares initiatives to minimize the negative environmental impact of related projects in an attempt to be socially sustainable.

8. Conclusions and future development

In this thesis, a whole ANTs-based brain MRI processing pipeline was implemented, in addition to a resource usage monitoring method and a quality check system that evaluates brain segmentation into tissues. The results of this pipeline were used to feed an XGBoost brain age prediction system relying on cortical regions' volumes and thicknesses. This pipeline can be used to perform high dimension processing tasks in computing services able to provide a substantial amount of resources, and it can also be easily adapted to future research necessities due to its modular design.

The optimal configuration for this pipeline involved deleting unnecessary files as soon as they were not required and running various subjects in parallel to minimize the total processing time. The quality control module rated each subject's segmentation from 0 to 100 based on previous outlier classifiers. Outlier detection methods included using the subject's Pearson correlation and mutual information between the template's gray matter and the registered, low pass filtered subjects' gray matter in addition to pairing subjects' posteriors' volumes to draw data clusters and identifying the isolated subjects using Local Outlier Factor. Region volume and cortical thickness extraction used ANTs' tools to parcel the brain's cortical thickness into 62 ROI using 20 DKT atlases and the binarized subject's cortical thickness as a mask. The resource usage monitoring system proved that using basic Linux commands in addition to a text parser was sufficient to summarize multiple resource consumption metrics along the total processing time. Regarding the brain age prediction model, the data integrity feature included in the XGBoost script was used to identify data samples that had complete information. Also, discarding wrongly segmented subjects when training the system demonstrated to slightly improve the results, while showing that the brain segmentation into tissues had low influence on the regional volume and cortical thickness extraction process.

Finally, many future research ideas spring up from this project:

- Testing the brain age prediction module using another processing core such as SPM.
- Using another Machine Learning system to predict the brain age with the same data.
- Testing the system with data extracted from the cortical thickness surface, for example the mesh or its vertices, in addition to adding other tissues' information.
- Developing a brain age prediction model optimized to ANTs' data and perform more robust processing core comparisons, or exploring new ML models.
- Testing dedicated tools to monitor the resource usage rather than sticking to Linux basic commands.
- Testing more masks and different atlases on the cortical parcellation stage.
- Performing the QC module using smaller regions rather than whole tissues, for example, the 62 volumes outputted from the region data extraction module.
- Exploring more metrics and more methods to run the QC module with, for example, N-dimensional volume groups, different kinds of correlations, removing high level information with morphological techniques rather than low pass filters, or test unsupervised approaches such as clustering with k-means.
- Configuring the basic pipeline with other parameters, such as using different templates or setting other options allowed in the ANTs' script.

Bibliography

[1] Fjell, Anders M et al. “What is normal in normal aging? Effects of aging, amyloid and Alzheimer's disease on the cerebral cortex and the hippocampus”. *Progress in neurobiology*, vol. 117, pp. 20-40, June 2014. doi: 10.1016/j.pneurobio.2014.02.004.

[2] Lorenzo Pini, Michela Pievani, Martina Bocchetta, Daniele Altomare, Paolo Bosco, Enrica Cavedo, Samantha Galluzzi, Moira Marizzoni, Giovanni B. Frisoni. “Brain atrophy in Alzheimer’s Disease and aging”. *Ageing Research Reviews*, *ISSN 1568-1637*, vol. 30, pp. 25-48, 2016. doi: 10.1016/j.arr.2016.01.002.

[3] Bruce Fischl. “FreeSurfer”. *NeuroImage*, vol. 62, issue 2, pp. 774-781, 2012. doi: 10.1016/j.neuroimage.2012.01.021.

[4] Anders M. Dale, Bruce Fischl, Martin I. Sereno. “Cortical Surface-Based Analysis: I. Segmentation and Surface Reconstruction”. *NeuroImage*, vol. 9, issue 2, pp. 179-194, 1999. doi: 10.1006/nimg.1998.0395.

[5] Avants, Brian B., Nick Tustison, and Gang Song. "Advanced normalization tools (ANTS)." *The Insight Journal*, vol. 2, issue 365, pp.1-35, July 2009. doi: 10.54294/uvnhin.

[6] Das, Sandhitsu R et al. “Registration based cortical thickness measurement.” *NeuroImage,* vol. 45,3, pp. 867-79, 2009. doi: 10.1016/j.neuroimage.2008.12.016.

[7] Hodge, V., Austin, J. “A Survey of Outlier Detection Methodologies”. *Artificial Intelligence*, review 22, pp. 85–126, 2004. doi: 10.1023/B:AIRE.0000045502.10941.a9.

[8] Abir Smiti. “A critical overview of outlier detection methods”. *Computer Science Review*, vol. 38, 2020. doi: 0.1016/j.cosrev.2020.100306.

[9] Juan Eugenio Iglesias, Mert R. Sabuncu. “Multi-atlas segmentation of biomedical images: A survey”. *Medical Image Analysis*, vol. 24, issue 1, pp. 205-219. doi: 10.1016/j.media.2015.06.012.

[10] Aljabar, P., Heckemann, R., Hammers, A., Hajnal, J.V., Rueckert, D. “Classifier Selection Strategies for Label Fusion Using Large Atlas Databases”. *Computer Science*, *Springer,* vol 4791, 2007. doi: 10.1007/978-3-540-75757-3_64.

[11] Qureshi, Muhammad Naveed Iqbal & Min, Beomjun & Jo, Hang & Lee, Boreom. “Multiclass Classification for the Differential Diagnosis on the ADHD Subtypes Using Recursive Feature Elimination and Hierarchical Extreme Learning Machine: Structural MRI Study”. *PLOS ONE*, vol. 11, August 2016. doi: 10.1371/journal.pone.0160697.

[12] Klein A, Ghosh SS, Bao FS, Giard J, Hame Y, Stavsky E, Lee N, Rossa B, Reuter M, Neto EC, Keshavan A. “Mindboggling morphometry of human brains”. *PLoS Computational Biology*, vol. 13(3), 2017. doi: 10.1371/journal.pcbi.1005350.

[13] Chen, T. & Guestrin, C. “XGBoost: a scalable tree boosting system”. *In Proc. 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, San Francisco, CA, USA. pp. 785–794. doi: 10.1145/2939672.2939785.

[14] James H. Cole, Rudra P.K. Poudel, Dimosthenis Tsagkrasoulis, Matthan W.A. Caan, Claire Steves, Tim D. Spector, Giovanni Montana. “Predicting brain age with deep learning from raw imaging data results in a reliable and heritable biomarker”. *NeuroImage*, vol. 163, pp. 115-124, 2017. doi: 10.1016/j.neuroimage.2017.07.059.

[15] Ann-Marie G. de Lange, Melis Anatürk, Sana Suri, Tobias Kaufmann, James H. Cole, Ludovica Griffanti, Enikő Zsoldos, Daria E.A. Jensen, Nicola Filippini, Archana Singh-Manoux, Mika Kivimäki, Lars T. Westlye, Klaus P. Ebmeier. “Multimodal brain-age prediction and cardiovascular risk: The Whitehall II MRI sub-study”. *NeuroImage*, vol. 222, November 2020. doi: 10.1016/j.neuroimage.2020.117292.

[16] Nicholas J. Tustison, Philip A. Cook, Arno Klein, Gang Song, Sandhitsu R. Das, Jeffrey T. Duda, Benjamin M. Kandel, Niels van Strien, James R. Stone, James C. Gee, Brian B. Avants. “Large-scale evaluation of ANTs and FreeSurfer cortical thickness measurements”. *NeuroImage*, vol. 99, pp. 166-179, 2014. doi: 10.1016/j.neuroimage.2014.05.044.

[17] Tustison, Nicholas & Cook, Philip & Holbrook, Andrew & Johnson, Hans & Muschelli, John & Devenyi, Gabriel & Duda, Jeffrey & Das, Sandhitsu & Cullen, Nicholas & Gillen, Daniel & Yassa, Michael & Stone, James & Gee, James & Avants, Brian. (2021). “The ANTsX ecosystem for

quantitative biological and medical imaging”. *Scientific Reports*, vol. 11, April 2021. doi: 10.1038/s41598-021-87564-6.

