POLITECNICO
MILANO 1863

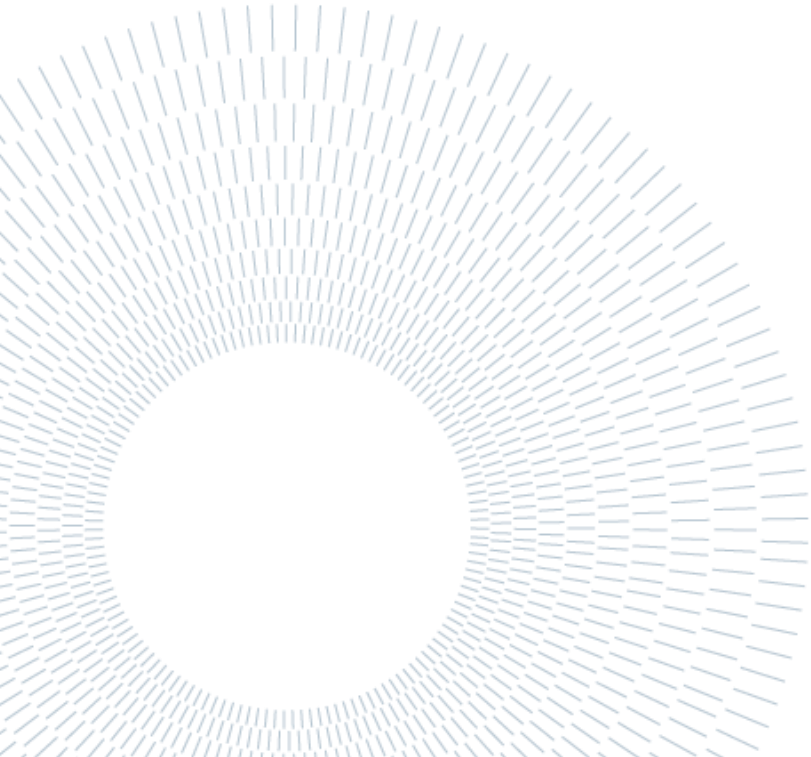
SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Assessment of White Matter Hyperintensities segmentation and their correlation with Dementia

TESI DI LAUREA MAGISTRALE IN
BIOMEDICAL ENGINEERING
INGEGNERIA BIOMEDICA

Author: **Stefano Calafà**

Student ID: 10496117
Advisor: Prof. Giuseppe Baselli
Co-advisor: Valentina Bordin
Academic Year: 2021-22



Abstract

CONTEXT - The growth in life expectancy occurred over the last few decades has contributed to an increase in the incidence of Neurodegenerative (NDs) and age-related conditions, such as cognitive decline, Multiple Sclerosis, Parkinson's or Alzheimer's disease (AD). In this context, an early-stage diagnosis is often associated with better rates of success in both limiting the disease progression and preventing symptoms development. Therefore, a strong pressure has been exerted on the scientific and medical community to identify biomarkers for both the detection and monitoring of NDs. Among these, White Matter Hyperintensities (WMHs), have recently gained more and more importance. These areas of abnormal intensity on Magnetic Resonance Imaging (MRI), were once treated as simple artifacts or neurological signs of healthy aging. However, latest findings have proved their association with demyelination and axonal loss, and have linked their presence with an increased risk of developing cognitive impairment along with several NDs. WMH segmentation is particularly important in both medical and research contexts, and several machine learning (ML) strategies have been developed over the past twenty years aiming to automatically segment and quantify their volume. Among these we find BIANCA, an automatic tool developed by the Oxford University based on the k-NN method. Regardless of being widely known, and despite the significant optimization endeavors that have been recently carried out to optimize its outcomes, some aspects still remain unexplored and open the lead for further improvements.

OBJECTIVES - The purpose of this thesis is twofold. On one hand, we aim to evaluate BIANCA performance according to different parameters such as: i) the number of subjects used for the training phase; ii) the different combinations of MRI contrasts involved in the process; iii) the different training strategy used (i.e., training and testing on images from different databases vs training and testing on images from the same database). This in order to find the proper combination of values able to optimize the final WMH segmentation. While carrying out the third step, a training set from a former harmonization study (involving the Whitehall and UK Biobank populations) was used, thus allowing for a simultaneous validation of its applicability. On the other hand, we aimed at demonstrating the role of WMHs as early-stage biomarker for AD dementia. This was assessed by feeding both clinical and imaging data (i.e., the WMH volumes extracted using BIANCA) to a

classification algorithm aimed to evaluate the Clinical Dementia Rate (CDR) of subjects affected by AD. Finally, the the importance exerted by WMHs on the final classification was Evaluated.

METHODS - To achieve the first objective of the thesis we downloaded 40 subjects having FLAIR, T1-weighted and Susceptibility Weighted Imaging (SWI) scans from the OASIS3 database. The images were pre-processed (i.e., brain extracted, biasfield corrected and registered to a common MRI modality), manually segmented and eventually fed to BIANCA. Its performance was evaluated according to the following steps: i) training on a pool of incremental subjects (10, 15, 20, 25, 30) and testing on the entire dataset using a 4-fold validation approach; ii) training and testing with the following combination of MRI modalities: "FLAIR only", "FLAIR + T1w", "FLAIR + SWI", "FLAIR + T1w + SWI" and "SWI only" on the entire dataset using a Leave-One-Out validation approach; iii) training BIANCA on the external training set provided by a former harmonization study and testing on the OASIS3 images (mixed training approach). As for the last step, performance was compared with both the existing literature and results from the incremental training set analysis (single-site training approach). The evaluation metric used to assess results was the Dice Similarity Index (DICE).
To achieve the second objective of this thesis, clinical variables from the OASIS3 were matched with their imaging counterpart (i.e., the WMH volumes extracted using BIANCA) to train and test three different Classification models: Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest Classifier (RFC). Performance was evaluated by means of accuracy, precision, recall and F1-score. The importance exerted by the different input variable on the final classification was instead evaluated through the permutation importance score.

RESULTS – As for BIANCA results, it is possible to notice an increased performance when adding more subjects to the training. However, a plateau is reached when the number of subjects approaches the value of 20 or 25 depending on the accuracy of the manual segmentations used to carry out the evaluation. The SWI contrast used is unable to provide any useful information, leading to insufficient results when used alone and to worsened performance when combined with other contrasts. Finally, when tested on the OASIS3, BIANCA trained with data from Whitehall and UK Biobank achieved comparable performance with that of the existing literature. In addition, the single-site training approach (represented by the incremental training analysis) showed improved results with respect to the mixed (i.e., harmonization training set) only when the number of involved subjects involved in the process was higher than or equal to 20.
As for the classification model, the SVM showed acceptable performance and pointed to WMHs as one of the most relevant features used to predict CDR. This

confirmed the importance of this neurological sign as imaging hallmarks for AD dementia.

CONCLUSIONS – With these findings we have optimized the automatic segmentation strategy represented by BIANCA. In addition, we have validated a preliminary harmonization pipeline created to derive integrated measures of WMH volumes. Eventually, we have further proved the role of WMHs as early-stage biomarker for AD dementia.

Key-words: White Matter Hyperintensities; Magnetic Resonance Imaging; Neurodegeneration; Alzheimer's Disease; Segmentation; BIANCA; Machine Learning.

Abstract in italiano

CONTESTO - La crescita dell'aspettativa di vita avvenuta negli ultimi decenni ha contribuito ad aumentare l'incidenza di Patologie Neurodegenerative (ND) e legate all'età, come il declino cognitivo, la Sclerosi Multipla, il morbo di Parkinson o di Alzheimer (AD). In questo contesto, una diagnosi allo stadio iniziale è spesso associata a migliori percentuali di successo sia nel limitare la progressione della malattia sia nel prevenire lo sviluppo dei sintomi. Pertanto, è stata esercitata una forte pressione sulla comunità scientifica e medica per identificare biomarcatori sia per il rilevamento che per il monitoraggio delle ND. Tra questi, le White Matter Hyperintensities (WMH), hanno recentemente acquisito sempre più importanza. Queste aree di intensità anormale sulla risonanza magnetica (MRI), una volta venivano trattate come semplici artefatti o segni neurologici di invecchiamento sano. Tuttavia, le ultime scoperte hanno dimostrato la loro associazione con demielinizzazione e perdita assonale e hanno collegato la loro presenza a un aumentato rischio di sviluppare deterioramento cognitivo insieme a diverse ND. La segmentazione WMH è particolarmente importante sia in ambito medico che di ricerca e negli ultimi vent'anni sono state sviluppate diverse strategie di Machine Learning (ML) volte a segmentare e quantificare automaticamente il loro volume. Tra questi troviamo BIANCA, uno strumento automatico sviluppato dall'Università di Oxford basato sul metodo k-NN. Nonostante sia ampiamente conosciuto e malgrado i notevoli sforzi di ottimizzazione che sono stati recentemente effettuati per migliorarne i risultati, alcuni aspetti rimangono ancora inesplorati e aprono la strada a ulteriori miglioramenti.

OBIETTIVI - Lo scopo di questa tesi è duplice. Da un lato, ci proponiamo di valutare le prestazioni di BIANCA secondo diversi parametri quali: i) il numero di soggetti utilizzati per la fase di Training; ii) le diverse combinazioni di contrasti MRI coinvolti nel processo; iii) la diversa strategia di Training utilizzata (ad esempio, Training e Testing su immagini provenienti da database diversi contro Training e Testing su immagini dello stesso database). Questo al fine di trovare la giusta combinazione di valori in grado di ottimizzare la segmentazione finale delle WMH. Durante l'esecuzione della terza fase è stato utilizzato un Training Set da un precedente studio di armonizzazione (che ha coinvolto le popolazioni di Whitehall e UK Biobank), consentendo così una convalida simultanea della sua applicabilità. Inoltre, abbiamo mirato a dimostrare il ruolo delle WMH come biomarcatori in fase

iniziale per la demenza di AD. Ciò è stato valutato allenando un algoritmo di classificazione sia con i dati clinici che con quelli di imaging (cioè i volumi WMH estratti utilizzando BIANCA) e il cui obiettivo è quello di stimare il tasso di demenza clinica (CDR) dei soggetti affetti da AD. Infine, è stata valutata l'importanza esercitata dalle WMH sulla classificazione finale.

METODI - Per raggiungere il primo obiettivo della tesi abbiamo scaricato 40 soggetti con scansioni FLAIR, T1-weighted e Susceptibility Weighted Imaging (SWI) dal database OASIS3. Le immagini sono state pre-elaborate (ovvero, estrazione del cervello, correzione del biasfield e registrazione a una modalità di risonanza magnetica comune), segmentate manualmente e infine utilizzate per allenare BIANCA. Le sue prestazioni sono state valutate secondo i seguenti passaggi: i) Training su un pool di soggetti incrementali (10, 15, 20, 25, 30) e Testing sull'intero set di dati utilizzando un approccio di validazione 4-fold; ii) Training e Testing con le seguenti combinazioni di modalità di risonanza magnetica: "FLAIR only", "FLAIR + T1w", "FLAIR + SWI", "FLAIR + T1w + SWI" e "SWI only" sull'intero set di dati utilizzando un approccio di validazione Leave-One-Out; iii) Training di BIANCA sul Training Set esterno fornito da un precedente studio di armonizzazione e Testing sulle immagini di OASIS3 (approccio Training misto). Per quanto riguarda l'ultimo passaggio, le prestazioni sono state confrontate sia con la letteratura esistente che con i risultati dell'analisi incrementale del Training Set (approccio Training a singolo sito). La metrica di valutazione utilizzata è stata il Dice Similarity Index (DICE).

Per raggiungere il secondo obiettivo di questa tesi, le variabili cliniche di OASIS3 sono state accoppiate alla loro controparte di imaging (cioè i volumi WMH estratti utilizzando BIANCA) per addestrare e testare tre diversi modelli di classificazione: Support Vector Machine (SVM), Artificial Neural Network (ANN), Random Forest Classifier (RFC). Le prestazioni sono state valutate tramite i punteggi di accuratezza, precisione, recall e F1-score. L'importanza esercitata dalla diversa variabile di input sulla classificazione finale è stata invece valutata attraverso il punteggio di "permutation importance".

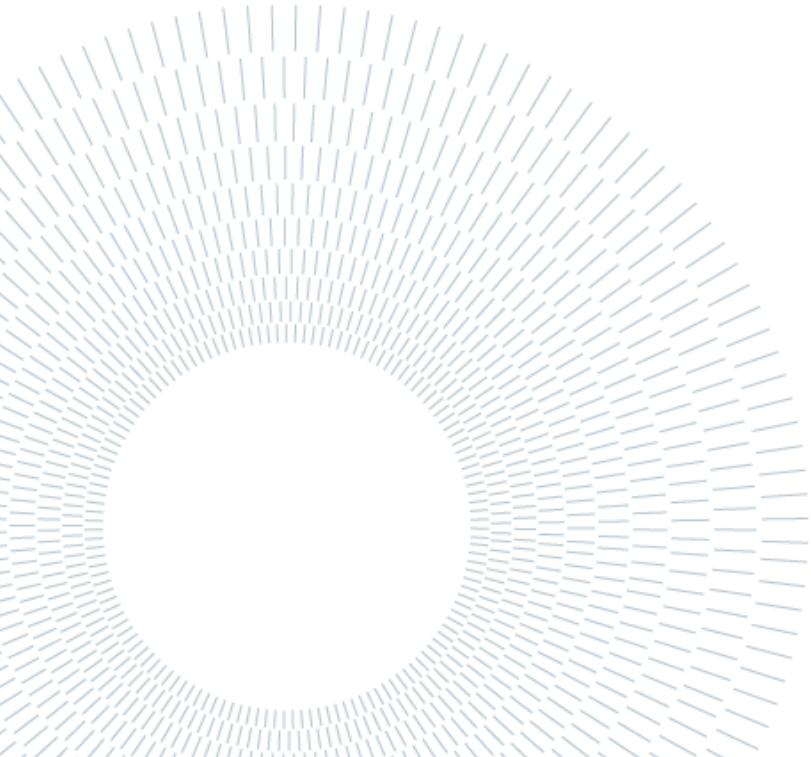
RISULTATI – Per quanto riguarda i risultati BIANCA, è possibile notare un aumento delle prestazioni aumentando il numero soggetti al Training. Tuttavia, si raggiunge un plateau quando questo numero si avvicina al valore di 20 o 25 a seconda dell'accuratezza delle segmentazioni manuali utilizzate durante il Training. Il contrasto SWI utilizzato non è in grado di fornire alcuna informazione utile, portando a risultati insufficienti se utilizzato da solo e ad un peggioramento delle prestazioni se combinato con altri contrasti. Infine, quando testato su OASIS3, BIANCA allenato con i dati di Whitehall e UK Biobank ha ottenuto prestazioni

comparabili con quelle della letteratura esistente. Inoltre, l'approccio Training a singolo sito (rappresentato dall'analisi del Training incrementale) ha mostrato risultati migliori rispetto a quello misto (ossia, Training Set proveniente dallo studio di armonizzazione) solo quando il numero di soggetti coinvolti nel Training è stato maggiore o uguale a 20.

Per quanto riguarda il modello di classificazione, SVM ha mostrato prestazioni accettabili e ha indicato le WMH come una delle variabili più rilevanti utilizzate per prevedere il CDR. Ciò ha confermato l'importanza di questo segno neurologico come caratteristica distintiva di imaging per la demenza AD.

CONCLUSIONI – Con questi risultati abbiamo ottimizzato la strategia di segmentazione automatica rappresentata da BIANCA. Inoltre, abbiamo convalidato una pipeline di armonizzazione preliminare creata per derivare misure integrate dei volumi WMH. Infine, abbiamo ulteriormente dimostrato il ruolo dei WMH come biomarcatori in fase precoce per la demenza AD.