[18] Yaakub, S.N., Heckemann, R.A., Keller, S.S. et al. “On brain atlas choice and automatic segmentation methods: a comparison of MAPER & FreeSurfer using three atlas databases”. *Sci Rep 10*, vol. 10, issue 1, p. 2837, February 2020. doi: 10.1038/s41598-020-57951-6

Appendices

1. Work plan development

The most detailed items from the introduction can be seen in this appendix if any detailed explanation is needed.

1.1 Requirements and specifications

REQUIREMENT	SPECIFICATION
Prioritize using software from the ANTs' toolkit to perform any kind of image processing technique.	At least 95% of all the image processing techniques have to be performed with ANTs' tools.
Work with BBRC usual software.	Spend at least 80 hours working with Python and ANTs to achieve the required level of expertise.
Find or create useful metrics for evaluating the resource cost of running the preprocessing step.	Use at least 3 metrics found from various sources and/or commonly used by BBRC.
Find or create useful metrics for assessing the most important outputs of the preprocessing pipeline.	Use at least 3 metrics found from various sources and/or commonly used by BBRC.
Apply the previous metrics to a subset of data and obtain remarkable scores when assessing the results.	Obtain at least a 80% score with at least 90% of subjects. The total number of subjects should not be inferior to 100.
Fully process a batch involving all the implemented steps.	The batch size has to be of 200 subjects minimum. All subjects have to include their demographic information.
Assess the brain age prediction model with the data extracted from the preprocessing pipeline.	Test the model with a minimum of 150 input subjects with their corresponding demographic data.

Table A1: Project requirements and specifications

1.2 Methods and procedures

This project is a semi-isolated snippet of Irene Cumplido's PhD dissertation, and it therefore complies with the BBRC regulations and guidelines. It takes advantage of resources from various origins:

- Advanced Normalization Tools (ANTs): developed by Brian B. Avants, Nicholas J. Tustison and Hans J. Johnson, this software is broadly used to perform the full brain normalization pipeline, including cortical thickness extraction.
- Local Outlier Factor (LOF): developed by Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng and Jörg Sander, this algorithm is commonly used to identify outliers from a point cloud.

- UK Biobank (UKB) dataset: a publicly available cohort that contains MRI scans for 50,000 participants (www.ukbiobank.ac.uk). This contains all the input images from this project, and it is also used to create a template representing the dataset.
- Montreal Neurological Institute (MNI) MRI template: an MRI template freely provided by the MNI and created from 152 brain images. This is used as a test template for many images.
- Mind Boggle dataset: developed by Arno Klein and Jason Tourville, this dataset consists of several brain scans manually segmented according to the same atlas, and it is used to segment images into various regions of interest (ROI).
- Brain age prediction model: developed by Irene Cumplido, this ML system uses Extreme Gradient Boost algorithm (XGBoost) to predict a subject's age based on features such as demographics and cortical regions' thicknesses and volumes.
- Calcula: this computing service arranged by the UPC provides computational resources that allow running demanding experiments remotely. This service has been used along the majority of the project to run a variety of simulations.

1.3 Work plan

1.3.1. Work packages

Project: MRI dataset preprocessing	WP ref: 1	
Major constituent: Research	Sheet 1 of 1	
Short description: To have a better understanding of the project and the agents involved in it, it is necessary to perform a literature review previous to the work itself. This will include purely neuroscientific matters, image processing subjects, and neuroimaging state of the art topics.	Planned start date: 14/02/2022 Planned end date: 28/02/2022 Start date: 14/02/2022 End date: 28/02/2022	
Internal task T1: Research on MRI, brain-aging and neuropathologies. *Internal task T2:* Research on tools: programming languages, MRI normalization tools, version control, tools provided by BBRC. *Internal task T3:* Research on state of the art neuroimaging techniques.	Deliverables: None	Dates: None

Table A2: WP #1 - Research.

Project: MRI dataset preprocessing	WP ref: 2	
Major constituent: Software	Sheet 1 of 1	
Short description: Set up the required tools in the working environment: Python, Bash, ANTsx, Gitlab, dataset cloud.	Planned start date: 28/02/2022 Planned end date: 7/03/2022 Start date: 28/02/2022 End date: 7/03/2022	
Internal task T1: Set up a development environment with the desired programming languages and related packages. *Internal task T2:* Access to BBRC tools: Gitlab and dataset cloud.	Deliverables: None	Dates: None

Table A3: WP #2 - Environment setup.

Project: MRI dataset preprocessing	WP ref: 3	
Major constituent: Computational resource monitoring	Sheet 1 of 1	
Short description: Create, implement and compute metrics to register the most computational resource respectful way of preprocessing the dataset.	Planned start date: 7/03/2022 Planned end date: 14/03/2022 Start date: 7/03/2022 End date: 20/03/2022	
Internal task T1: Find or create and implement the metrics. *Internal task T2:* Test the metrics with a subset of images and enhance.	Deliverables: Code	Dates: 21/06/2022

Table A4: WP #3 - Resource usage monitoring.

Project: MRI dataset preprocessing	WP ref: 4	
Major constituent: Quality control	Sheet 1 of 1	
Short description: Create or find, implement and compute metrics to numerically validate and rate the processed data. The goal of this work package is to develop a truthful technique that allows to quality check images putting aside human intervention, and use it to determine the optimum preprocessing techniques before actually running the process.	Planned start date: 14/03/2022 Planned end date: 14/04/2022 Start date: 14/03/2022 End date: 31/05/2022	
Internal task T1: Find or create and implement the metrics. *Internal task T2:* Develop a first version that detects outliers (wrongly processed subjects). *Internal task T3:* Use the first version to achieve a final one that provides a numerical score to each of the processed subjects. *Internal task T4:* Test the system with a small subset of images and enhance it.	Deliverables: Code	Dates: 21/06/2022

Table A5: WP #4 - Processed images' quality control.

Project: MRI dataset preprocessing	WP ref: 5	
Major constituent: Data treatment	Sheet 1 of 1	
Short description: Using a subset of MRI scans, extract thickness and volume for each brain cortical segment. These segments are already defined in an atlas previously used by the BBRC, so the main tasks consist in performing the segmentation and extracting the data using the ANTs toolkit.	Planned start date: 14/05/2022 Planned end date: 31/05/2022 Start date: 14/05/2022 End date: 31/05/2022	
Internal task T1: Explore the tools that the ANTs toolkit offers for segmenting images and find the most suitable one. *Internal task T2:* Find a way to extract thickness and volume regional data.	Deliverables: Code	Dates: 21/06/2022

Internal task T3: Test on small batch and enhance the extraction process.		

Table A6: WP #5 - Region thickness and volume extraction.

Project: MRI dataset preprocessing	WP ref: 6	
Major constituent: Machine Learning	Sheet 1 of 1	
Short description: Adapt the Machine Learning system provided by the BBRC to accept the data extracted in the previous WP.	Planned start date: 01/06/2022 Planned end date: 04/06/2022 Start date: 01/06/2022 End date: 04/06/2022	
Internal task T1: Adapt the system's input reading mechanism *Internal task T2:* Implement visuals to assess the results.	Deliverables: Code	Dates: 21/06/2022

Table A7: WP #6 - Machine Learning model adaptation.

Project: MRI dataset preprocessing	WP ref: 7	
Major constituent: Image processing	Sheet 1 of 1	
Short description: Preprocess a bigger batch of the dataset optimally according to the previous work packages. Use the 6 main stages offered by the ANTs toolkit and add the previously implemented ones.	Planned start date: 14/04/2022 Planned end date: 14/05/2022 Start date: 01/06/2022 Planned end date: 10/06/2022	
Internal task T1: Skull stripping: erase skull from MRI. *Internal task T2:* Registration and bias correction: remove low frequency intensity non-uniformities and register input to template. *Internal task T3:* Segmentation: classify the different types of brain tissues: gray matter, white matter, … *Internal task T4:* Second registration: enhance the first registration, optional. *Internal task T5:* Estimate cortical thickness from the previously obtained gray matter segment. *Internal task T6:* Internal quality control: images showing segmentations and cortical thickness. *Internal task T7:* Compute the previously implemented validation quality check. *Internal task T8:* Extract regional volume and thickness. *Internal task T9:* Feed the regional data to the adapted Machine Learning system and assess the results.	Deliverables: Processed dataset extracted numerical data (quality control, regional volumes and thicknesses)	Dates: 10/06/2022

Table A8: WP #7 - Preprocessing.

Project: MRI dataset preprocessing	WP ref: 8	
Major constituent: Documentation	Sheet 1 of 1	
Short description: Document the progress and sources used along the process.	Planned start date: 14/02/2022 Planned end date: 21/06/2022	
Internal task T1: Use Gitlab to register the technical progress and scripts. *Internal task T2:* Create visuals to support the thesis' topics. *Internal task T3:* Gather all the information in the different text deliverables.	Deliverables: Codes Work plan Critical review Final report	Dates: 21/06/2022 08/03/2022 14/04/2022 21/06/2022

Table A9: WP #8 - Documentation.

1.3.2. Milestones

WP#	Task#	Short title	Milestone / deliverable	Date (week)
2	2	Set-up complete	Working environment and information sources ready to begin with the actual work.	3
3	2	Resource usage registered	The resource cost metrics for processing the dataset are now computed and an optimum way of running the scripts has been obtained.	5
4	4	Quality check processes defined	The processing quality check system has been implemented and enhanced within the test span.	16
5	3	Thickness and volume data extracted	A method to segment the brain cortex and extract each segment's thickness and volume has been implemented.	16
6	1	Machine Learning model adapted	The brain age prediction model has been adapted to read the output of the processing pipeline	17
7	8	Pipeline performed	The whole processing pipeline has been performed on a bigger batch and the results have been used to test the brain age prediction system.	18
8	3	Thesis completed	All the information besides illustrative visuals have been written into the thesis document	19

Table A10: Project milestones.

1.3.3. Gantt diagram

The Gantt diagram has changed since the Critical Review according to the WP deviations. Mainly, WP #7 (preprocessing) has been implemented with a small subject batch instead of the whole dataset due to time limitations. Furthermore, WP #6 (numerical data extraction) has been added to achieve a more complete project regarding the Machine Learning field. Finally, WP #4 (quality control) took more time than expected, but it did not imply any major delays in the whole plan.

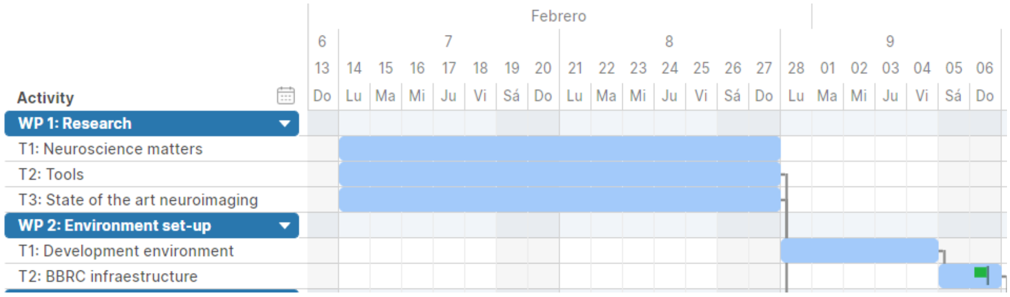


Figure A1: Gantt diagram weeks 1 to 3.

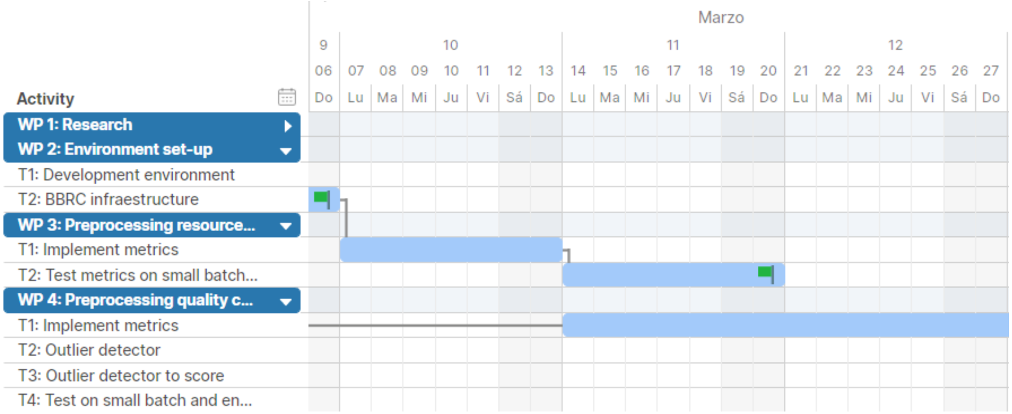


Figure A2: Gantt diagram weeks 4 to 6.

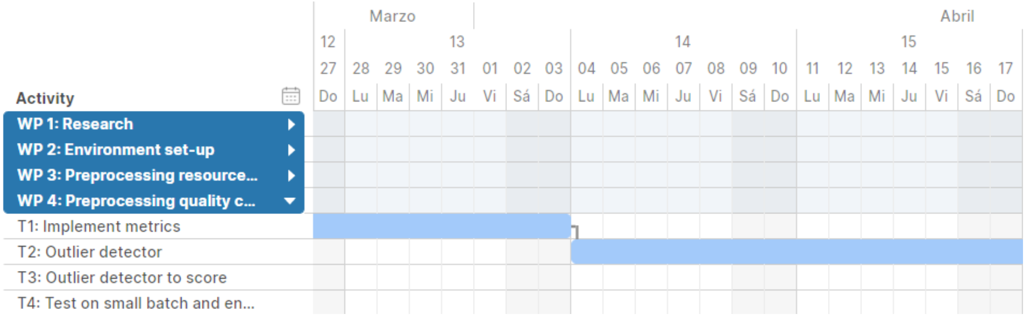


Figure A3: Gantt diagram weeks 7 to 9.

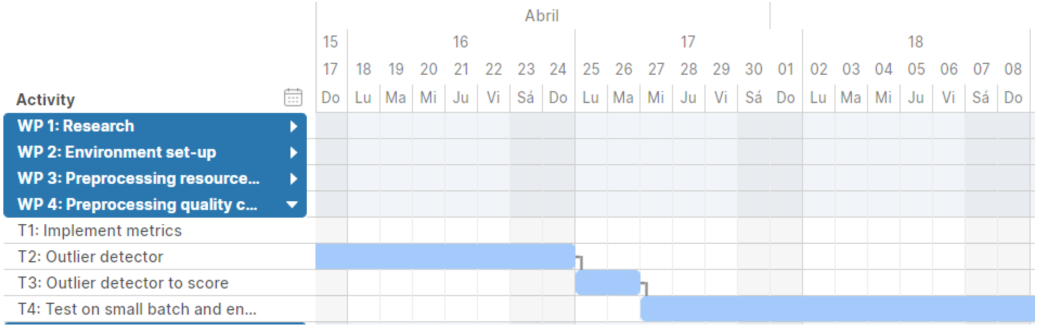


Figure A4: Gantt diagram weeks 10 to 12.

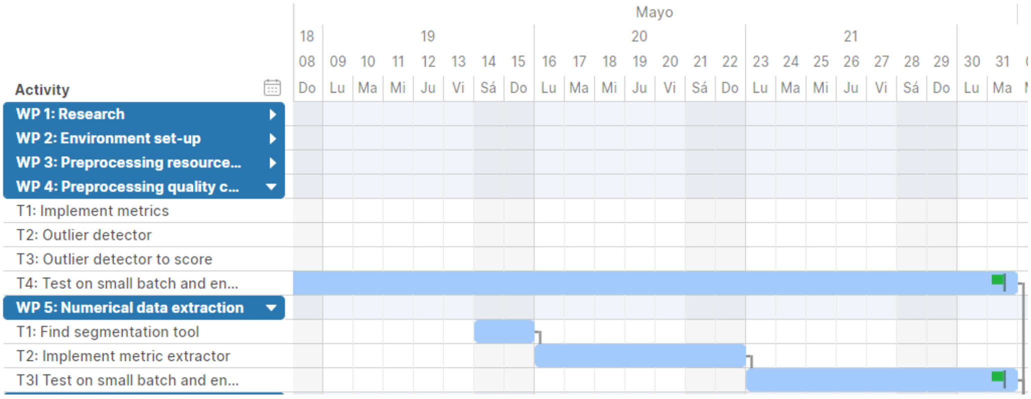


Figure A5: Gantt diagram weeks 13 to 15.