Parole chiave: White Matter Hyperintensities; Risonanza Magnetica Nucleare; Neuro-degenerazione; Morbo di Alzheimer; Segmentazione; BIANCA; Machine Learning.



Contents

Abstract ..... **i**

Abstract in italiano ..... **v**

Contents ..... **ix**

1 Introduction **13**

1.1. White Matter Hyperintensities 14

1.2. Pathophysiology 15

1.3. Clinical context 16

1.4. OASIS3 17

1.5. Machine Learning 18

1.6. Machine Learning in medical fields 19

1.7. Thesis aims 20

2 Methods **23**

2.1. Software 23

2.1.1. FSL 23

2.1.2. Jim8 26

2.2. Machine Learning 27

2.2.1. K nearest neighbors 29

2.2.2. Decision Tree and Random Forest 29

2.2.3. Support Vector Machines 31

2.2.4. Artificial Neural Networks 32

2.2.5. Principal Component Analysis 34

2.2.6. Google Colab 35

3 Materials and Methods **36**

3.1. Evaluation of BIANCA performance 36

3.1.1. Download procedure and Patients selection 37

3.1.2. WMH masks creation 41
3.1.3. Imaging data organization 45
3.1.4. Images preprocessing 46
3.1.5. Running BIANCA 53
3.2. The Classification algorithm 60
3.2.1. Dataset preparation 61
3.2.2. Model creation 67
3.2.3. Machine Learning metrics 73
3.3. Statistical analysis 75
3.3.1. ANOVA test 75
4 Results 77
4.1. Evaluation of BIANCA performance 77
4.1.1. Incremental training analysis 78
4.1.2. Intra-rater variability 83
4.1.3. Multimodality analysis 84
4.1.4. Harmonization pipeline 87
4.2. Classification Model 90
4.2.1. Data pre-processing 90
5 Discussion and conclusions 102
5.1. Conclusion 106
Bibliogaphy 109
A Appendix A 115
5.2. Spreadsheet filtering 115
5.3. Renaming external 116
5.4. BET run 117
5.5. Fast run 118
5.6. FLIRT run 119
5.7. BIANCA masking 120
5.8. Masterfile creation 121

5.9. BIANCA training 122
5.10. BIANCA testing 141
5.11. BIANCA performance 146
B Appendix B 151
5.12. FLAIR T1w run 151
5.13. Dataset join 154
List of Figures 156
List if Tables 161
Acknowledgments 165

1 Introduction

Neurodegenerative Diseases (NDs) affect millions of people worldwide with an incidence significantly higher in elder individuals. The increase of life expectancy occurred over the past few decades has contributed to their spreading diffusion, thus imposing significant pressure on the health care systems of most countries and increasing social and economic costs. Current treatments are able to relieve some of the physical and mental symptoms associated with these conditions and contribute in slowing their progression. Unfortunately, though, they are incapable of reverting any of the morphological or microstructural changes induced by NDs. Their permanent nature has attributed enormous importance to early-stage diagnosis, which significantly impacts on the subject quality of life by allowing a timely access to cures and a prompt management of the symptoms. Therefore, the scientific community is highly investing in the research of neurological markers of both ND onset and progression. Amongst these, White Matter Hyperintensities (WMHs) – a common finding on brain Magnetic Resonance Imaging (MRI) – have recently gained more and more importance as early neuroimaging sign for several neurological and vascular conditions, among which we can find Alzheimer's disease (AD), Parkinson's Disease (PD), etc.. Numerous attempts have been made to automatically segment WMH volume to quantify the lesional load. However, most of the developed machine learning (ML) solutions lack both generalization capabilities and fully optimized performance, thus hindering their wide application to medical or clinical contexts. A major issue is the poor absolute quantification of MRI contrast, which is strongly machine-dependent, thus requiring suitable harmonization procedures in large studies aiming at the fixation of WMH load quantification standards (Bordin et al. 2021).

Importantly, ML outcomes are severely conditioned by the training process and efforts are devoted both to its improvement and its validation. Namely, this thesis addresses BIANCA, of the FSL open package developed by the Oxford University. BIANCA was specifically developed for WMH segmentation, based on fully automated and supervised k-NN clustering. Most application address mono-modal MRIs (e.g., FLAIR, T1, T2); nonetheless, BIANCA admits multimodal training and application, which feature is a major point in this study.

BIANCA performance will be assessed on a population of cognitive declined AD subjects according to different parameters such as: the number of training subjects, the combination of MRI modalities and the training strategy utilized. Ultimately, we also aim at validating the role of WMHs as early-stage diagnosis biomarker for the AD. In this perspective, the WMH load will be compared to clinical variables (including dementia severity levels) aiming at the assessment of the predictive power of MRI outcomes.

1.1. White Matter Hyperintensities

WMHs, also known as Leukoaraiosis (Hachinski, Potter, and Merskey n.d.), are brain lesions usually associated to normal aging that have recently gained importance as neuroimaging biomarkers for several neurological and vascular conditions (Morris et al. 2009).

They are particularly noticeable on both Magnetic resonance Imaging (MRI) and Computerized Tomography (CT), however, the former ability to better contrast changes in the soft tissues allows for an optimal identification of these structures and makes MRI the standard diagnostic tool to reveal their presence.

WMHs are a relatively new discovery in neuro-radiology (F. Fazekas et al. 1993a) due to the following reasons: first, until recently, MRI scanners could not generate magnetic fields with intensities higher than 0.5 Tesla, therefore being less sensible to small changes in the local intensity of the field and bringing very little influence on the resulting images; secondly, WMHs are much more noticeable in FLAIR images, but this type of contrast was less used due to its lower signal/noise ratio(Wardlaw, Valdés Hernández, and Muñoz-Maniega 2015). Nonetheless, later advancements in MRI methodologies have allowed for a well-rounded assessment of the lesioned areas, enabling their adequate contrast with respect to surrounding tissues and reestablishing FLAIR as the most sensitive and utilized sequence for WMH detection.

The presence of lesioned areas influences the local intensity of the magnetic field during scans and shifts the recovery time, thereby creating hyperintense zones on T2-weighted, T2*-weighted, FLuid Attenuated Inversion Recovery (FLAIR) and Proton Density-weighted (PD) images. Conversely, they appear as hypointense areas in T1-weighted scans(Wardlaw et al. 2013).

WMHs are generally bilateral and heterogeneously distributed, and their volumetric extent reflects advancements in normal aging or pathological processes. An example of their aspect can be found in Fig. 1.1.

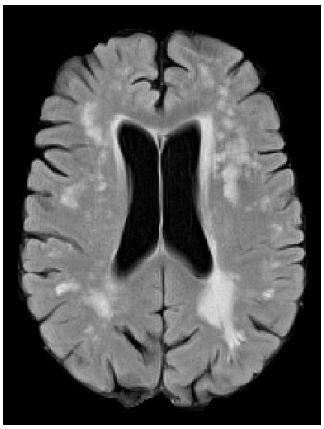


Figure 1.1 Wide WMH lesions clearly visible on a FLAIR image as white marks.

1.2. Pathophysiology

Pathology studies are unfortunately less frequent than imaging analysis investigating the WMH morphology (Gouw et al. 2011; Moran, Phan, and Srikanth 2012). This is due to a significant difficulty in linking these neurological findings to their histological counterpart (Black, Gao, and Bilbao 2009; Shoamanesh, Kwok, and Benavente 2011). However, existing literature have reported signs of demyelination and axonal loss in areas where WMHs lesions are most present(F. Fazekas et al. 1993b), further indicating those changes as being permanent.

In addition, WMHs have been associated to lesions of the small blood vessels, causing micro-bleedings with ferritin and calcium deposits in different regions of the brain. Besides, there are reports indicating the presence of proteins often seen in cases of Small Vessels Diseases (SVDs) (Tomimoto et al. 1996) and thickened capillary walls in areas where hyperintense lesions are visible(Munoz 2003).

Many in vivo studies have also reported a higher albumin presence in advancing age subjects or in patients affected by Alzheimer's disease, vascular dementia and WMHs, suggesting an important role for chronic brain edema in the development of such conditions (Farrall and Wardlaw 2009).

The pathophysiology of WMHs can be seen in various types of MRI contrasts: the T1 relaxation time and the mean diffusivity do increase in lesioned areas (Bastin et al. 2009; Franz Fazekas et al. 2005; Maniega et al. 2015; O'Sullivan 2004), highlighting

the presence of pathological processes such as increased water content and mobility, which adds to the above-mentioned demyelination and axonal loss. Furthermore, perfusion MRI shows a great decrease of the Cerebral Blood Flow in regions underlying the presence of lesioned (Markus 2000; O'Sullivan et al. 2002).

Ultimately, a notable distinction can be made in the pathological connotates of WMHs according to their volumetric extent: restrained lesions are indeed associated to microglial and endothelial activation, while broader ones present vacuolation and reduced density of the glia (Wardlaw, Valdés Hernández, and Muñoz-Maniega 2015).

1.3. Clinical context

Despite being a precocious symptom of the advancing age, in the analysis of many longitudinal studies WMHs have been correlated with progressive cognitive impairment, a two times higher possibility of developing dementia and a three times higher risk of being affected by stroke(Debette and Markus 2010). Late research has defined them as symptoms of various SVDs such as small subcortical ischemic and hemorrhagic stroke, microbleeds and brain atrophy(Wardlaw et al. 2013; Wardlaw, Smith, and Dichgans 2013). However, WMHs also have important associations with depression, disability, mortality, and can be found in patients affected by autoimmune disorders or psychiatric conditions(Balakrishnan, Valdés Hernández, and Farrall 2021).

The presence of WMHs is prominent in patients with vascular risk factors such as hypertension(Dufouil et al. 1991; Maillard et al. 2012), smoking(Gons et al. n.d.; Staals et al. 2014), high cholesterol and diabetes(Ferguson et al. n.d.). Nonetheless, genetic factors also play a very important role in understanding the origin and association of WMHs. Their presence is indeed highly hereditable(Turner et al. 2004), lower in patients with long-lived parents(Altmann-Schneider et al. 2013) and inversely proportional to higher intelligence in youth(Valdés Hernández et al. 2013).

To conclude, the clinical context around these neurological signs is highly heterogeneous and highlights the need for further research to extensively evaluate both the impact and diagnostic/prognostic role they exert on the health conditions of participants.

Many different factors have limited recent WMHs studies: starting with the need for artifacts detection and removal, we move on to the requirement for extensive

images processing and conclude with a severe lack of quick and reliable approaches to evaluate the severity of lesions according to their volume extent. Indeed, manual rating procedure are cumbersome and time-consuming(Wardlaw, Valdés Hernández, and Muñoz-Maniega 2015), additionally requiring the service of expert operators specifically trained to recognize WMHs.

The need for semi-automated and automated approaches allowing their identification is therefore mandatory to provide clinician and researchers with a deeper level understanding around the origin and progression of these hallmarks. Over the past few decades, this demand has paved the way for the development of several Machine Learning (ML) strategies, which have attempted to segment and quantify WMHs, progressively optimizing the obtained results. In the present study, we investigate such methods on a large population of normal aging and neurodegenerative subjects: the OASIS3 database.

1.4. OASIS3

The third release of the Open Access Series of Imaging Studies (OASIS3) is a longitudinal cohort of data which collected, over the course of thirty years, thousands of medical records across several different research projects(LaMontagne et al. 2019). It was built aiming to provide the scientific community with a large and freely available neuroimaging dataset, complemented by clinical and neuropsychological assessments.

The OASIS3 focuses on the effects of both normal aging and early-stage Alzheimer's Disease (AD) including a compilation of data from 1076 participants. Out of those, 605 were cognitively normal adults (Healthy Control (HC) group), while the remaining 493 were affected by various stages of AD cognitive decline (AD group).

As regards the imaging sub-part, the dataset includes over two thousand MRI sessions with a combination of various contrast: T1-weighted, T2-weighted, FLAIR, susceptibility weighted imaging (SWI), diffusion weighted imaging (DWI) and many more. The acquisition process was conducted at the Knight Alzheimer Research Imaging Program at the Washington University in St. Louis, using a 3T scanner.

Many of the MRI sessions were also accompanied by volumetric segmentations produced using Freesurfer. PET scans were also available, but resulted of minor interest for the purpose of our analysis.

On the other hand, the clinical information was of extremely relevance. There are over six thousand records present in OASIS3 containing information about the subjects' demographic, habits, medical history, and cognitive status, thus allowing the user to have a well-rounded view over their health status. In this context, the Clinical Dementia Rate (CDR) index revealed to be of particular importance, enabling a stratification of the AD group in several levels according to severity of their cognitive decline. The score ranges from 0 (cognitively normal) to 2 (moderate dementia) and has intermediate values equal to 0.5 (very mild impairment) and 1 (mild cognitive impairment).

All the available OASIS3 data is hosted by the XNAT central repository (central.xnat.org) and the complementary information can be found via https://www.oasis-brains.org/. Unfortunately, clinical variables and MRI scans are in no way matched: subjects can have more clinical records than imaging sessions (or vice versa) and often those information pieces are assessed in different days. Therefore, a precise coupling strategy needs to be defined prior to the beginning of any analysis. Nonetheless, the great variety of data contained in this database, allowed us to exploit multiple ML approaches to both optimise current WMH segmentation strategies and to build robust predictive models focused on AD.

1.5. Machine Learning

ML is a branch of computer science focused on algorithms and their ability to improve automatically through experience and data usage. As a consequence, this group of techniques benefits from large repositories rather than small gatherings of information: with more "examples" at their disposition algorithms are able to recognize in a better and faster way the patterns that link different types of variables, hence allowing to accomplish either Regression, Detection, Classification or Clustering tasks. ML has grown more and more important in many fields thanks to its adaptability and capacity to perform complex functions such as data prediction, image enhancement, feature detection and many others. However, at the beginning, many downsides have prevented its widespread applicability. First, there was very little information and experience around them due to their recent development. The low level of comprehension and the algorithms outline, often unclear or inaccessible, contributed spreading a feeling of distrust towards their use. Secondly, the need for heavy computational power, alongside the great amount of

data and time required by algorithms to "learn" tasks have discouraged the integration of such solutions in most practices.

Nevertheless, recent improvements in the hardware, together with the advent of faster internet connection and larger storage capacity, have made ML more accessible than ever.

A steep increase in its usage has been witnessed over the past few years and has led to enormous advancements in many fields. The significant computational time reduction has allowed to complete activities that used to require hours or days in a bunch of minutes, sometimes even seconds.

As a consequence, plenty of algorithms are nowadays integrated in everyday applications: from email spam blockers to the image analysis systems embedded in smartphones for automatic facial recognition; from Artificial Intelligence (AI) to make videogames more entertaining and unpredictable to the automated control of medical devices. Life has substantially been simplified.

1.6. Machine Learning in medical fields

Among the most popular contexts where ML has been exploited, we can mention the medical field, where ML has grown consistently thanks to a multitude of factors. As anticipated, the time-saving attribute of automated algorithms is especially helpful, allowing clinicians to focus on other tasks while huge amounts of data, that may help them in the decision-making process, are elaborated. An example is represented by algorithms trained to provide medical diagnosis based on information about patient symptoms. By feeding those data to a dedicated ML tool it's possible to look for similarities among the medical records of former hospitalized subjects and to formulate suggestions. This saves significant time and effort to the medical team in charge of monitoring their health conditions. But the potential of ML does not stop here.

Another important task for which it is massively adopted within medical frameworks is the processing of diagnostic images. Resolution augmentation is amongst the most impressive and helpful ability these techniques can provide(Dong et al. 2016; Yano and Watanabe 2020). An example is given by MRI acquisitions, that always require lots of time being the result of a complex 3D or even 4D reconstruction processes. However, the advent of ML has allowed to use lower resolutions and to later upscale them through the use of automated

approaches. This preserves quality while reducing the overall amount of time necessary to derive images.

Alongside this aspect, many additional processing techniques are currently involved in the modern medical practice by means of ML. Among them we can find: adaptive noise filtering, image sharpening, automatic contrast regulation, and the adjustment of x-ray beam geometries in real time tumor tracking contexts.

An important role is also played by the automated segmentation of relevant anatomical areas, in particular for the case of lesions or tumors(Shin et al. 2016). It can take hours to correctly identify those tissues with respect to the surroundings and it requires an expert manual operator to focus his energies on a process that nowadays can be accomplished very rapidly by algorithms. The manual reference still remains a difficult gold standard to reach, but recently developed solutions have pushed performance to competitive levels, generally under the name of Radiomics.

Among the anatomical structures that are more relevant to identify, this thesis focuses on WMHs, whose importance as neurodegeneration biomarker has been discussed in the previous sections. Since they were discovered only recently, the ML algorithms developed for their segmentation are limited in number, not easily accessible and often incomplete, being in beta version or even discontinued. Additionally, they are sometimes tested only on small samples of data and therefore result as highly dependent on the specific protocol or study involved in their own development.

In an attempt to bridge these gaps, the Brain Intensity AbNormality Classification Algorithm (BIANCA) was developed in 2016 by the Oxford Centre for Functional MRI of the Brain (FMRIB). It is a tool based on the k-Nearest Neighbor (k-NN) algorithm that comes fully supported, easily obtainable and entirely free. However, despite the significant optimization endeavors that have been recently carried out, some aspects still remain unexplored and open the lead for further improvements.

1.7. Thesis aims

As previously introduced, the general purpose of this thesis is twofold: on one hand, we attempt to optimise the WMH segmentation by means of existing ML strategies, while on the other we try to validate their role as neuroimaging biomarkers for the cognitive decline process associated to AD.

As for the first part, we take into account BIANCA, a tool which currently offers the best solution for the automatic identification of such lesions. From this starting point we try to move forward and work on some of the pitfalls and unexplored aspects that are preventing its widespread application to both clinical and medical contexts.

First, no prior study has ever investigated the relationship between BIANCA results and the number of subjects used for the training, except when considering its dependence from the WMH load of the involved participants (Griffanti et al. 2016). The relationship within a uniform group (being either 'high', 'low' or 'any' WMH load) remains unexplored despite the significant information it could provide to calibrate our data requirement. For this reason, we will train BIANCA using images from an incremental number of individuals and eventually compare the obtained results looking for the presence of a potential plateau.

Secondly, the algorithm performance are yet not fully optimized. This is particularly true when it comes to the need of harmonizing data derived from heterogeneous studies or different cohorts. Indeed, one of the latest solutions designed to remove the non-biological variability from data (Bordin et al. 2021)has highlighted a significant drop in segmentation performance as a result of better data integration. Aiming to overcome this limitation, we will try a multimodality approach, which has revealed being promising in the same study (Bordin et al. 2021). We will train and test BIANCA using different combinations of MRI scans, introducing novel contrasts such as SWI, alongside more conventional ones like FLAIR and T1.

Ultimately, we will compare results from the multimodality strategies (derived after training on the OASIS3 dataset) with those obtained training BIANCA on data from the Whitehall (WH) and UK Biobank (UKB) cohorts. The aim for this step is twofold. First, we want to assess whether the performance obtained using a single-site training (considered as the optimal case), are either better or worse with respect to the mixed training case represented by WH+UKB (considered sub-optimal but acceptable). This will serve as a reference to evaluate how the improvement or worsening provided by certain multimodal combinations do relate with respect to a well-known threshold. Secondly, the WH+UKB training set – derived from the above-mentioned harmonization study (Bordin et al. 2021)– has not been fully validated yet, having been tasted only on the populations involved in its own development. Therefore, we will take this chance to compare the performance obtained training BIANCA on WH+UKB and testing on OASIS3 with that of the existing literature.

As for the second part, we will use the population at our disposition to test common prior knowledge about the diagnostic role of WMHs in neurodegenerative contexts such as AD (Debette and Markus 2010; F. Fazekas et al. 1993a; Wardlaw, Valdés

Hernández, and Muñoz-Maniega 2015).We will pursue this goal by means of a Classification algorithm, that we will be trained to predict the patients cognitive decline status (identified using the CDR score) based on both clinical and imaging information (this latter being represented by the volumetric amount of WMHs derived using BIANCA). However, our ultimate goal won't be represented by the classification performance itself, which nonetheless will be pushed to improved levels, but by a thorough evaluation of the amount of variance explained by the different variables fed to the model. Indeed, the contribution provided by WMHs to the final prediction could either confirm or challenge their importance as neuroimaging hallmark for the early-stage diagnosis of AD dementia.

2 Methods

In this section we present the major tools, libraries and techniques that were used to carry out the analysis. The different steps undertaken will then be discussed in the following chapter.

Here we introduce the software used for both data pre-processing and the manual/automatic segmentation of WMHs. Then we move to the machine learning algorithms lying behind such tools and eventually to the ones we used to construct the predictive model for clinical data.

2.1. Software

As regards the utilized software we present both FSL and JIM8.

2.1.1. FSL

FSL is a comprehensive library of analysis tools for FMRI, MRI and DTI brain imaging data that was developed and is currently updated by the Analysis Group of the Oxford FMRIB laboratory (Jenkinson et al. 2012; Smith et al. 2004; Woolrich et al. 2009). FSL comprehends a wide variety of programs for visualization, segmentation, registration and processing of brain images, that will be briefly summarized here. Alongside those, it also provides BIANCA, an automated software for WMH segmentation, that is presented below.

FSL can run either on Linux or Apple devices, while there is currently no release for Windows operating systems. Nonetheless, it can also be used on these latter thanks

to the Windows Subsystem for Linux, an integrated Windows tool released by Microsoft.

2.1.1.1. Tools

FSL includes different utilities for the analysis of brain images, each with its specific role. Among the most important ones we can find: BET, FAST, FLIRT and FNIRT.

BET (Brain Extraction Tool) is an automated tool for brain extraction which deletes non-brain tissue from images of the whole head (Smith 2002) It can also estimate the inner and outer skull surfaces, if good quality of the input images is provided (Parker Jones, Alfaro-Almagro, and Jbabdi 2018).

FAST (FMRIB's Automated Segmentation Tool), on the other hand, does segment 3D images of the brain into different tissue types (Grey Matter, White Matter, CSF, etc.), whilst also correcting for spatial intensity variations (known as bias field (BF) or RF inhomogeneities) (Zhang, Brady, and Smith 2001). The whole process is fully automated and produces as outcomes a bias field-corrected version of the input image and its probabilistic and/or partial volume tissue segmentation. FAST is robust and reliable, compared to most finite mixture model-based methods, which are sensitive to noise.

FLIRT(Greve and Fischl 2009; M Jenkinson 2002; Mark Jenkinson and Smith 2001) and FNIRT are two fully automated and accurate tools to perform linear (affine) and non-linear registration of brain images that allow to work both in an intra- and inter-modality fashion.

Alongside these functions, FSL does provides a tool for visualizing neuroimaging data which is capable of handling either 3D images, timeseries or surfaces: FSLeyes (McCarthy 2021).

Eventually, fslutils and fsl_anat are also worth of mention. The first is a miscellaneous command-line programs including fslmaths and fslstats which allow for conversion and processing of Analyze and Nifti file formats.

The second is a general pipeline for processing anatomical images that involves a standard use of the FSL tools but allows for an improved correction of the bias-field. This is particularly helpful for multi-coil arrays and high-field scanners, where inhomogeneities as most frequent.

2.1.1.2. BIANCA

BIANCA is a fully automated and supervised method for WMH detection, based on the k-NN algorithm (Griffanti et al. 2016), which will be explained below. It works by classifying the image's voxels based on their intensity and spatial features and provides as outcome a probability map where each voxel is assigned to a value representing its probability to be part of a WMH lesion.

In order for BIANCA to perform optimally a thorough data preparation is required. Firstly, since the tool works with multiple MRI modalities but in single subject's space, all the input images need to be registered to a common MRI scan, selected as reference (e.g., FLAIR). In this way, they are all reduced to the same dimension in terms of both resolution and FOV. The MRI scan normally selected as base image, is the FLAIR. Additionally, to calculate spatial features without the presence of any bias, a registration matrix from base image to standard MNI space needs to be derived. This step is not necessary in case that couple of scans is already aligned. The second important operation that must be performed is the brain extraction of at least one of the available MRI modalities. This allows to derive a binary mask of the subjects' brain inside which BIANCA will search for lesions. Eventually, to avoid the presence of false positive, it could be helpful to apply an additional mask to remove regions mostly affected by artifacts of that may appear as hyperintense on FLAIR. This ulterior masking is usually done by removing the cortical GM and structures such as putamen, globus pallidus, nucleus accumbens, thalamus, brainstem, cerebellum, hippocampus, amygdala, which deeply affect BIANCA performance.

Once images have been properly processed, the algorithm training phase can begin. BIANCA requires a set of subjects with manual WMH masks and – for each – it selects a certain number of lesion and non-lesion points on the different MRI modalities provided. For each of these pre-classified voxels, the following features are then extracted: punctual intensity, local intensity averaged on a 3D patch of pre-specified dimension, and location of the spatial coordinates in the MNI standard space. This operation allows to create a set of feature vectors for both lesion and non-lesion classes. Each of them will eventually be reported to a k-dimensional space (namely the k-space) where every axis corresponds to one of the extracted features.

Once the training phase is completed, BIANCA is ready to be used. The algorithm works by projecting all of the voxels contained in the input image to the k-space and

by calculating their probability to be a WMHs based on majority voting from the k-nearest training points. The resulting output is therefore a probability map, made of continuous values ranging from 0 to 1.

To obtain binary masks able to distinguish lesioned from healthy tissues, a thresholding operation is finally required. Alternatively, supervised methods such as LOCATE (LOCally Adaptive Thresholds Estimation) can be used to automatically determine the optimal thresholds for each of the different regions present within the brain. LOCATE is currently only available as beta release in MATLAB, but will certainly be integrated as part of the FSL library in future publications.

Eventually, the automatic WMH mask created with this procedure is compared with a WMH mask derived by a rater (considered as gold standard) to evaluate the obtained performance.

Few general considerations are also worth mentioning before concluding. First, when using BIANCA, testing subjects must provide the same combination of MRI contrast with respect to the ones involved in training. For example, if the latter was conducted using both FLAIR and T1, the algorithm can only work if both scans are available for testing. On the contrary, a single MRI modality would give rise to an error message, while any different pair (e.g., FLAIR and SWI) would lead to very imprecise results. Secondly, BIANCA offers great flexibility as for the choice of different options like: the weighting factor attributed to spatial information, the local intensity averaging technique and the number and location of training points. The final output will therefore critically depend on their values. Alongside this, the quality of training data and of their manual segmentations also plays a very important role. Automated tuning methods and pre-trained datasets will hopefully be available in an upcoming release but, currently, the optimal solutions still need to be identified according to the dataset under analysis.

2.1.2. Jim8

Jim8 is a medical image display package developed by Xinapse (https://www.xinapse.com/) that allows for an easy viewing and easy analysis of MRI, CT and other types of medical image. It can handle a wide variety of data formats, including DICOM, Analyze, NIFTI-1, NIFTI-2, and it is geared towards the analysis of multi-slice and multi-dimensional imaging sets.

Jim8 contains the tools needed for most common analysis tasks, such as:

- Creation of Region Of Interest (ROIs)
- Image fitting
- Image masking
- Fat/Water separation
- Image registration

It has an up-to-the-minute design with a familiar user-interface and runs natively on any 64 bit Operating System. However, a Java installation needs to be performed for its utilization.

For the purposes of this thesis Jim8 was exclusively used for its powerful and easy to use segmentation utility that allowed us to manually extract ROIs representing WMH lesions.

Once the ROI toolkit (https://www.xinapse.com/Manual/roi_sections.html) is loaded, users can highlight regions over any diagnostic image by using different methods such as circular, rectangular, spline, or irregular contouring. In case of WMH segmentation, the most useful option is represented by the latter. Starting from the pixel/voxel selected with the cursor, the irregular method tries indeed to automatically include in the ROI all the surrounding voxels that match a certain intensity similarity criterion. This strongly facilitates the segmentation procedure, which can eventually be refined by using the 'Edit' tools present within the utility. This selection method can also be aided by specifying in the appropriate banner which kind of contour the algorithm has to look for, expressing whether the ROI is going to be brighter or darker with respect to background. As for WMHs, like the name suggests, lesions will always be brighter than the other tissues.

Finally, since BIANCA needs binary lesion masks as reference to operate, the Jim8 Masker Tool (https://www.xinapse.com/Manual/masking.html) must be used right after the ROIs creation. This utility allows to convert the highlighted regions in NIFTI binary files by removing the background voxel outside its contours (which are set to 0), while retaining the ones inside of them (which are set to 1).

2.2. Machine Learning

After discussing the importance of ML and the great advancements it has recently allowed (see Chapter 0.3, "Introduction"), in this section we will deal with the technical aspect of algorithms. Afterwards we will proceed by illustrating examples from some of the ML techniques exploited in the context of this thesis.

The first thing to know about ML algorithm is they can all be inserted in one of the three following categories: Classification, Regression or Clustering.

The aim of Classification algorithms is to learn a function that can be used to predict the correct category for new input data, based on the characteristics of a given training set. An example is represented by automatic email spam detectors. They are able to classify an incoming mail as being either spam or not based on many input information like the presence of links inside of it and the sender's address.

Clustering, on the other hand, is a branch of ML very similar to Classification, but somehow yet different. As for the latter, predefined labels are assigned to input instances according to their properties and are subsequently used to train the algorithm. Conversely, in Clustering tasks those labels are not known a-priori and data are assigned to a specific class according to certain similarity criteria. Therefore, Clustering methods usually group variables using algorithms which are less complex than the ones involved in Classification.

Eventually, Regression algorithms differ from both of the above-mentioned methods as they do not have any class to output, but instead are trying to predict a continuous value. In most cases, Regression strategies attempt to find the inner relationship that links certain input variables to an output that needs to be forecasted. Among common applications we can fins weather prediction: given air pressure, wind currents, humidity and other variables collected within the last few hours or days, algorithms manage to predict the temperature which is about to be reached.

It is also important to highlight that some Regression algorithms can be turned into Classification ones. Indeed, the continuous output from a Regression task can be binarized or divided into multiple bands by means of thresholding to obtain either two or more classes.

Despite its own nature, every algorithm needs to be trained before being able to either cluster data, classify them or predict the output value corresponding to unseen observations. The training procedure can either be supervised, or unsupervised depending on the properties of the involved dataset.

Supervised Learning is defined by its use of labeled datasets to train algorithms that try to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights until the model has been fitted appropriately.

On the other hand, Unsupervised Learning uses algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Their ability to discover similarities and

differences in information make it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

For the purposes of this thesis, we mostly focused on supervised ML techniques. As previously stated, BIANCA is based on the k-NN algorithm. On the other hand, to implement the predictive model focused on the health condition of participants, we dealt with Decision Trees, Random Forest (RF), Support Vector Machines (SVM), Artificial Neural Networks (ANN) and the Principal Component Analysis (PCA). These techniques will all be described in the following paragraphs.