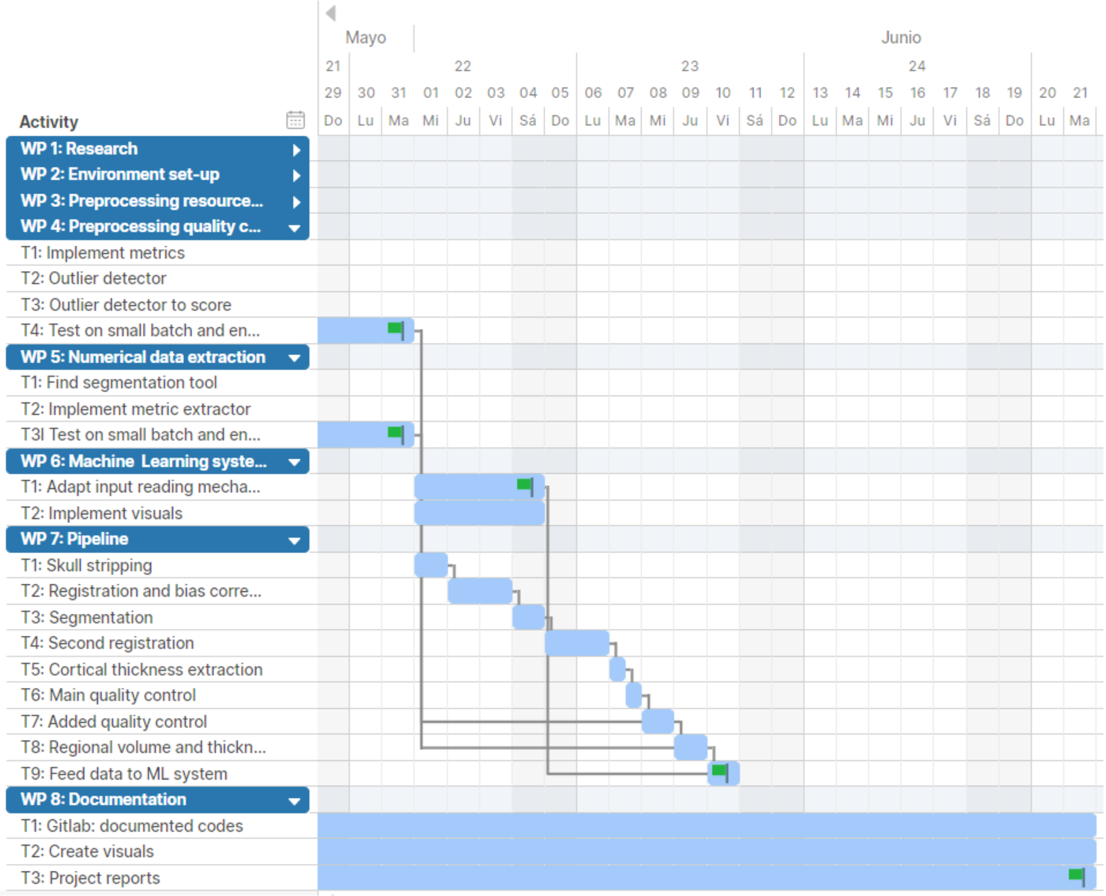


Figure A6: Gantt diagram weeks 16 to 19.

The complete Gantt diagram can be seen in the Tom's planner platform accessing with: https://plan.tomsplanner.es/public/adriasolanatfg

1.4 Initial plan deviations and incidences descriptions

There have been some problems that may have contributed to delay some of the milestones. Briefly listing them:

At the beginning of the project, the main computer designated to run the tests was not able to complete even the ones that needed the least resources. The UPC provided computing services to deal with this issue, but both soliciting the services and the adaptation process took about two weeks.

Once the project was migrated to the computing services, a maintenance by UPC staff took place just starting the month of May. This maintenance, whilst scheduled to last 8 hours, turned out lasting 4 days intermittently, killing all running scripts each time. Furthermore, the services' users were not given previous notice of when this was going to happen, so there was no way of planning ahead.

2. ITK Mutual Information

ANTs uses ITK tools to perform several operations. This package provides mutual information as a correlation metric, but the values outputted are not the ones expected for this metric. Please see the explanation in the following ITK wiki page:

https://itk.org/Wiki/ITK/Mutual_Information

3. DKT atlas labels

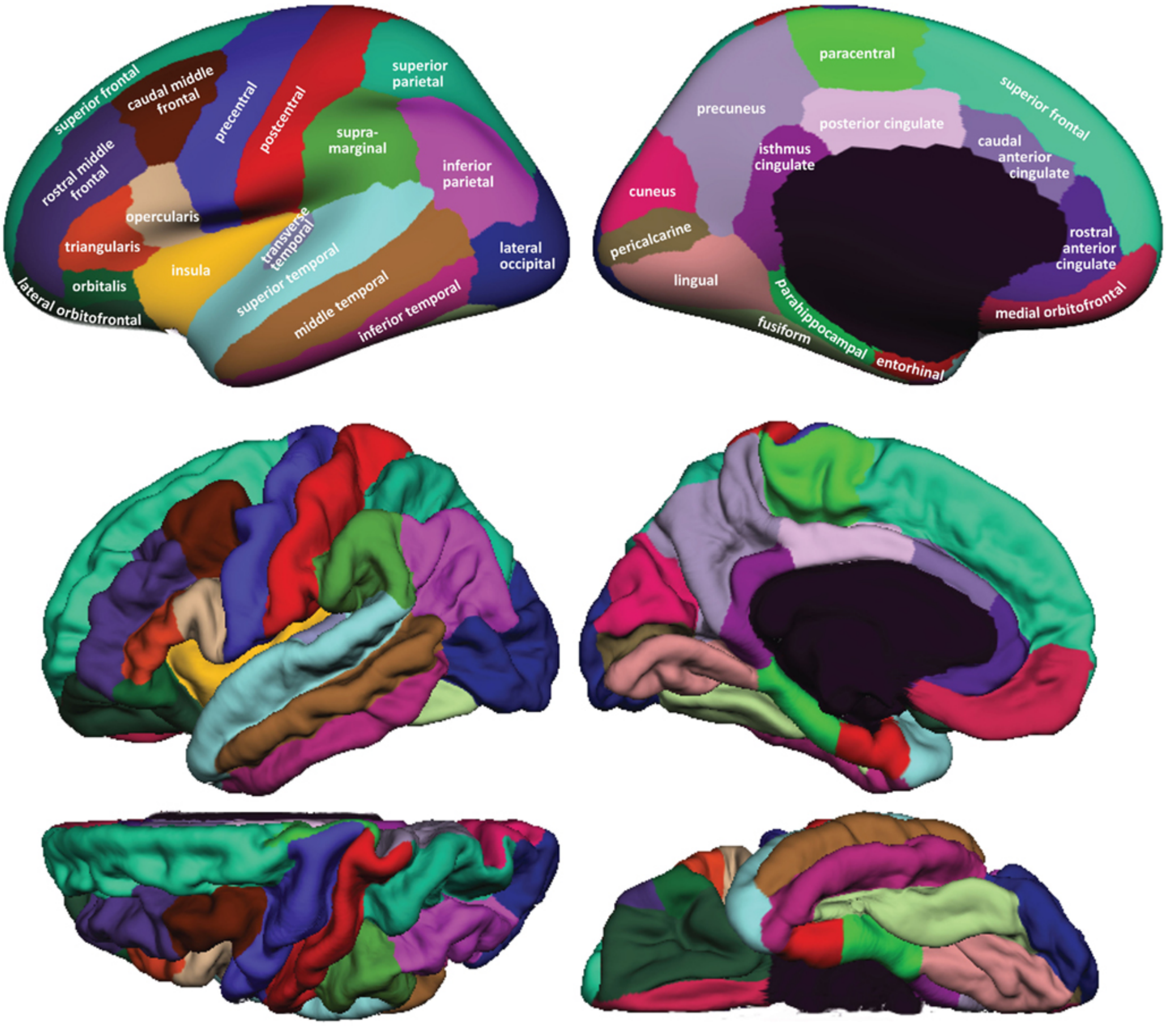


Figure A7: DKT atlas regions of interest (or labels). Source: [11]