2.2.1. K nearest neighbors

The principle lying at the core of the k-NN algorithm is easily understandable and allows to perform both Classification and Regression(Cover and Hart 1967), with very little difference.

The training phase is required and allows the algorithm to learn from data. It is based on a quite simple principle: from each input observation a certain amount of features is extracted and stored into a vector; each vector is then associated to a point in a multidimensional space (namely the k-space), where information about its class are also specified.

As regards the testing phase, a constant parameter, namely *k*, is always involved. Its value needs to be defined by the user and its role is very similar in both Classification and Regression tasks. As for the former, when a new point needs to be labelled, its class is decided according to the one that is most frequent amongst the *k* training samples closest to it in k-space. When it comes to latter, instead of a class assigned by majority voting, the output value (which is continuous) is calculated as the mean of all the k nearest points. From here, the name of the algorithm.

2.2.2. Decision Tree and Random Forest

The Random Forest Classifier (RFC) is a well-known ML technique which, in turn, is based on the Decision Tree algorithm. For this reason, the latter will be explained first (Lepetit and Fua 2006).

Decision Trees are supervised ML techniques that identify ways to split a dataset based on various conditions until an outcome is generated. The main components of a Decision Tree are the following:

- Root: it represents the whole population and will be divided in at least two sets.
- Decision Node: a node which does split in further sub-nodes.
- Terminal Node or Leaf: a node which does not create any other nodes.

The algorithm starts at the Root and compares all the input variables to decide how to split data. Then it creates new nodes. This search is based on entropy or the Gini index of variables and then it continues to generate new nodes until either the maximum depth, which is an hyperparameter set in advance, is reached or the algorithm cannot further identify nodes.

The Gini Index is calculated by subtracting the sum of the squared probabilities of each class from one. Its possible value varies between 0 and 1, where 0 represents purity of the classification and 1 denotes random distribution of elements among various classes. A Gini Index of 0.5 shows that there is equal distribution of elements across some classes.

The Gini index can be seen as a numerical value of the amount of probability of a specific feature that is classified incorrectly when selected randomly. The minimum value of the Gini Index is 0. This happens when the node is pure, this means that all the contained elements in the node are of one unique class. Therefore, this node will not be split again. Thus, the optimum split is chosen by the features with less Gini Index.

On the other hand, Entropy is a measure of information that indicates the disorder of the features with the target and is calculated as $E(S) = -\sum_{i=1}^{c} Pi \log 2 \cdot pi$. This index can have any value between 0 and 1 and is much more computationally intensive than its Gini counterpart, this is the reason that the latter has ultimately more success in Decision Trees.

Using Entropy, the splitting at a node is made trying to minimize the entropy of the two new groups that would be created after the split.

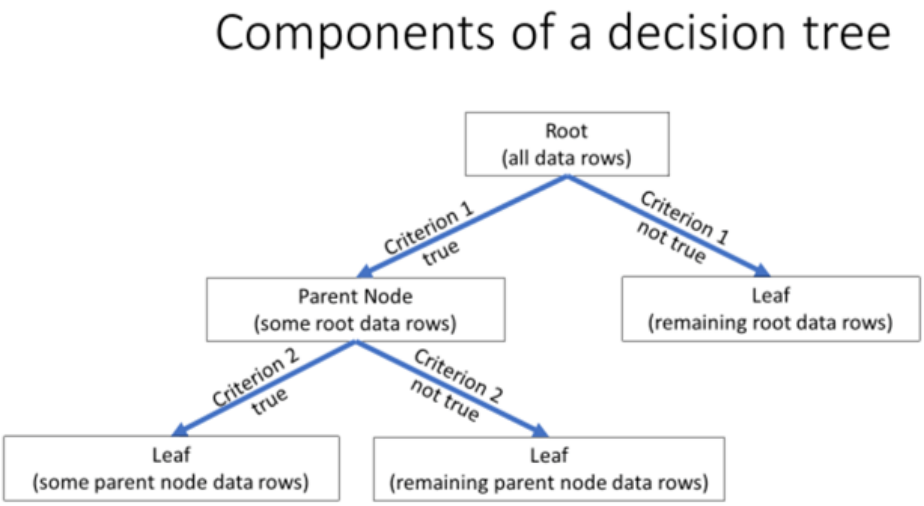


Figure 2.1 A schematic of a Decision Tree

As for the k-NN algorithm, it's important to point out that Decision Trees can also be used to perform both Classification and Regression tasks.

Going back to RFCs, as the name may suggest, they can be described as ML techniques implying the generation of multiple Decision Trees. When a new observation needs to be processed, the algorithm feeds that input to all of the available trees and then determines the output by means of a majority voting process.

2.2.3. Support Vector Machines

SVMs are supervised algorithms used for both Classification and Regression tasks that separate variables using hyperplanes according to the class where they belong (Evgeniou and Pontil 2001).

Their structure is more complex than the one of k-NN or RFC methods. Nonetheless, they are one of the most robust prediction methods, being based on statistical learning: the objective of the SVM is to find a hyperplane in an N-dimensional space (N is the number of features) that optimally separate the points in the training set in their respective classes. The best separation hyperplane is ideally achieved when it has the largest distance from any of the available training points. Once the training phase has concluded, each new point which needs to be classified will be inserted in the space of the hyperplane and its class will be decided compared to the position of the dividing hyperplane itself.

Originally, hyperplanes used to be strictly linear but nowadays, thanks to the so-called "kernel trick", it is possible to shape their outline to better suit the distribution of data points, thus improving class separation. This is achieved by using polynomial, sigmoid or Radial Basis Function (RBF) kernels.

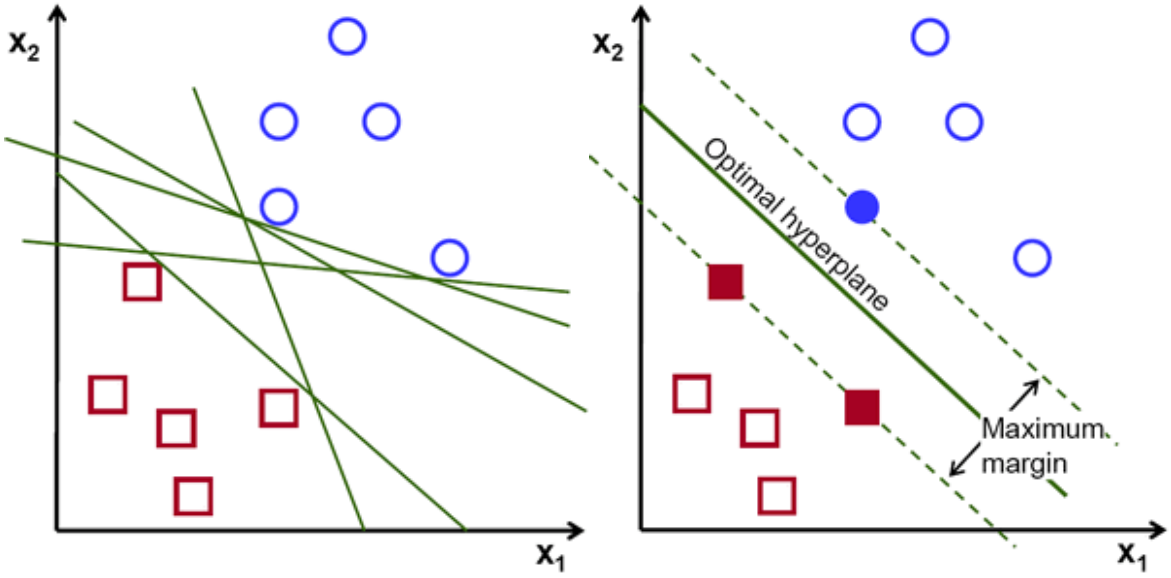


Figure 2.2. An example of SVM.

2.2.4. Artificial Neural Networks

ANNs are a vast typology of ML algorithms in which many nodes called neurons – often linked one another in particular structure-like forms – do perform articulate operations on variables to output either a classification or prediction result (Grossi and Buscema 2007). ANNs are capable of handling very complex tasks and are much more sophisticated than most of the other ML algorithms. On the other hand, they are also hardly "explainable": it is indeed very difficult to understand what kind of operations have allowed a net (i.e. a cluster of neurons) to produce certain outputs.

The simplest ANN is made by a single neuron, called perceptron, which receives a certain number of input variables but returns only one output. The model works as follows: it individually weighs all the inputs and linearly combine them to obtain a polynomial equation; then it feeds this latter into an activation function that, providing the value of the final result, determines whether the neuron is activated or not. The activation function include a wide variety of curves such as sigmoidal,

step or exponential. Their characteristics allow the perceptron to perform non-linear operations.

To create a more complex ANNs multiple perceptrons must be linked together. In this regard, there are different kinds of layers that can be constructed, each with its specific function: the *input layer* is the one receiving the input variables; *hidden layers* (which can be either missing or present in variable number), on the other hand, are the ones where the main processing is done; eventually the *output layer* is the one providing results. Alongside these, additional "optional layers" can also be fund in the most complex ANNs. A first example is given by the *max-pooling layer* which is used to perform downsampling on intermediate layers. Another one is represented by the *softmax*, which is used in multiclass classification tasks to normalize the different output probabilities so that they sum to 1.

The training in ANN is based on the backpropagation of the error, in which the error travels back from the output to the inputs changing the weights of the connections of the neurons. Being a supervised method, the error can for example be an incorrect output class, in that case the difference between the real class and the predicted one (true class is 0, predicted class is 1, the error is -1) is the value that will be propagated to the previous neurons changing their connections weights.

Since ANNs can be created in many different shapes linking perceptrons, diverse networks can be assembled to reach very specific purposes. Among the most famous, we find Convolutional Neural Networks (CNNs), which are constituted by 2D and 3D layers of interconnected neurons that proved being extremely useful in almost any type of image analysis and processing task. Another example is represented by Generative Adversarial Network (GAN), where two different ANNs placed in series do compete: one tries to generate fake data while the other tries to recognize whether the inputs it receives (from both the previous ANN and an external dataset) are original or manufactured. The practicality of GANs is broad: they can help generate missing parts of images, modify videos inserting a character which is not really present or even create "replacements" for certain deleted parts. As for this latter, in one of the latest conferences, Google has presented a smartphone able to take photos, automatically select any undesirable person (bystanders, motorcyclists exc.), delete them and then fill the empty pixels with what the background should have looked like if such person had not stood there in the first place.

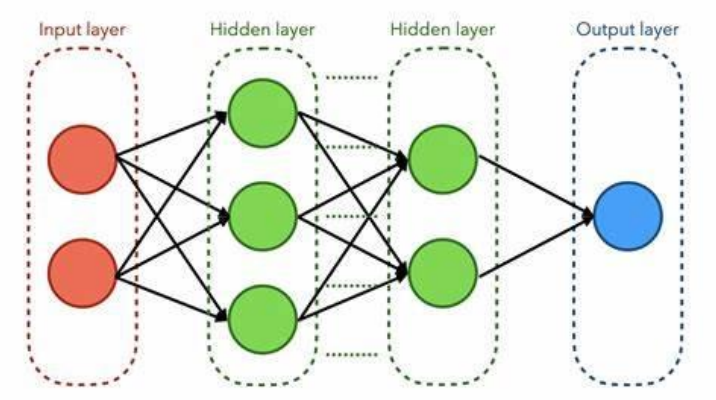


Figure 2.3. An example of an Artificial Neural Network.

2.2.5. Principal Component Analysis

Principal Component Analysis (PCA) is a technique used mainly for dimensionality reduction in ML tasks(Qu et al. 2002). The idea at its core is indeed to reduce the number of features within a dataset, while retaining as much variability as possible.

This is done by creating new variables, namely the Principal Components (PCs), which are built as linear combinations of the original ones with the attribute of being also perpendicular to each other in the PC's. Among them, the ones explaining the highest amount of variance are then considered as new basis for the dataset, while the remaining ones are discarded.

Even though in most situations PCA is used for the above-described purpose, in others it can be employed to simply scale and rotate data. This is the case of our thesis, where PCA was used to transform variables and link them together before the application of a Random Forest classifier.

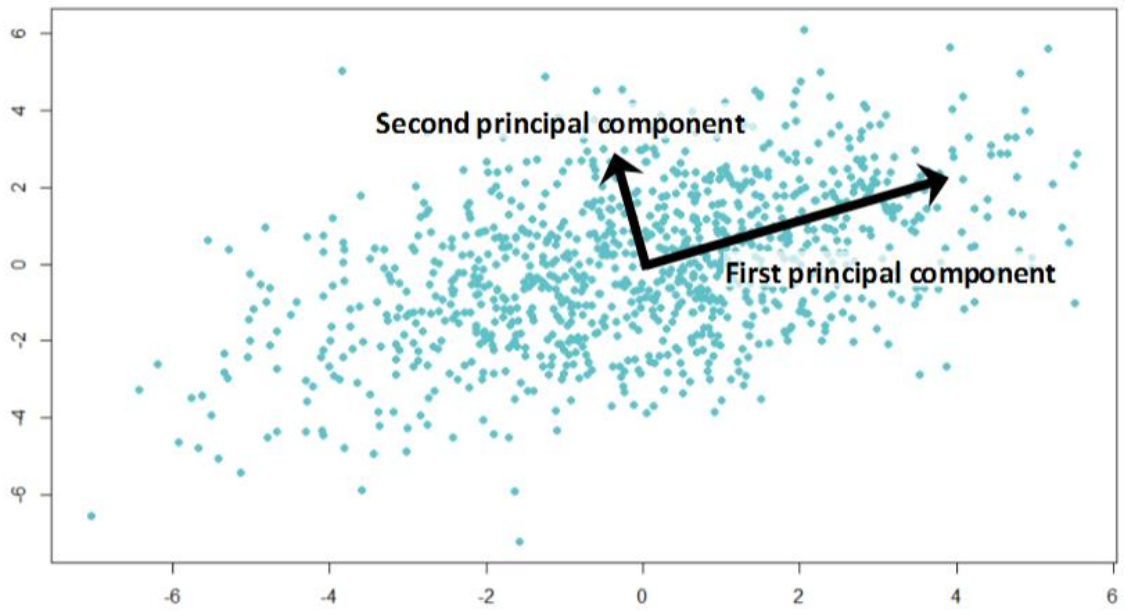


Figure 2.4. An example of the two Principal Components of a 2D dataset.

2.2.6. Google Colab

Google Colab (or Colaboratory) (https://colab.research.google.com/) is a cloud service born to execute Python code directly on browser. It does not need any kind of installation and gives as only requirement that of having a Google account.

This online platform is extremely convenient to anyone who does not have enough disk space, memory or raw power to execute complex algorithms, such as deep learning on large datasets.

This service comes free with its basic features, which are mostly enough for small to medium projects. Among them we can find: 1 core with 2 threads of an Intel Xeon CPU, 13 GB of RAM, 33 GB of available disk space, 12 hours of runtime and the possibility to use a Nvidia Tesla K80 GPU if available at the moment. However, the paywall, in addition to increased RAM, disk space and runtime duration, provides the user with higher priority on GPU runtimes.

As previously mentioned, in this thesis we used the Google Colab environment in its free version to train many ML Classification algorithms. We evaluated different parameters and models aiming to assess the role of WMH in the development of Clinical Dementia for an AD population.

3 Materials and Methods

In this chapter we present the full details on the experiments conducted during this thesis highlighting the reason behind the choices we made.

Firstly, we discuss the analysis conducted on BIANCA performance. We start with the data collection procedure and carry the explanation until final results are extracted and properly evaluated.

Afterwards, we present the Classification algorithm used to explain the relevance of WMHs when predicting scores of AD Clinical Dementia. Details on model selection, training strategies, and used libraries are carefully outlined. Finally, we discuss the evaluation metrics used to draw conclusions.

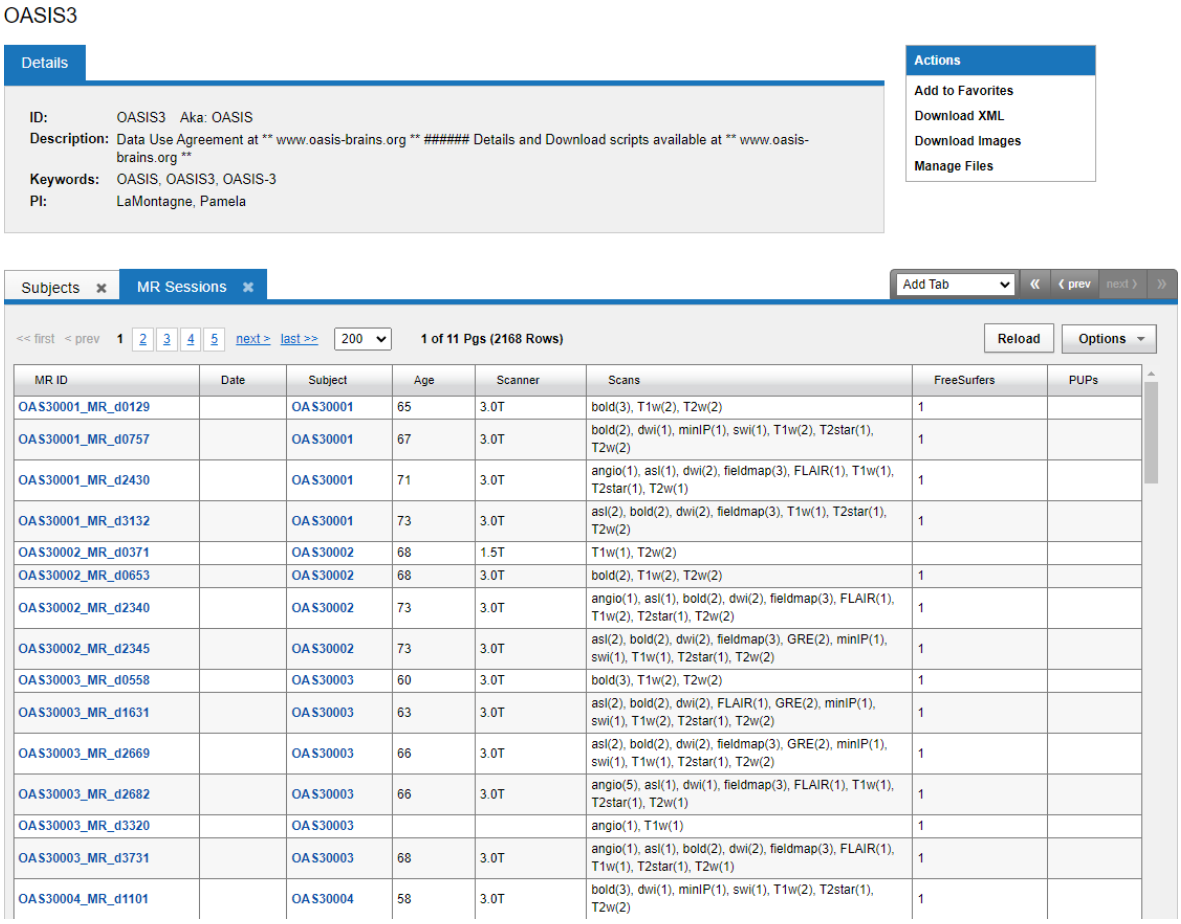
3.1. Evaluation of BIANCA performance

As previously introduced, the first part of the project is focused on the evaluation of BIANCA performance according to different training conditions: an increasing number of training subjects, a variable combination of MRI contrasts, and a pre-defined training set built using data from a population alternative to OASIS3.

The different steps undertaken will be fully explained. We start with the data collection procedure that was conducted downloading FLAIR, T1-weighted and SWI scans from the OASIS3 repository. Binary WMH masks were then outlined by a manual rater. Afterwards, the available MRI scans underwent a thorough pre-processing before being fed to BIANCA. The following steps were involved: brain extraction, BF correction, inter-modality registration and BIANCA masking. Eventually, the resulting images were processed by the segmentation tool to derive automatic WMH masks. The different steps required to run it were presented alongside the available BIANCA calls. As for performance evaluation, three separate analyses were conducted and fully detailed in independent paragraphs. To conclude the overlap indicators used to assess the quality of segmentations were described, together with the statistical tests used to draw comparisons.

3.1.1. Download procedure and Patients selection

After creating an account on the central.xnat.org website and logging in, it is possible to access the OASIS3 database from a dedicated section. Thereafter, since we are interested in MRI sessions, we have to add the corresponding Tab (i.e. "MR Sessions") with a simple scroll down menu called "Add Tab". Once this operation is done, we are presented with a page similar to the one displayed in Fig. 3.1.



MR ID	Date	Subject	Age	Scanner	Scans	FreeSurfers	PUPs
OAS30001_MR_d0129		OAS30001	65	3.0T	bold(3), T1w(2), T2w(2)	1	
OAS30001_MR_d0757		OAS30001	67	3.0T	bold(2), dwi(1), minIP(1), swi(1), T1w(2), T2star(1), T2w(2)	1	
OAS30001_MR_d2430		OAS30001	71	3.0T	angio(1), asl(1), dwi(2), fieldmap(3), FLAIR(1), T1w(1), T2star(1), T2w(1)	1	
OAS30001_MR_d3132		OAS30001	73	3.0T	asl(2), bold(2), dwi(2), fieldmap(3), T1w(1), T2star(1), T2w(2)	1	
OAS30002_MR_d0371		OAS30002	68	1.5T	T1w(1), T2w(2)		
OAS30002_MR_d0653		OAS30002	68	3.0T	bold(2), T1w(2), T2w(2)	1	
OAS30002_MR_d2340		OAS30002	73	3.0T	angio(1), asl(1), bold(2), dwi(2), fieldmap(3), FLAIR(1), T1w(2), T2star(1), T2w(2)	1	
OAS30002_MR_d2345		OAS30002	73	3.0T	asl(2), bold(2), dwi(2), fieldmap(3), GRE(2), minIP(1), swi(1), T1w(1), T2star(1), T2w(2)	1	
OAS30003_MR_d0558		OAS30003	60	3.0T	bold(3), T1w(2), T2w(2)	1	
OAS30003_MR_d1631		OAS30003	63	3.0T	asl(2), bold(2), dwi(2), FLAIR(1), GRE(2), minIP(1), swi(1), T1w(2), T2star(1), T2w(2)	1	
OAS30003_MR_d2669		OAS30003	66	3.0T	asl(2), bold(2), dwi(2), fieldmap(3), GRE(2), minIP(1), swi(1), T1w(1), T2star(1), T2w(2)	1	
OAS30003_MR_d2682		OAS30003	66	3.0T	angio(5), asl(1), dwi(1), fieldmap(3), FLAIR(1), T1w(1), T2star(1), T2w(1)	1	
OAS30003_MR_d3320		OAS30003			angio(1), T1w(1)	1	
OAS30003_MR_d3731		OAS30003	68	3.0T	angio(1), asl(1), bold(2), dwi(2), fieldmap(3), FLAIR(1), T1w(1), T2star(1), T2w(1)	1	
OAS30004_MR_d1101		OAS30004	58	3.0T	bold(3), dwi(1), minIP(1), swi(1), T1w(2), T2star(1), T2w(2)	1	

Figure 3.1. "MR Sessions" Tab opened withing the OASIS3 dedicated webpage.

The table-like structure of the records is immediately noticeable, with the main information being stored as column names. For the purposes of our thesis, we are mostly interested in the variable named "Scans", which illustrates the different MRI contrasts acquired during a certain session. Particularly, we need to find rows having a combination of the following scans: FLAIR, T1-weighted and SWI.

Unfortunately, the filtered search in OASIS3 does not support more than a keyword at a time. Therefore, to avoid downloading sessions without the required contrasts – thus wasting both time and disk space – it is necessary to use an external program that will be described afterwards.

Before using it, we need to download a spreadsheet of the whole "MR Sessions" tab, which is easily done by selecting the voice "spreadsheet" from the "Options" menu on the upper right corner of the screen. The download of a "cvs" file does begin and finish shortly after.

The spreadsheet is an exact replica of the "MR Sessions" tab. However, being in csv format it provides a major flexibility in our search for the required records, which is easily achievable using and the "Pandas" python library. A detailed example on how to carry out this task can be found in the Appendix, section 5.1 "Spreadsheet filtering".

The above-mentioned program can be easily accessed and downloaded via https://github.com/NrgXnat/oasis-scripts. The GitHub repository contains a list of scripts which enormously help with the download of images from OASIS3. The one used for this project is reported in Figure 3.2.

```
./download_oasis_scans.sh <input_file.csv> <directory_name> <xnat_central_username> <scan_type>
```

Figure 3.2. Script used to download images from the OASIS3 dataset. Required inputs are displayed.

The name of the script is "download_oasis_scans.sh" and it has to be launched from a Linux or MacOS terminal. The parameters inside "< >" are the ones it requires to operate. Here is a detailed description of each of them:

- <input_file.csv> - A Unix formatted, comma-separated file containing a column for experiment_id (e.g. OAS30001_MR_d0129)

- <directory_name> - A directory path (relative or absolute) to save the scan files to. If this directory doesn't exist when the script is run, it will be automatically created.

- <xnat_central_username> - Your XNAT Central username used for accessing OASIS data on central.xnat.org (you will be prompted for your password before downloading)

- <scan_type> - (Optional) The scan type of the scan you want to download (e.g. T1w, angio, bold, fieldmap, or FLAIR). You can also enter multiple scan types separated by a comma with no whitespace (e.g. T2w,swi,bold). Without this argument, all scans for the given experiment_id will be downloaded. In our case "FLAIR,T1w,swi" will be inserted .

Once the command is properly launched, all the available files are downloaded. Practically, we obtained 172 patients distributed over 206 sessions, each having a folder with the required imaging files. Alongside them, "json" files are also downloaded. They contain metadata for each modality, thus allowing to access information such as image dimensions expressed in voxels, slice thickness, acquisition time and many others. Our interest laid particularly on three parameters that helped us differentiating images: the "Manufacturer", the "ManufacturersModelName" and the "DeviceSerialNumber" which are, respectively, the factory where the MRI scanner was produced, its model's name, and serial number. That information allowed us to create three different groups of subjects according to the MRI scanners used for acquisition. Indeed, despite being all produced by Siemens, two different models were involved: the Biograph_mMR and the Triotim, both in their 3.0 Tesla variant. As for the latter, a couple of machines were used for the study, each having its own serial number. Therefore, the following three groups were created:

- Biograph_mMR, SN: 51010 → 3 Patients
- Triotim, SN: 35177 → 10 MR Patients
- Triotim, SN: 35248 → 159 MR Patients

Afterwards, we decided to keep a single MR session for patients who had multiple ones. That in order to maximize the heterogeneity of WMHs when training and testing BIANCA: indeed, lesions from the same subject, although scanned at different time points, could be very similar to each another. Specifically, since WMHs are a symptom of AD in this population we kept the most recent visit, due to the possibility for lesion to be broader than in previous ones. This choice brought us to discard 34 visits from 32 different participants, thus remaining with 172 images.

A further choice that we made was to use only the images acquired with the Siemens Triotim 35248. Indeed, MRI scanners have unique characteristics (such as the magnetic field inhomogeneities, the imaging gradient non-linearity, etc.), that differentiate them even when the same model is used. The requirement for individual calibration alongside the need for tailored harmonization techniques, led

us to decide that using only images acquired with the same equipment would have substantially favoured the automatic segmentation process. We therefore discarded 13 images and remained with a total amount of 159.

Eventually, we visually inspected the FLAIR images of participants and considered the following inclusion criteria: high lesional loads for WMHs, no strong artifacts of any kind, and a "regular" brain anatomy. Following this procedure, we selected a group of 40 patients to carry out the analysis. Information about their demographics is reported as follows:

- Age = 69.83 ± 6.67;

- Female:Male ratio = 23:17.

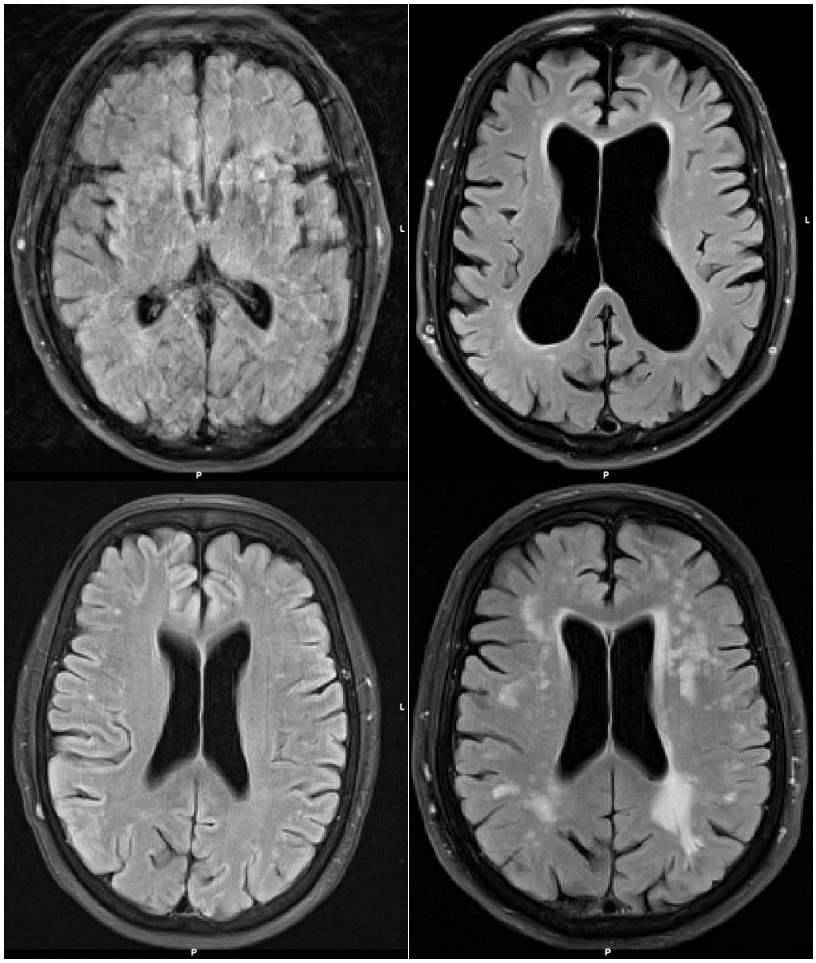


Figure 3.3. Example of four FLAIR images. The first three are all ineligible for WMH segmentation: the top left scan presents very strong motion artifacts; the top right scan shows extremely big ventricles which prevents the shape of the brain from being considered "regular"; the bottom left scan has visible WMH lesions which, however, are

very limited in numerosity and dimension. On the contrary, the last FLAIR image has heavy lesional load and is artifact-free, thus representing a valid example for the purposes of our work.