4. Decision tree examples

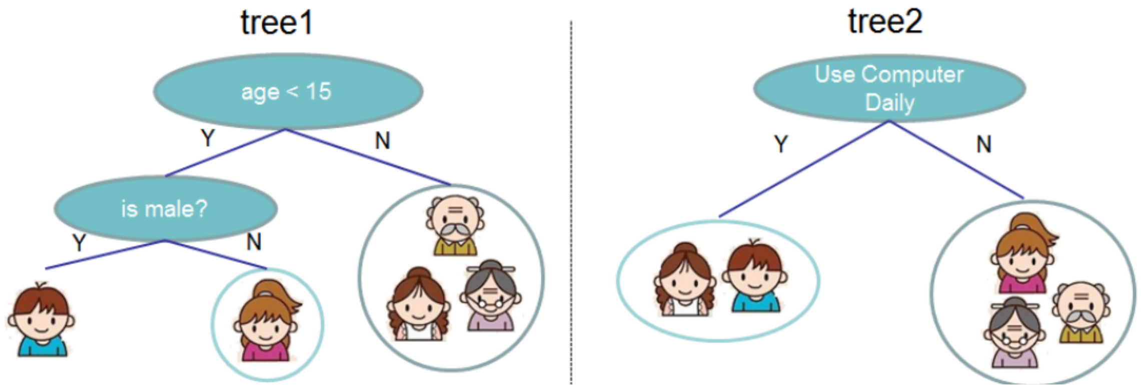


Figure A8: Decision tree examples. Source: [13]

5. Classifier assessment

Many metrics exist to evaluate a binary classifier’s performance, but the most commonly used are Precision, Recall and F-score. They are based on the following concepts:

- Positive (p): class 1
- Negative (n): class 0
- True positive (tp): a classifier has classified a class 1 input as class 1.
- True negative (tn): a classifier has classified a class 0 input as class 0.
- False positive (fp): a classifier has classified a class 0 input as class 1.
- False negative (fp): a classifier has classified a class 1 input as class 0.

		Real class	
		Positive	Negative
Predicted class	Positive	TP	FP
	Negative	FN	TN

Table A11: Classifier assessment table.

Intuitively, a classifier aims to maximize the numbers of true positives and true negatives. To assess the general classifier performance, two metrics have to be simultaneously taken into account:

- Precision: how many of the positive predictions are correct, tp / (tp + fp).
- Recall: how many of the positive classes have been identified as such, tp / (tp + fn)

Optimally, both metrics should be equal to 1.

F-score is the metric that joins these two values in a unique score that ranges from 0 to 1 using an harmonic mean:

- F-score = 2*(prec*rec)/(prec+rec)

6. Linux resource monitoring commands

There are mainly two basic linux commands that can be used to monitor processes resource usage:

- *top*: this is used to see the machine's CPU, RAM and swap memory usage along with the states of the selected processes, the state of the machine and the resources that each individual process is using. This command shows a screen in the terminal that refreshes over time, but it has an internal configuration that allows it to sample the processes' state over time.

  See: https://man7.org/linux/man-pages/man1/top.1.html
- *ps*: this can be used to obtain near to identical metrics than *top*, but the CPU provided by *ps* refers to the total time that the CPU has been running as a percentage of the process' execution time, rather than the "task's share of CPU since the last screen update".

  See: https://man7.org/linux/man-pages/man1/ps.1.html

7. Common Machine Learning evaluation metrics

Various metrics are used to evaluate the XGBoost system used on the cortical regions' volumes and thicknesses. These are:

- Pearson correlation coefficient (r): Pearson's r ranges from -1 to 1 and represents the correlation between two sets of data. If applied on the predicted age and the real age, it becomes a useful assessment tool. If this value equals -1 or 1, this means that the sets under evaluation are perfectly correlated. In the prediction's case, obtaining a Pearson's r equal to 1 would mean that the system can perfectly predict the brain age through the input data, while obtaining a value nearer to 0 would indicate that the model's prediction capacities are near null. This metric is computed as a ratio between the two sets covariance and the product of each sets' standard deviation.

$$r = \frac{\Sigma(x-\overline{x})\,(y-\overline{y})}{\sqrt{\Sigma(x-\overline{x})^2\Sigma(y-\overline{y})^2}}$$

- Mean Absolute Error (MAE): this is the mean of the absolute value of the difference between each input target age and its prediction. It is a good estimation of how many ages the model has erred in average, whether by excess or by shortage.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |(x - y)|$$

- Root Mean Square Error (RMSE): it is calculated in a similar way than MAE, but it elevates each difference to the power of two and roots the resulting “mean”. Due to squaring properties, big errors get emphasized, so it is a more adequate metric when it is a priority to not let some errors get relatively too big.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x - y)^2}$$

8. QC tests on the 106 subject batch

The two correlation classifiers tested on the 106 subject batch were:

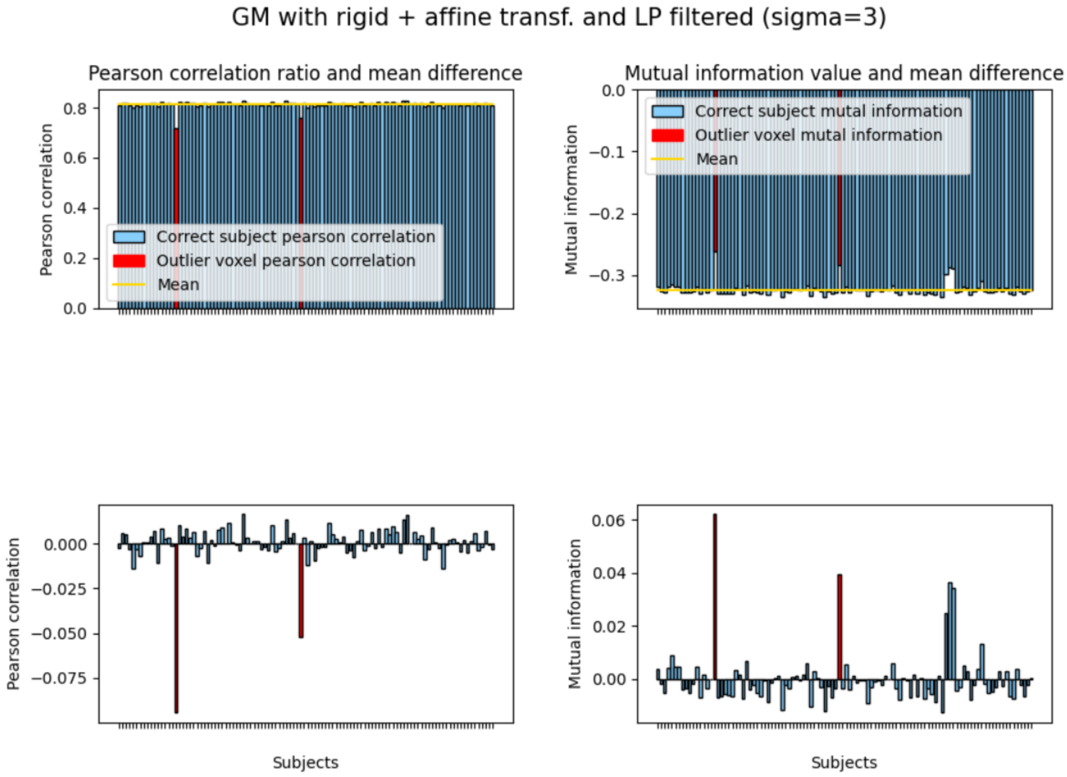


Figure A9: Correlations of each subject’s GM affine registered to the template’s GM and low pass filtered with a sigma = 3 using the 106 subject batch,