3.1.2. WMH masks creation

After data selection, in order to carry out the training phase for BIANCA, it was necessary to segment the WMH lesions. To achieve this goal, we used the Jim8 software (see Par. 2.1.2) which, once started, does prompt a window like the one displayed in Fig. 3.4

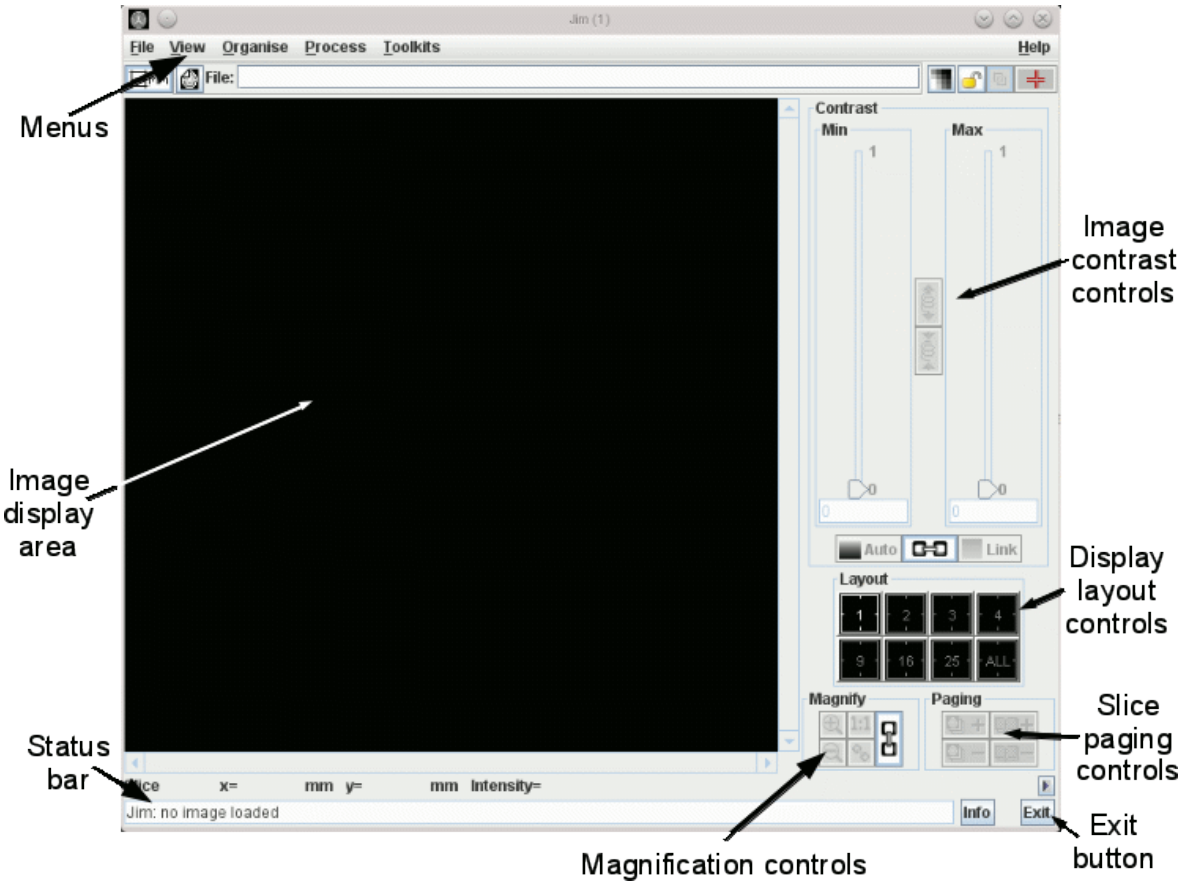


Figure 3.4. Jim8 startup page.

To load a medical image, it is necessary to click on the "File" button of the Menu, then select "Load" from the scroll down tab that does appear. At that point, it is either possible to select the required scan or, if its own folder is already open, to simply drag and drop it in the display area.

Once the image is loaded and fully visible in the central area of the program, the ROI analysis tool can be launched. That is easily done by clicking on the "Toolkits" button of the Menu and selecting the first option: "ROI Analysis". A detached window does eventually appear.

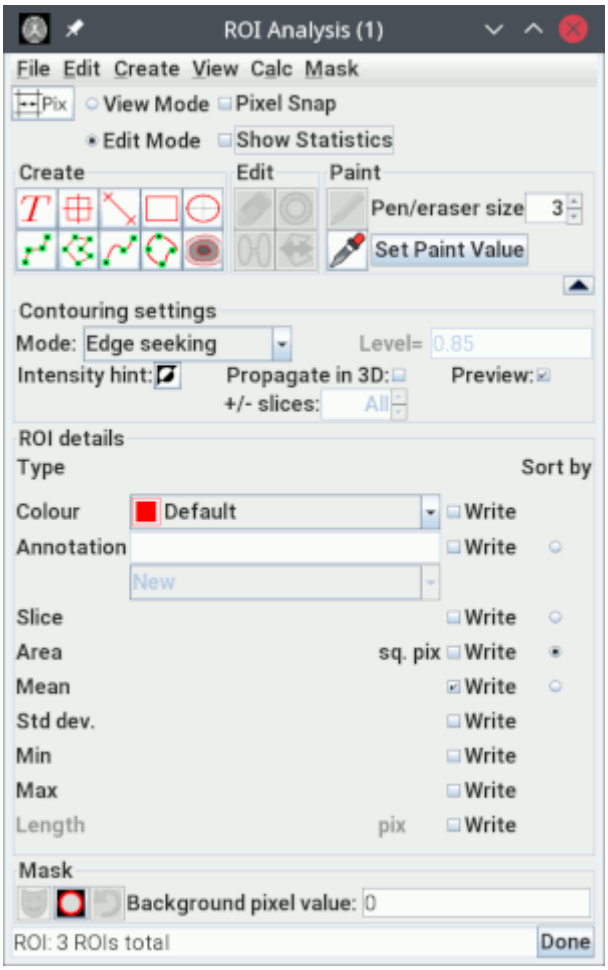


Figure 3.5. Jim8 ROI toolkit window.

In the "Create" section that is provided, we can choose among the different types of selection approaches available for our ROIs. In the case of our thesis, we decided for the irregular/contour following method.

Before going through the details of its usage, a quick mention about the options that needs to be set is required. First, we have the "Intensity hint" feature, which has already been mentioned in Par. 2.1.2 "Jim8". This is an aid we can give the software to automatically look for specific patterns within images. In case of WMHs, the "Intensity hint" option should always be set to "features brighter than the background" in order to facilitate the identification process (except for the case of

hollow lesions). Secondly, we have the "propagate in 3D" button, able to extend the selected ROI from current slice to the adjacent ones. This can be of help for large and extended lesions while, on the other hand, may not work properly for small WMHs, that could be affected by noise and uncertainties such as BF or the nearby biological structures. In addition, we also have the eraser tool from the "Edit" section, having an icon similar to a rubber. This tool, when active, allows the user to exclude from the ROI a certain amount of pixel manually highlighted using the cursor. This can help separating ROIs that have been erroneously joined by the contouring method, or eliminating morphological structures mistakenly included in the WMH lesion area.

Coming back to the Contour Following button, having it selected and moved the mouse over the image, Jim produces a contour, shown in light blue, outlining an image feature. Below is the result of applying edge detection and contouring to the edge of brain lesion.

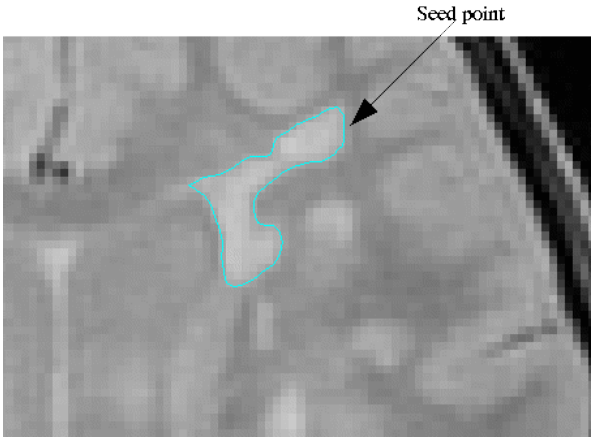


Figure 3.6. Higher intensity area automatically segmented with a light blue contour by the Jim8 "ROI Analysis" Toolkit.

The procedure of contour selection and manual refinement must be applied, slice by slice, to all the WMH lesions present within the image. Eventually, when a satisfying result has been reached, the ROI can be saved as a unique 3D file through the option "Save ROIs" displayed at the "File" section of the Toolkit.

However, while BIANCA needs binary WMH masks to carry out its training process, the saved contours are not yet binary. This issue is easily solved by using the "Masker" tool, prompted by the "Process" tab of the main Jim8 window.

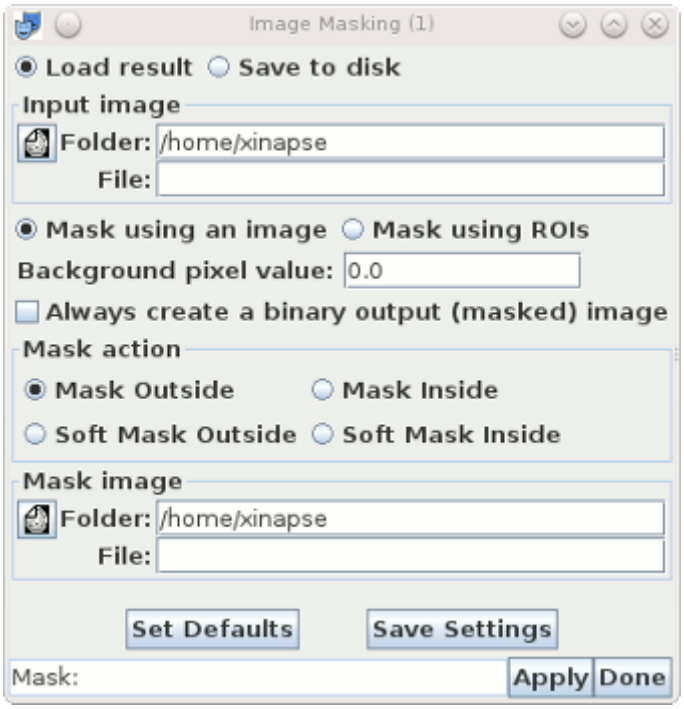


Figure 3.7. The Jim8 Masker tool window.

This tool is extremely easy to use. First, since we need to save the resulting binary output, the button "Save to disk" must be selected. Afterwards, both the original FLAIR and the previously derived ROI need to be loaded from the "Input image" and "Mask image" menus respectively. This can be done after selecting the "Mask using ROI" option from the dedicated section.

Finally, after selecting "Always create a binary output (masked) image" and "Mask outside", the binary ROI can be obtained by following the instructions appearing after we click on the "Apply" button.

The procedure of creating ROIs and binarizing them was repeated for all the 40 subjects selected as dataset to train and test BIANCA.

Eventually, in addition to the first 40 WMH segmentations, other 40 masks were drawn to evaluate the intra-rater variability. From now onwards, we will refer to the different available rounds as "Preliminary segmentation" and "Expert segmentation", respectively. They were both performed by the same rater (S.C.), but present some key difference. The formers were outlined in a time span of 4 weeks at the very beginning of this project and are thereby associated to little experience on both WMH morphology and Jim8 usage. On the contrary, the "Expert

segmentation" were performed in a time span of 5 days 4 months after the first ones. Thus, they reflected higher experience and improved skills.

3.1.3. Imaging data organization

Before moving to the image pre-processing step, it is important to explain the renaming and the moving procedure that needs to be carried out on the imaging files. After creating the WMH masks, each patient's folder is indeed organized as follows:

- OAS30027_MR_d2394
  - anat2
    - sub-OAS30027_ses-d2394_FLAIR.json
    - sub-OAS30027_ses-d2394_FLAIR.nii.gz
    - sub-OAS30027_ses-d2394_FLAIR_roi1.roi
    - sub-OAS30027_ses-d2394_FLAIR_roi1.nii.gz
    - sub-OAS30027_ses-d2394_FLAIR_roi2.roi
    - sub-OAS30027_ses-d2394_FLAIR_roi2.nii.gz
  - anat3
    - sub-OAS30027_ses-d2394_T1w.json
    - sub-OAS30027_ses-d2394_T1w.nii.gz
  - swi4
    - sub-OAS30027_ses-d2394_swi.json
    - sub-OAS30027_ses-d2394_swi.nii.gz

However, this causes problems with the data management procedures: firstly, the file names change drastically from one session to the other, thus making difficult any listing operation; secondly, the different anatomical folders do not always contain the same contrast: *anat3* could contain FLAIR images in one case and T1 in the other, further complicating the execution of any automatic procedures on data.

This issue can easily be solved by means of a thorough renaming strategy which is usually carried out with few lines of bash code directly inputted in the terminal. An example can be found in the Appendix, section 5.2 "Renaming external". After launching the commands, the directories outline should look like this:

- sub03
  - FLAIR
    - sub03_FLAIR.nii.gz

        - sub03_FLAIR_roi1.nii.gz
        - sub03_FLAIR_roi2.nii.gz
    - T1w
        - sub03_T1w.nii.gz
    - SWI
        - sub03_swi.nii.gz

3.1.4. Images preprocessing

Before performing any automatic segmentation, the imaging data need to be pre-processed. In this paragraph we describe in detail all the required steps, starting with brain extraction and BF correction. We then move on to the image registration and FLAIR masking procedures.

3.1.4.1. Brain extraction

The brain extraction task is achieved using the previously mentioned FSL's tool named BET. The corresponding command is launched via Linux or MacOS terminal and takes as parameters the input and the output file paths, along with additional processing options, if specified by the user. Here is an example of its call:

bet <input> <output> <options>

In our case, we only used "-R" which stands for "Robust". With this option BET runs a recursive brain centre estimation that works as follows: it repeatedly calls the function, using the same input image and main options, but updates the value of -c (which represents the starting centre of the brain estimation) with the centre-of-gravity derived during the previous brain extraction. The primary purpose of this process is to improve results when the input image contains a lot of non-brain matter – especially for those case where neck tissue in involved. By iterating this way, the centre-of-gravity is moved each time towards the "true centre", and the algorithm provides a better final estimate. Iterations stop when the centre-of-gravity is no longer updated or when a maximum of 10 cycles is reached.

The resulting output is displayed in Fig. 3.8.

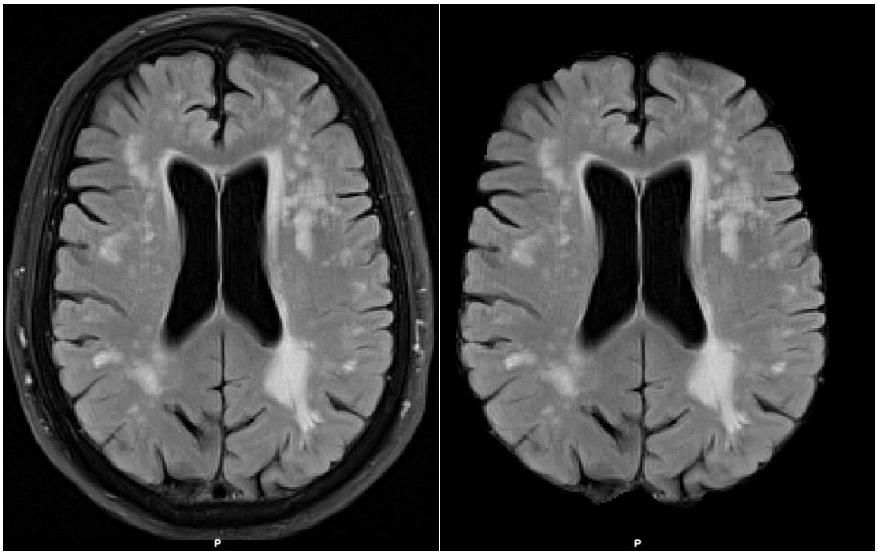


Figure 3.8. Original (left panel) and brain-extracted (right panel) FLAIR image.

Fig. 3.8 displays an example relative to a FLAIR image. However, the BET procedure was carried out on all the MRI scans before being fed to BIANCA for the automatic WMH segmentation. In the Appendix, section 5.2 "BET run" an example on how to automatically execute this function on all 40 subjects is also included.

3.1.4.2. Biasfield correction

The BF correction task is achieved with another FSL tool named FAST. The command is again launched via Linux or MacOS terminal and it only requires as parameters the input image along with additional processing options, if needed. The required call is the following:

fast <input> <options>

In our case we specified "-B", which instructs the algorithm to calculate the existing BF, to remove it from the original image, and to automatically save the resulting output by adding the suffix "_restore" to the original file name. Additionally, there is an important requirement in order to properly run the tool: the input image needs be brain-extracted beforehand. For this reason, BET and FAST must be executed in series with no possibility to modify the processing order.

The resulting outputs are presented in Fig. 3.9 for a FLAIR image, while the full utilized code is reported in the Appendix, section 5.3: "FAST run". Also in this

case, it's important to point out that the BF correction procedure was carried out on all the available MRI scans.

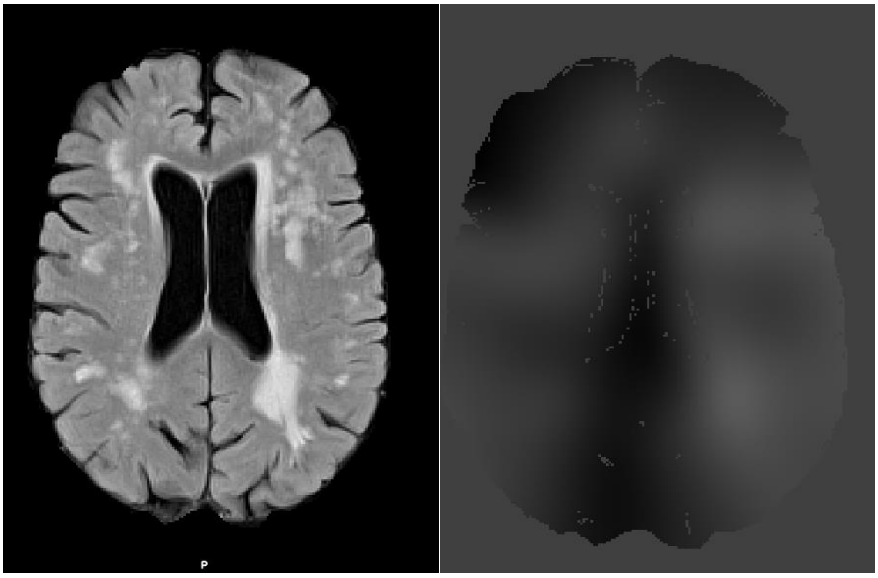


Figure 3.9. Original FLAIR image (left panel) later BF corrected using FAST (central panel). The outline of the magnetic field inhomogeneities is also displayed (right panel).

3.1.4.3. Image registration

As previously mentioned, selecting a reference image is necessary before feeding the MRI scans to BIANCA and, for this purpose, the existing literature identifies FLAIR as the most frequent choice. This is due to the following reasons. Firstly, FLAIR has a lower resolution with respect to other imaging modalities. This may appear as a disadvantage. However, registering a higher resolution image to a lower resolution one is simple and scarcely prone to errors, while the opposite solution requires upscaling procedures that may add flaws to the resulting image. Secondly, ROIs are usually outlined on FLAIR scans since WMHs are more easily recognizable on this contrast with respect to any other. Registering the FLAIR would therefore introduce changes to the manual WMH labels that may eventually impair the segmentation performance of BIANCA. For these reasons, FLAIR was chosen as reference image for our project, while the remaining scans SWI scans were all reported to its 3D coordinate space.

To carry out the registration process we used the FSL linear (i.e., affine) registration tool named FLIRT, which is executed through the following line command:

flirt -in invol -ref refvol -out outvol -omat invol2refvol.mat -dof 6

where *invol*, *refvol*, *outvol* are the input, reference and output volume filenames respectively, and *invol2refvol.mat* is the filename for the saved ascii transformation matrix. The option "-dof" indicates with how many degrees of freedom FLIRT can operate. The default value is 12 but, since we are performing an intra-subject registration (same subject, different MRI modalities), the 6 degrees of a rigid registration still allow to obtain optimal performance while maintaining a low computational load.

FLIRT works by taking a cost function that quantifies the quality of the registration and then finding the transformation which gives the minimum cost.

In Fig. 3.10 we present the obtained results for the registration process between a T1 and a FLAIR scan. The resolution reduction suffered by the latter as a consequence of the above-mentioned concepts can be visually appreciated.

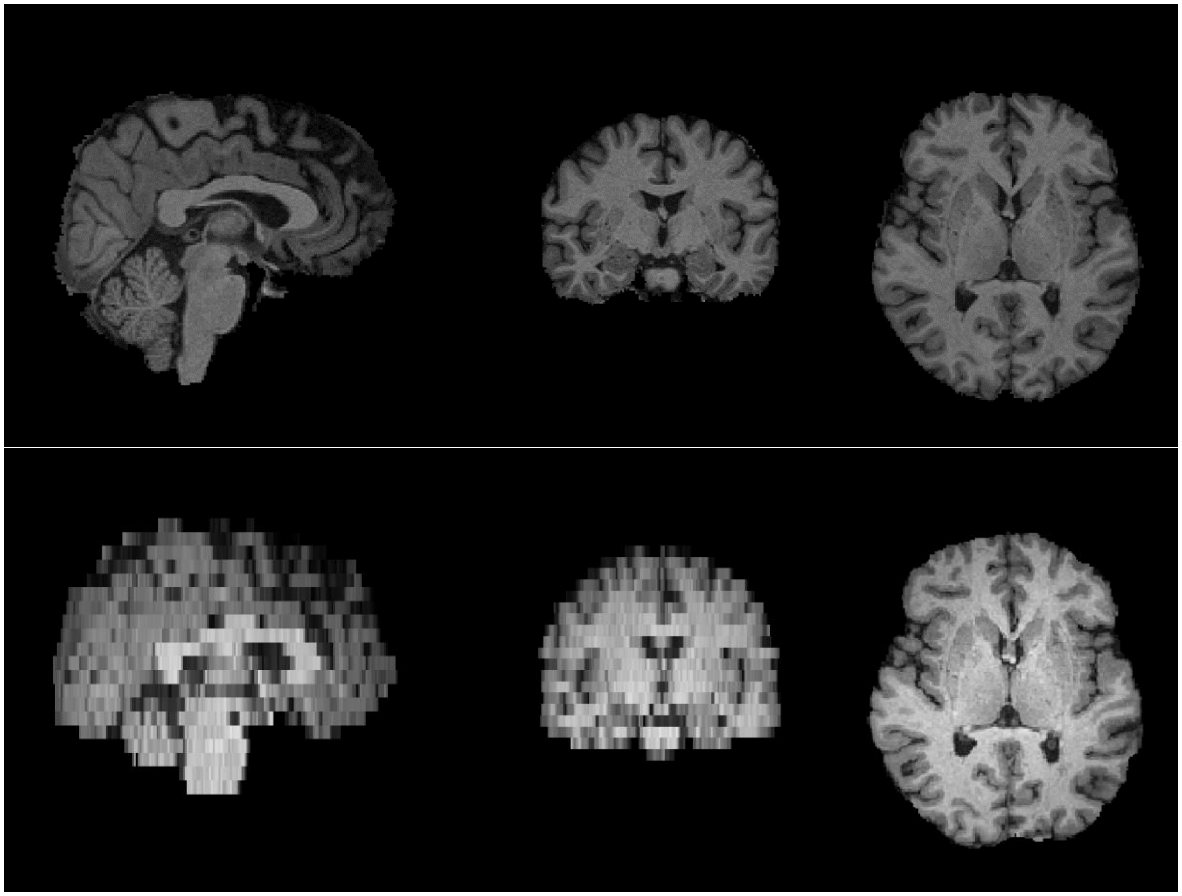


Figure 3.10. Noticeable spatial resolution loss suffered from a T1-weighted image (upper panel) after being registered to the corresponding FLAIR scan (lower panel).

For the purposes of this thesis, this step was carried out on both T1-weighed and SWI images. The full code to iteratively run FLIRT on our dataset can be found in the Appendix, section 5.5: "FLIRT run".

3.1.4.4. FLAIR masking

As already introduced, to reduce the false positive rate affecting the automatic segmentation of BIANCA, a masking procedure on one of the involved MRI modalities is usually recommended. This is done by retaining the White Matter (WM) voxels, while setting to 0 the ones relative to cortical grey matter and to structures like putamen, globus pallidus, nucleus accumbens, thalamus, brainstem, cerebellum, hippocampus, amygdala. These areas are indeed the most prone to artifacts or to the presence of hyperintense spots, thus possibly impairing the resulting performance.

To mask them out, four FSL tools will be used:

- the "fsl_anat" pipeline, used for processing the T1 anatomical image.
- the "make_bianca_mask" script, which creates the binary exclusion mask directly on the T1 space.
- FLIRT, to align the mask with respect to selected MRI modality, when the latter is different from T1 (like in our case).
- fslmath, to eventually apply the binary mask to the MRI modality itself.

As to fsl_anat, the pipeline outputs many different files for several different purposes. However, our interest only lies in three of them: the brain-extracted and bias field corrected T1 image, the Cerebrospinal Fluid (CSF) partial volume effect map, and the non-linear transformation warp file from standard MNI space to structural image space. Note that, the CSF map is the result of a tissue-type segmentation that aims to distinguish three binary classes: CSF, GM and WM. However, since "make_bianca_mask" only needs the first one, we will focus on that.

An example call for the tool is given by the following:

fsl_anat -i <input>

The only required parameter is the input imaging file representing the original T1-weighted scan (i.e. without brain extraction or BF correction applied on it).

At this point, after obtaining the images required for the second step, the script "make_bianca_mask" creates the binary mask of exclusion for the above-mentioned structures. The cortical GM is excluded from the brain mask by extracting the cortical CSF from single-subject's CSF map, dilating it to reach the cortical GM, and excluding these areas. The other structures are identified in the MNI space, non-linearly registered to the single-subjects' images, and eventually removed from the brain mask.

A command line of example is given by the following:

*make_bianca_mask < *_biascorr> <*_fast_pve_0> <*_nonlin_field.nii.gz>*

Where inputs are exactly the three files mentioned during the fsl_anat description. The output is instead a file of the exclusion mask which is automatically saved by the call.

Now, if the MRI modality selected for masking is different from T1, the binary mask needs to be registered from its native space. This step can be done using FLIRT.

In our case, since every scan was already registered to the FLAIR space, we decided to align to it also the exclusion mask. This was done by exploiting the previously derived T1_to_FLAIR transformation matrix. An example call is given by the following expression:

flirt -in newvol -ref refvol -out outvol -init invol2refvol.mat -applyxfm

Note that the transformation matrix (*invol2refvol*) is loaded using the *–init* command without need for further estimation and the size of the output volume is determined by *refvol* (which is the only information used taken from this image).

Finally, to apply the binary mask directly onto the selected MRI modality the following command needs to be run:

fslmaths input_image_2_FLAIR -
mas T1_bianca_mask_to_FLAIR input_image_2_FLAIR_maskedThe -mas parameter tells the algorithm that this is a masking process, thus masking the input file FLAIR with T1_bianca_mask_to_FLAIR and outputting FLAIR_masked.

The -mas parameter tells the algorithm we are conducting a masking process over the input file *input_image_2_FLAIR* , *using T1_bianca_mask_to_FLAIR* to finally output the *input_image_2_FLAIR_masked* file .

The whole masking procedure was usually conducted on FLAIR scans. However, when single modality approaches were compared to multimodality ones, this very last step was also performed using SWI (obviously registered to the FLAIR space) as the input image.

An example of the obtained results is reported in Fig. 3.11 for a FLAIR scan, while a full list of the codes can be found in the Appendix, section 5.6: “BIANCA masking”.

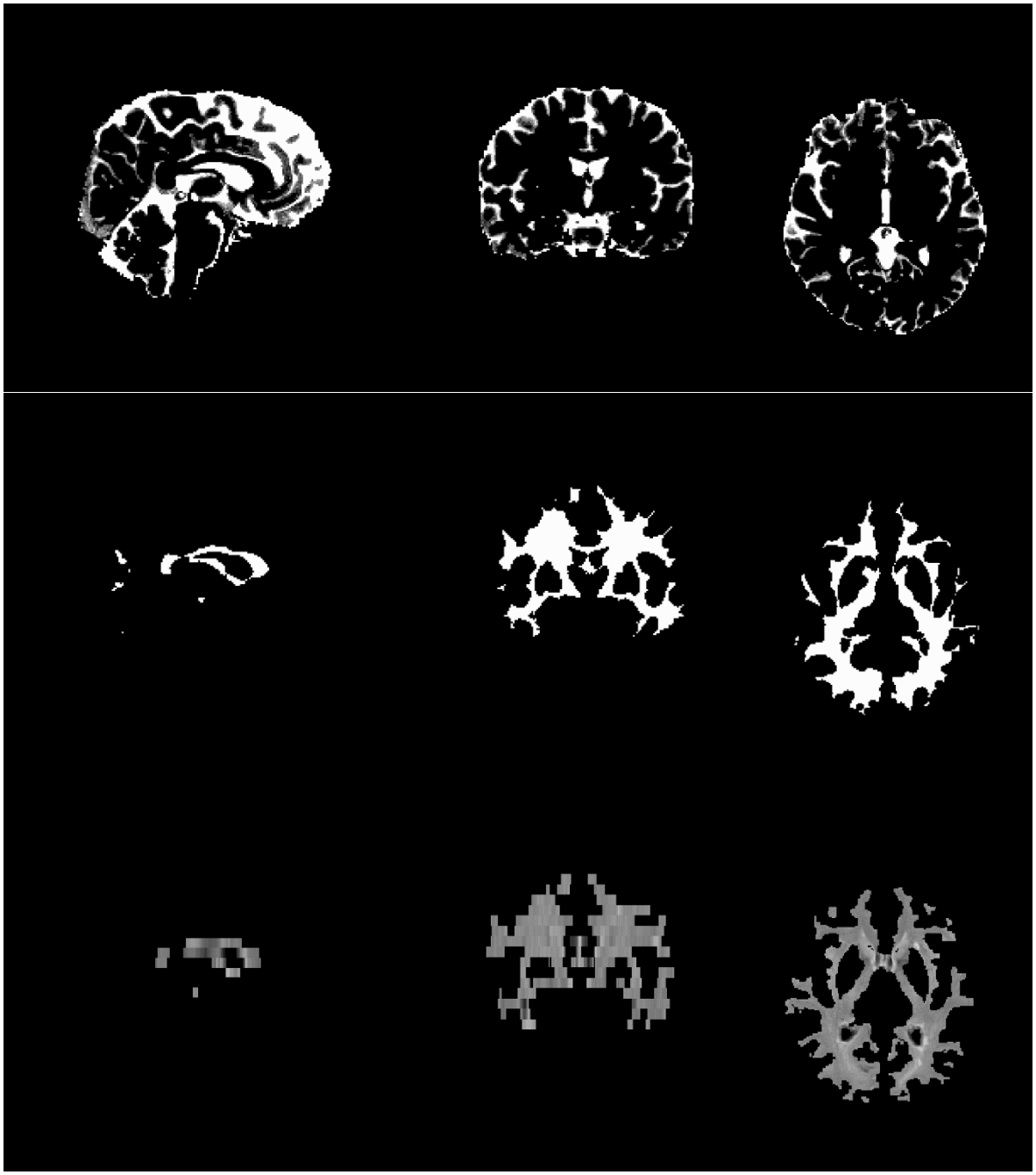


Figure 3.11. Binary image of the CSF mask (top panel); binary mask created using the "make_bianca_mask" command (central panel); bianca mask registered and applied to the FLAIR image (bottom panel).