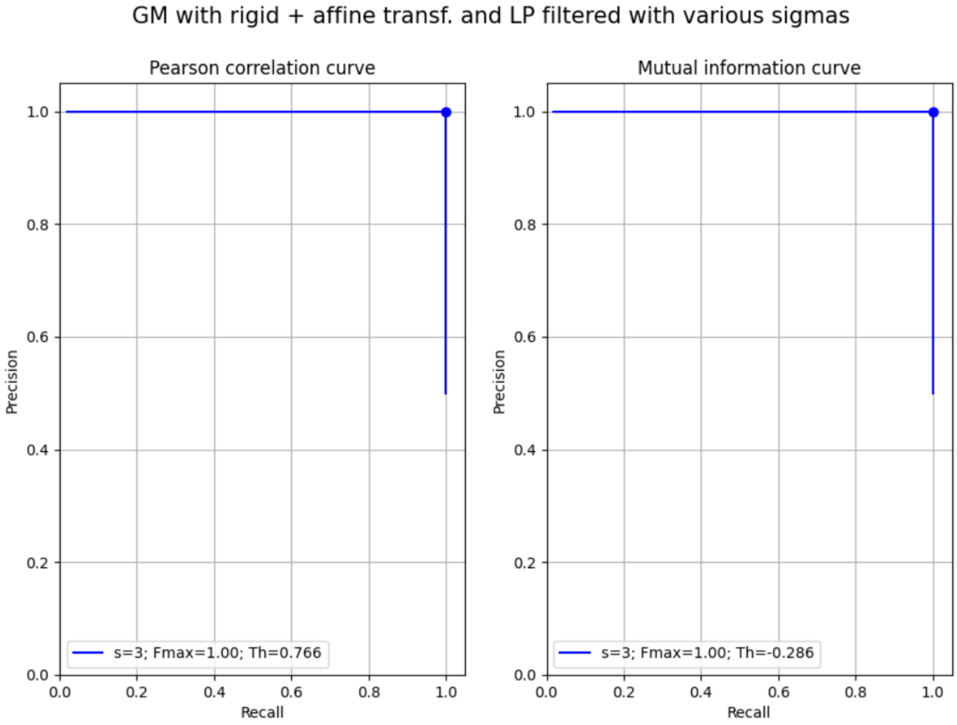


Figure A10: Precision-recall curves over a distance-based classifier applied on the previous correlations.

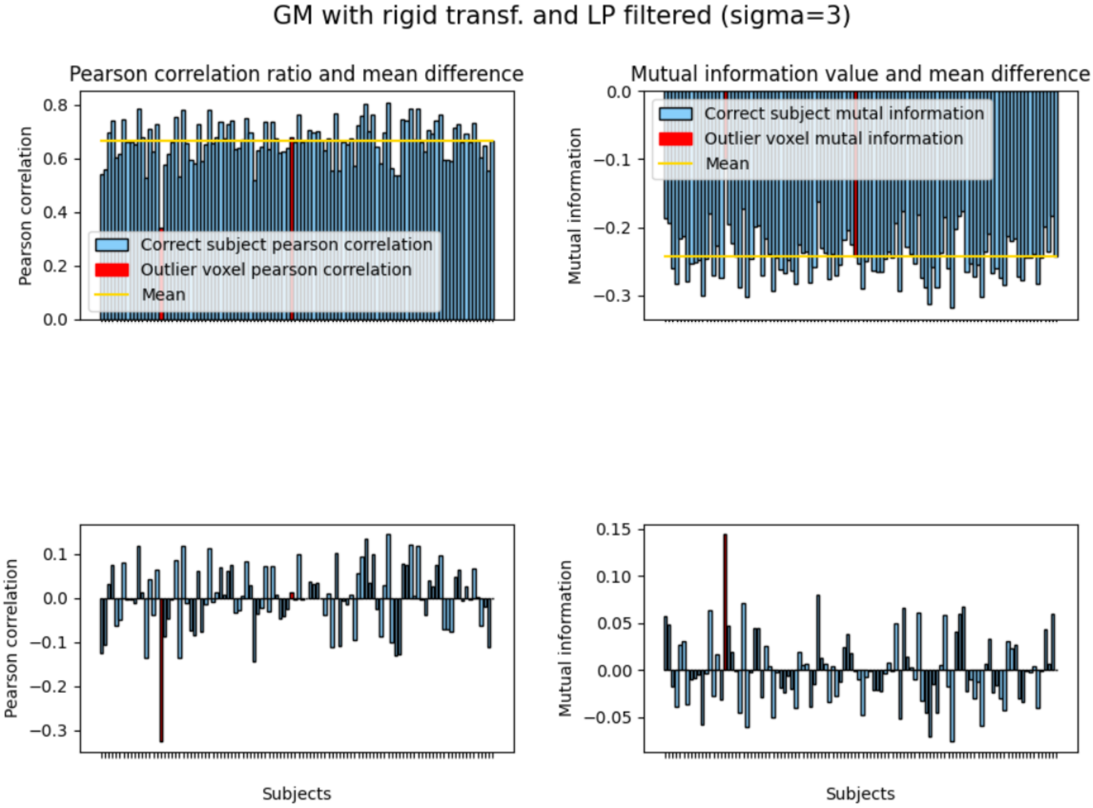


Figure A11: Correlations of each subject's GM rigidly registered to the template's GM and low pass filtered with a sigma = 3 using the 106 subject batch.

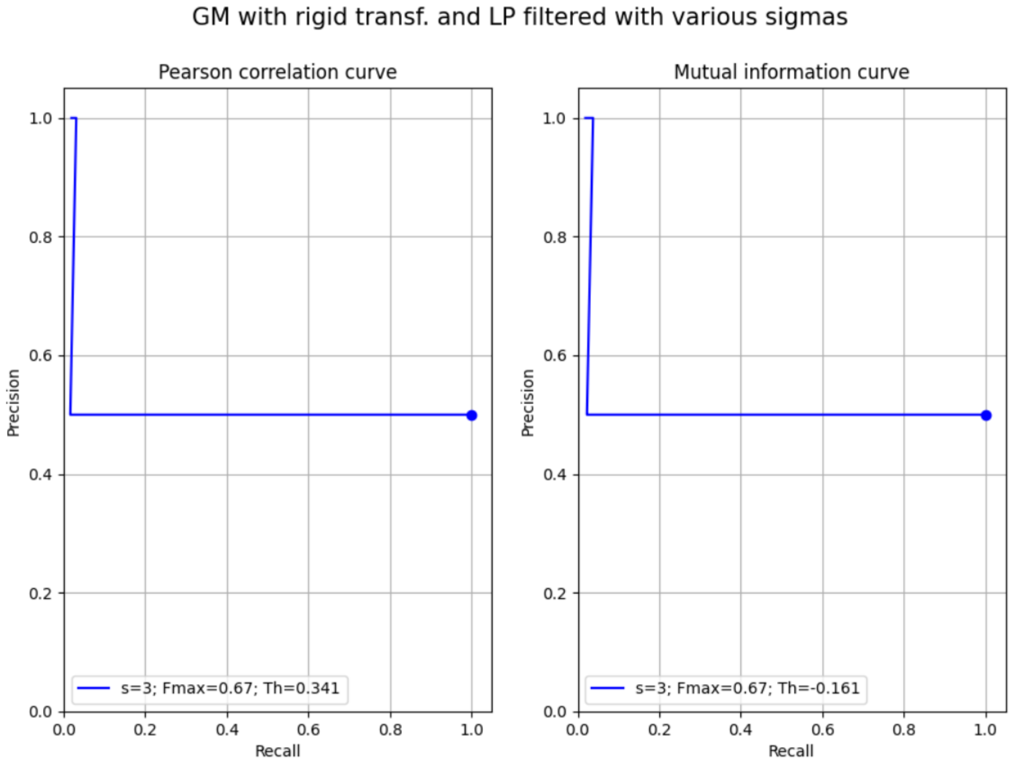


Figure A12: Precision-recall curves over a distance-based classifier applied on the previous correlations.

9. Representation of the resource usage module outcome

File containing a summary of the resource usage metrics, averaged over all the process rather than grouped by subject (*resources_summary.log*) for three parallel subjects:

```
================================================================================
                        AVERAGE PROCESS RESOURCE USAGE
================================================================================

        CPU PARAMETERS
------------------------------------
Userspace process time (us): 20.30 %
Kernelspace process time (sy): 0.56 %
Nice value (ni): 0.00 %
Idle time (id): 78.98 %
I/0 completion time (wa): 0.02 %
Hardware interruption time (hi): 0.00 %
Software interruption time (si): 0.13 %
CPU busy in other VM (st): 0.00 %
CPU usage percentage (ps command): 0.0 %

        RAM PARAMETERS
------------------------------------
Total memory (total): 270239.57 MB
Free memory (free): 181975.91 MB
Used memory (used): 9431.27 MB
Cache memory (buff/cache): 78832.38 MB

        SWAP PARAMETERS
-------------------------------------
Total memory (total): 2146.52 MB
Free memory (free): 1857.93 MB
Used memory (used): 288.58 MB
```

```
Available non swapping memory (avail mem): 258816.37 MB

        PROCESS PARAMETERS
------------------------------------
Total consumed virtual memory (VIRT): 15.74 MB
Total consumed resident memory (RES): 3.44 MB

           I/O SIZE
------------------------------------
Total input size: 88.67 MB (88666976.00 B)
Total output size: 272.64 MB (272638380.00 B)
I/O size ratio: 3.07

======================================================================
                         SLURM JOB INFO
======================================================================
1372314.0     |   TIME    |  CPUs  |   MEM   |   GPUs   | GPU MEM |
-------------------|--------------|-------------|------------|-------------|----------------|
REQUESTED |  24h0m  |  6.00   |   16G   |    0      |    0        |
USED          |  7h45m  |  5.23   |  2.35G  | unknown | unknown   |
======================================================================

======================================================================
                            CPU INFO
======================================================================
Architecture:        x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              8
On-line CPU(s) list: 0-7
Thread(s) per core:  1
Core(s) per socket:  1
Socket(s):           8
NUMA node(s):        1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               61
Model name:          Intel Core Processor (Broadwell, IBRS)
Stepping:            2
CPU MHz:             2199.998
BogoMIPS:            4399.99
Hypervisor vendor:   KVM
Virtualization type: full
L1d cache:           32K
L1i cache:           32K
L2 cache:            4096K
L3 cache:            16384K
NUMA node0 CPU(s):   0-7
Flags:               fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr
sse sse2 ss syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopology cpuid tsc_known_freq pni
pclmulqdq
ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand
hypervisor lahf_
lm abm 3dnowprefetch cpuid_fault invpcid_single pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1 hle avx2
smep bmi2 erms invpcid rtm rdseed adx smap xsaveopt arat md_clear
```

File containing each the total processing time (*time.log*) for one the same batch of 3 subjects:

```
---------------------------- TIME REGISTER FOR 3 SUBJECT BATCH ----------------------------
```

```
3 subjects processed in 27937 seconds or 465 minutes or 7 hours
Each subjects processed in 9312 seconds or 155 minutes or 2 hours
```

File containing each module's processing time (*time.log*) for one single subject. In this case, the various modules' times are individually displayed.

```
---------------------------- TIME REGISTER FOR 1 SUBJECT BATCH ----------------------------


Basic processing performed in 5870 seconds or 97 minutes or 1 hours

Quality check performed in 111 seconds or 1 minutes or 0 hours

Region thickness performed in 11793 seconds or 196 minutes or 3 hours


Pipeline performed in 17774 seconds or 296 minutes or 4 hours
```

10. Brain age prediction dataset description

Used demographic information was sex and age.

Feature	Specifications	
	202 subject batch	196 subject batch
Gender	100 males 102 females	95 males 101 females
Age	Mean: 62.48 Std: 9.51 Min: 45.00 25%: 54.25 50% : 64.00 75%: 71.00 Max: 80.00	Mean: 62.64 Std: 9.49 Min: 45.00 25%: 55.00 50% : 64.00 75%: 71.00 Max: 80.00

Table A12: Brain age prediction dataset definition.

11. MRI modalities

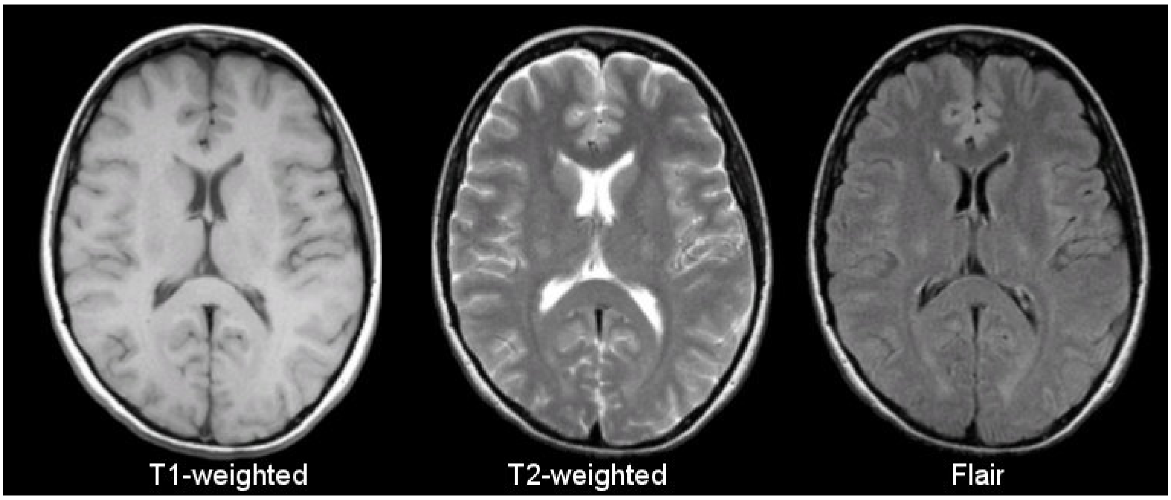


Figure A13: MRI scans retrieved with different modalities. Source: https://case.edu/med/neurology/NR/MRI%20Basics.htm

12. Coordinate systems

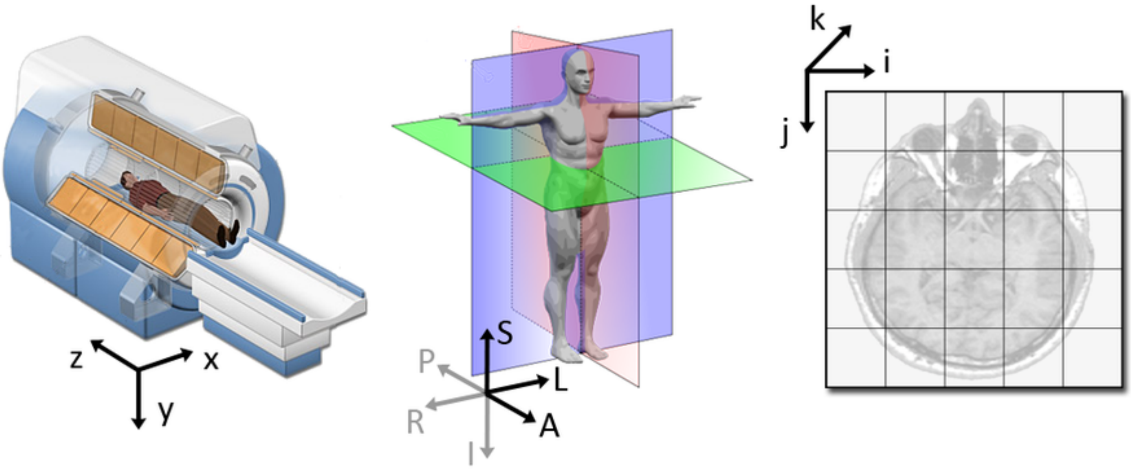


Figure A14: World, anatomical and image coordinates. Source: https://www.slicer.org/wiki/Coordinate_systems

13. Template examples

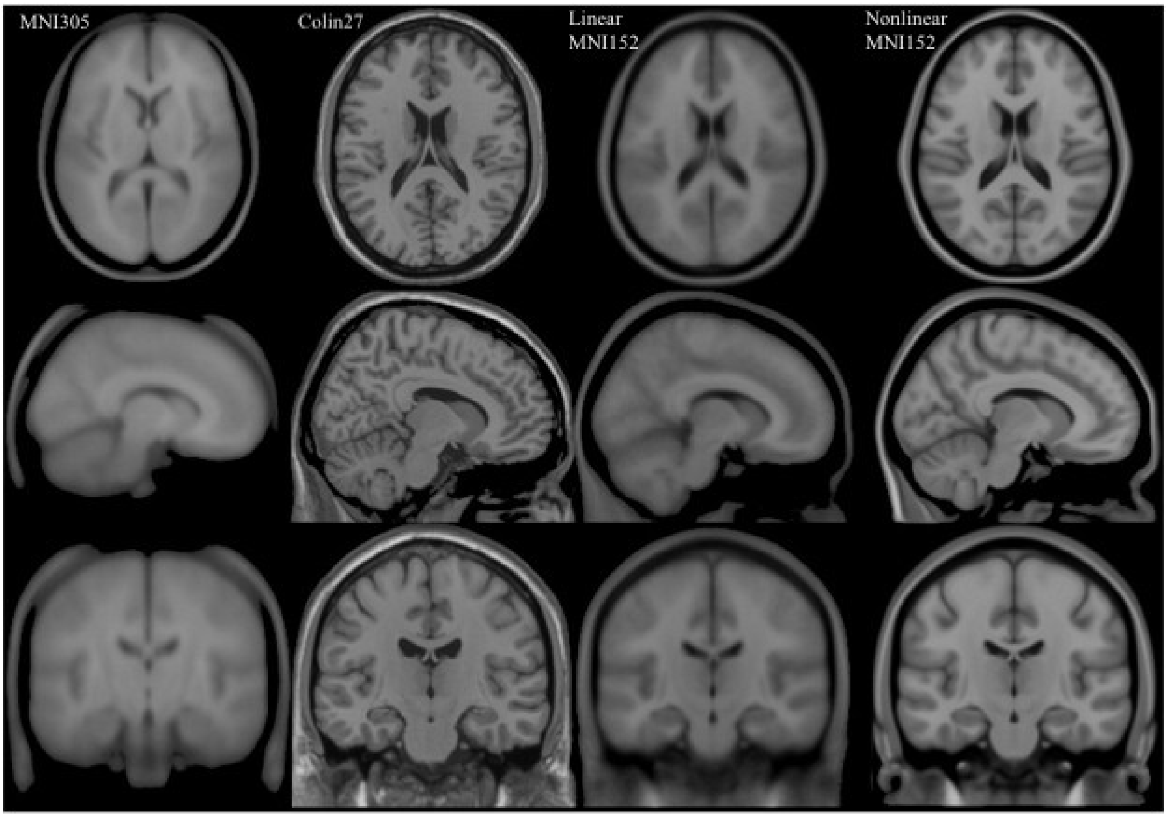


Figure A15: Template examples from 3 views. Source: https://carpentries-incubator.github.io/SDC-BIDS-sMRI/03-Image_Spatial_Normalization/index.html

14. Atlas examples

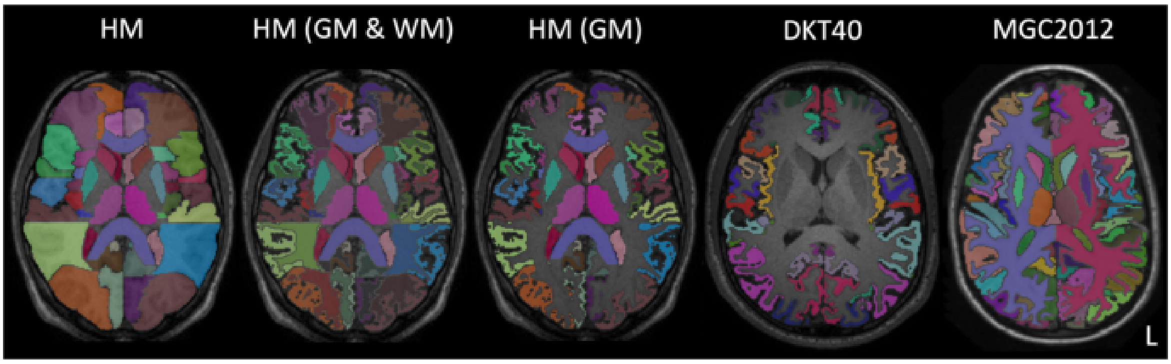


Figure A16: Different brain atlases examples. Source: [18]

15. Space and registration example

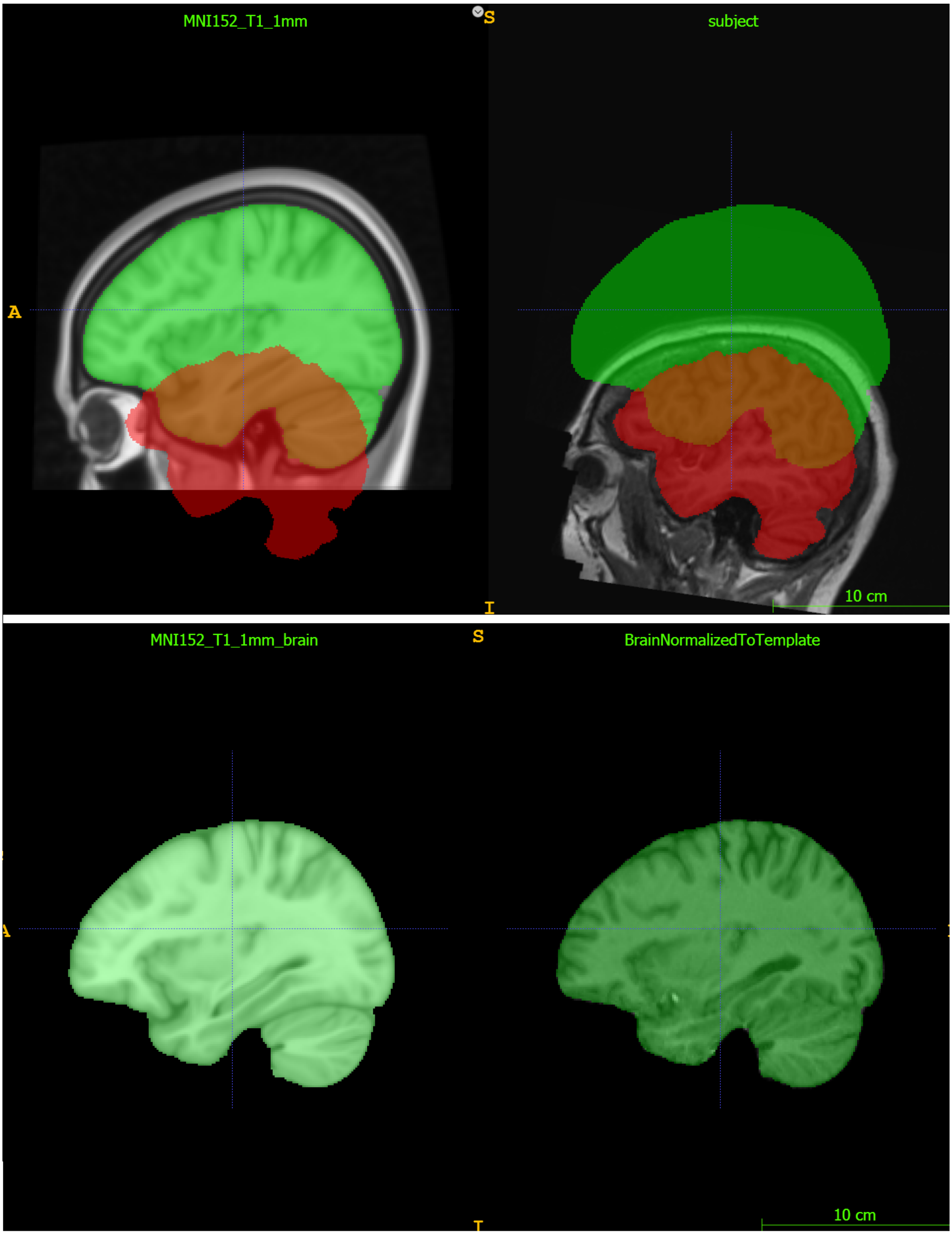


Figure A17: Subject MRI (right column) registered to template MRI (left column). First row shows the spaces before the registration and the second row displays its results for the extracted brains. The green mask represents the template's space and the red mask represents the subject space. At the end, both spaces match.

Glossary

Alphabetically:

- AD: Alzheimer’s Disease.
- ANTs: Advanced Normalization Tools.
- BBRC: Barcelonabeta Brain Research Center.
- CSF: Cerebrospinal Fluid.
- DGM: Deep Gray Matter.
- DL: Deep Learning.
- GM: Gray matter.
- LOF: Local Outlier Factor.
- MI: Mutual Information.
- ML: Machine Learning.
- MNI: Montreal Neurological Institute.
- MRI: Magnetic Resonance Imaging.
- OASIS: Open Access Series of Imaging Studies.
- ROI: Region(s) Of Interest.
- SPM: Statistical Parametric Mapping.
- UKB: United Kingdom Biobank.
- UPC: Universitat Politècnica de Catalunya.
- WM: White Matter.