3.1.5. Running BIANCA

This present paragraph is divided in six separate parts: the first introduces the masterfile required to load images within BIANCA; the second presents the three different modalities which can be used to run the tool and goes through the detail of its parameters; finally, the following three section focus on how BIANCA is used for our purposes, thus covering the incremental training analysis, the multimodality analysis and the validation step carried out on an existing harmonization pipeline; the final part describes how probability maps derived from BIANCA are thresholded and how segmentation performance are eventually quantified by means of overlap metrics.

3.1.5.1. Masterfile creation

After the pre-processing steps, MRI scans are ready to be fed to BIANCA.

However, in order to work properly the tool requires the creation of a Masterfile. That is represented by a text file containing a row for every subject involved and, for each row, a list of all the files needed for that subject. Files are written following a specific order, which needs to be maintained throughout the whole document.

In order for BIANCA to execute both training and testing procedures, the masterfile has to provide a location for the following files:

- The base image;
- The lesion mask outlined on the base image (for training subjects only; for query subjects this path can be used along with any other "placehold" name to keep the same column order);
- Additional images registered to the base one for multimodality sessions;
- The transformation matrix from base image to standard MNI space, used to extract the spatial coordinates of every voxel.

As for the latter, the FLAIR_to_MNI matrix is the only file not directly obtained from the former pre-processing steps. However, we are in possession of both a matrix from T1-weighted to FLAIR space (which is the result of the registration step performed using FLIRT) and a matrix from T1 to MNI space (which is among the files output by “fsl_anat”). Thanks to a FLIRT script called “convert_xfm”, the former can be inverted to obtain the FLAIR to T1 transformation. This latter can

eventually be concatenated with the T1_to_MNI file in order to provide the required transformation matrix.

Inverting a matrix using "convert_xfm" is fairly simple and can be done using the following:

convert_xfm -omat refvol2invol.mat -inverse invol2refvol.mat

The *-omat* parameter indicates the output matrix while the *-inverse* one points at the matrix that needs to be inverted.

On the other hand, to concatenate a couple of matrices we need to use a modified version of the former call:

convert_xfm -omat AtoC.mat -concat BtoC.mat AtoB.mat

where "*-concat*" points at the required mat files.

The following image presents an example of how the general outline of a Masterfile should look like.

```
Example master file (masterfile.txt):
subj01/FLAIR_brain.nii.gz subj01/T1_to_FLAIR.nii.gz subj01/FLAIR_to_MNI.mat subj01/WMHmask.nii.gz
subj02/FLAIR_brain.nii.gz subj02/T1_to_FLAIR.nii.gz subj02/FLAIR_to_MNI.mat subj02/WMHmask.nii.gz
...
subj<N>/FLAIR_brain.nii.gz subj<N>/T1_to_FLAIR.nii.gz subj<N>/FLAIR_to_MNI.mat subj<N>/WMHmask.nii.gz
```

Figure 3.12. Example of a possible Masterfile, directly taken from the user guide of BIANCA (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BIANCA/Userguide).

However, for the purposes of our thesis, the different files included in it were the following: path/sub<N>_FLAIR_masked, path/sub<N>_T1w_reg, path/sub<N>_swi_reg, path/sub<N>_roi1, path/sub<N>_roi2, path/sub<N>_FLAIR_to_MNI, path/sub<N>_swi_masked; where <N> reanges from 01 to 40.

They will be used in different combination (always considering subgroups rather than the entire set of files) to perform the different steps of our analysis.

An example on how to easily create a Masterfile for our data can be found in the Appendix of this thesis, section 5.7: "Masterfile creation".

3.1.5.2. BIANCA call

As said before, BIANCA is a ML algorithm, this implies a time-consuming process for learning, but fortunately there is the possibility to train on a particular set of data, then save the parameters calculated during the training phase as a file and reloading them when BIANCA is applied on new data, based on the previous training. With this method it is possible to use the classifier trained with different datasets without owning or having access the datasets themselves, this possibility will also be used in this thesis to test BIANCA trained with images from UK Biobank and Whitehall datasets and then tested on images from OASIS3.

There are three different modalities BIANCA can be called: the first one is "BIANCA Training", this is meant to be used to train and save the classifier and its parameters, to be used in subsequent calls; the second is "BIANCA Testing", in this modality a previously saved classifier is loaded and used on images to be segmented; finally "Leave One Out Validation", in which all the subjects' images indicated in the masterfile are used to train the algorithm, except for the ones of the test subject, which are automatically excluded by the training pool.

As mentioned in the previous chapter, BIANCA has no GUI, so, in order to run it, a Linux or MacOS terminal command must be created with all the information the program needs.

There are four main arguments which are compulsory in a BIANCA's call:

- `--singlefile=<masterfile>` location and name of the master file (e.g. path/masterfile.txt)
- `--querysubjectnum=<num>` row number in the master file of the query subject (the one to be segmented)
- `--brainmaskfeaturenum=<num>` column number in the master file containing the name of the image to derive the lesion probability map over, in our case will be always the Base Image and, except for one single case, FLAIR will be the one selected for this argument.
- Training dataset specification:
    - If the training phase needs to be performed in order to use BIANCA (thus in "BIANCA Training" or "Leave One Out Validation" modality) , the following arguments need to be specified:
        - `--labelfeaturenum=<num>` column number in the master file containing the name of the manual lesion mask, since in our case we have 2 ROIs, only one at a time can be called.

        - `--trainingnums=<val>` subjects to be used in training. List of row numbers (comma separated, no spaces) or 'all' as input to use all the subjects in the master file. If the query subject is also a training subject, it is automatically excluded from the training dataset and the lesions are estimated from the remaining training subjects.
    - Alternatively load from file (previously saved with `-saveclassifierdata`, used in "BIANCA Training" modality):
        - `--loadclassifierdata=<name>` load training data (and labels) from a file.

The three different modalities recognized through the presence of the following parameters:

- BIANCA Training: `--saveclassifierdata`
- BIANCA Testing: `--loadclassifierdata`
- Leave One Out Validation: `--trainingnums` set to 'all', no `--saveclassifierdata` or `--loadclassifierdata` used

Additionally, different optional parameters can be used in the call, such as which contrast images to use during the Training and the Testing phase, specify in which column of the masterfile the transformation matrix is located. In addition to those, advanced parameters for a finer tuning can be specified. The reader can find all the specified option we used in our thesis in the Appendix section 5.8 "BIANCA Training" and 5.9 "BIANCA Testing".

For a better comprehension of the codes written in the Appendix, it must be noted that, differently from some program languages, for a correct BIANCA's call, all row and column numbers start counting from 1 (not zero).

3.1.5.3. Incremental training analysis

At first, we evaluated the performance of BIANCA obtained with an incremental number of training subjects. The aim was to assess the existence for potential minimum/maximum numerosity values that could be taken into account when trying to optimize the segmentation outcome. The MRI modalities initially involved

in this step were FLAIR and T1, used together. However, we then extended the investigation to all the possible combinations listed in the multimodality paragraph (Par. 3.1.5.4. "Multimodality analysis"). For every combination of modality, evaluated independently from the others, the analysis was conducted using separate procedures for training and testing. As for the former, the following amounts of subjects were investigated:

- 10 subjects;
- 15 subjects;
- 20 subjects;
- 25 subjects;
- 30 subjects.

Since the higher training group is 30 subjects, in order to maximize the number of results (which would be equal to the number of the total subjects, 40), we decided to create 4 different Testing groups, which are described below:

- Group A:
  - Testing: subject 01 to 10
  - Training pool: subject 11 to 40
- Group B:
  - Testing subject: 11 to 20
  - Training pool: subject 01 to 10 and 21 to 40
- Group C:
  - Testing subject: 21 to 30
  - Training pool: subject 01 to 20 and 31 to 40
- Group D:
  - Testing: subject 31 to 40
  - Training pool: subject 01 to 30

This choice of group selection was made to prevent the possibility of BIANCA to be tested on a subject present also in the training pool. If this group division had not been done, we would have had only 10 results in the 30 subject BIANCA training, this ensures 40 results for each of the incremental training call.

Since the whole process was followed for both the Preliminary and the Expert Segmentations, results were evaluated with the overlap metrics described in Par. 3.1.5.6 using the correspondent manual masks used for each Training phases.

3.1.5.4. Intra-rater variability

Before carrying on with the analysis, the amount of variability affecting the manual rating procedure was evaluated. To do that, we calculated the Dice Similarity Index (see par 3.1.5.7 for further details) between the "preliminary" and the "expert" manual masks, aiming to evaluate the amount of overlap between their areas. Final results were displayed using a boxplot.

Afterwards, a visual inspection process was also carried out to investigate the nature of potential differences.

3.1.5.5. Multimodality analysis

Secondly, we evaluated how the different combination of MRI modalities impacted on the segmentation performance. To do that we fixed the number of training subjects to 40, including the entire dataset at our disposition. Then we assessed the following MRI combinations:

- FLAIR only
- FLAIR + T1-weighted
- FLAIR + SWI
- FLAIR + T1-weighted + SWI
- SWI only

The obtained results were eventually compared against each other. The analysis was conducted training and testing BIANCA in leave-one-out validation while performance were derived using the overlap metrics (Par. 3.1.5.7) after comparison with the Expert Segmentation masks.

3.1.5.6. Harmonization pipeline

Finally, we assessed the performance reached by BIANCA when trained on a well-known population and tested on a different one (mixed training approach). In particular, we used the training set from (Bordin et al. 2021) including data from both the Whitehall and UK Biobank dataset. The aim was to validate its ability to provide good segmentation outcomes even when applied to an entirely new dataset with respect to the ones involved in training. For this reason, after training, we tested BIANCA on the pool of all the 40 OASIS3 subject selected for our study.

The MRI modalities involved in the analysis were only FLAIR and T1 (used together), to respect the imaging composition of the pre-assembled dataset. The obtained overlap indices, calculated on the Expert manual masks were then compared with that of existing literature.

Eventually, results from the mixed training approach were compared with those obtained from the incremental training sets described in Par. 3.1.5.3 (evaluation on "expert" manual masks using a 4-fold cross validation). These latter cases represented indeed a single-site training strategy (training and testing performed on the same population) considered as optimal with respect to the mixed. Therefore, we wanted to assess for which the number of training subjects the optimal approach would become comparable to its alternative.

3.1.5.7. BIANCA's evaluation

Finally, to evaluate the WMH segmentation results, we used performance indicators of the overlap between the WMH lesion probability map – binarized with a global threshold of 0.9 – and the manually outlined WMH mask.

To derive those values, we used a dedicated and very easy-to-use FSL script: "bianca_overlap_measures". Its call included few simple parameters:

bianca_overlap_measures <lesionmask> <threshold> <manualmask> <saveoutput>

The first is the lesion probability map derived from BIANCA; the second is the threshold applied to it before calculating the overlap measures (if you have already thresholded the lesion probability map you can input 0 to the second parameter); the third is the manual mask, used as reference for the evaluation. If *<saveoutput>* is set to 0 the script will output both names and values for the different indicators directly on the terminal. Otherwise, if set to 1, output values will be saved in a separate text file.

The different overlap measures extracted by the tool are the following:

- Dice Similarity Index (SI): calculated as 2*(voxels in the intersection of manual and BIANCA masks)/(manual mask lesion voxels + BIANCA lesion voxels)
- Voxel-level false discovery rate (FDR): number of voxels incorrectly labelled as lesion (false positives, FP) divided by the total number of voxels labelled as lesion by BIANCA (positive voxels)

- Voxel-level false negative ratio (FNR): number of voxels incorrectly labelled as non-lesion (false negatives, FN) divided by the total number of voxels labelled as lesion in the manual mask (true voxels)
- Cluster-level FDR: number of clusters incorrectly labelled as lesion (FP) divided by the total number of clusters found by BIANCA (positive clusters)
- Cluster-level FNR: number of clusters incorrectly labelled as non-lesion (FN) divided by the total number of lesions in the manual mask (true clusters)
- Mean Total Area (MTA): average number of voxels in the manual mask and BIANCA output (true voxels + positive voxels)/2
- Detection error rate (DER): sum of voxels belonging to FP or FN clusters, divided by MTA
- Outline error rate (OER): sum of voxels belonging to true positive clusters (WMH clusters detected by both manual and BIANCA segmentation), excluding the overlapping voxels, divided by MTA
- Volume of BIANCA segmentation (after applying the specified threshold)
- Volume of manual mask

For the purposes of this thesis, we will focus mostly on DICE that allowed us to have an immediate grasp on the level of accuracy provided by the different options investigated throughout the analysis. The obtained results were compared trying to find the set of training parameters (number of training subjects and involved MRI modalities) able to optimize performance. They were also used to validate a preliminary harmonization strategy.

As always, the full list of codes utilized to calculate overlap indicators and to store their values in a single file can be found in the Appendix paragraph, section 5.10: "BIANCA performance".

3.2. The Classification algorithm

In this section we present the strategies used to build a classification model that aims to predict the Clinical Dementia Rate (CDR). This score indicates the degree of cognitive impairment affecting participants and was used as *target* variable for our analysis. *Input* variables were instead represented by the clinical information

extracted from OASIS3 and by the measures of WMH volumes derived for each subject. However, it's important to point out that our main goal was not directly the CDR prediction, but instead a thorough evaluation of the impact exerted by WMHs on the final performance.

To clearly explain the details of our analysis, this section is divided in two different parts: “Dataset preparation” and “Algorithm creation”

The first describes carefully the steps undertaken to compose the addressed database, starting from data selection/download up to the procedures that allow to extract variables. The second, on the other hand, presents the strategy used to train the algorithms and to derive results. We go through the required pre-processing and model building procedures and, eventually, describe the evaluation metrics used to assess the algorithm performance.

3.2.1. Dataset preparation

In order to build our dataset different steps were required. Firstly, we had to selects the imaging records to be addressed by the analysis. Then, for each record, we had to download MRI data. The opposite procedure was applied to the clinical records: they were downloaded first and selected afterwards. Secondly, the MRI data had to be processed and fed to BIANCA to derive the volumetric amounts of WMHs present on each subject’s brain. Eventually, the clinical data had to be filtered and properly organized to be suitable for the step of model creation (that will be discussed in Par. 3.1.2). That is the reason for this paragraph to be split in three sections.

3.2.1.1. Data selection and download

As to the imaging data, we recall that classification algorithms work better with large datasets and that the most utilized scans for the purpose of WMH segmentation are FLAIR and T1. Therefore, we selected from the OASIS3 repository only the MR Sessions having both those contrasts. To accomplish this task, we used our former knowledge on the dataset: in every MR Session, when a FLAIR was acquired, a T1-weighted scan was also performed. Therefore, we only had to parse the repository looking for the MR Session containing FLAIR images.

Applying a filtered search on the database was fairly simple. We opened the “MR Session” tab. Then, by clicking on the header of the column “Scans”, an interactive menu with five options was prompted. From there, we respectively selected the

"Filter", "SELECT" and "LIKE" tabs, on the different windows that appeared in sequence. Finally, we reached a writeable box where we entered the word "FLAIR". After submitting the operation through the dedicated button, results from our search showed a total amount of 735 records.

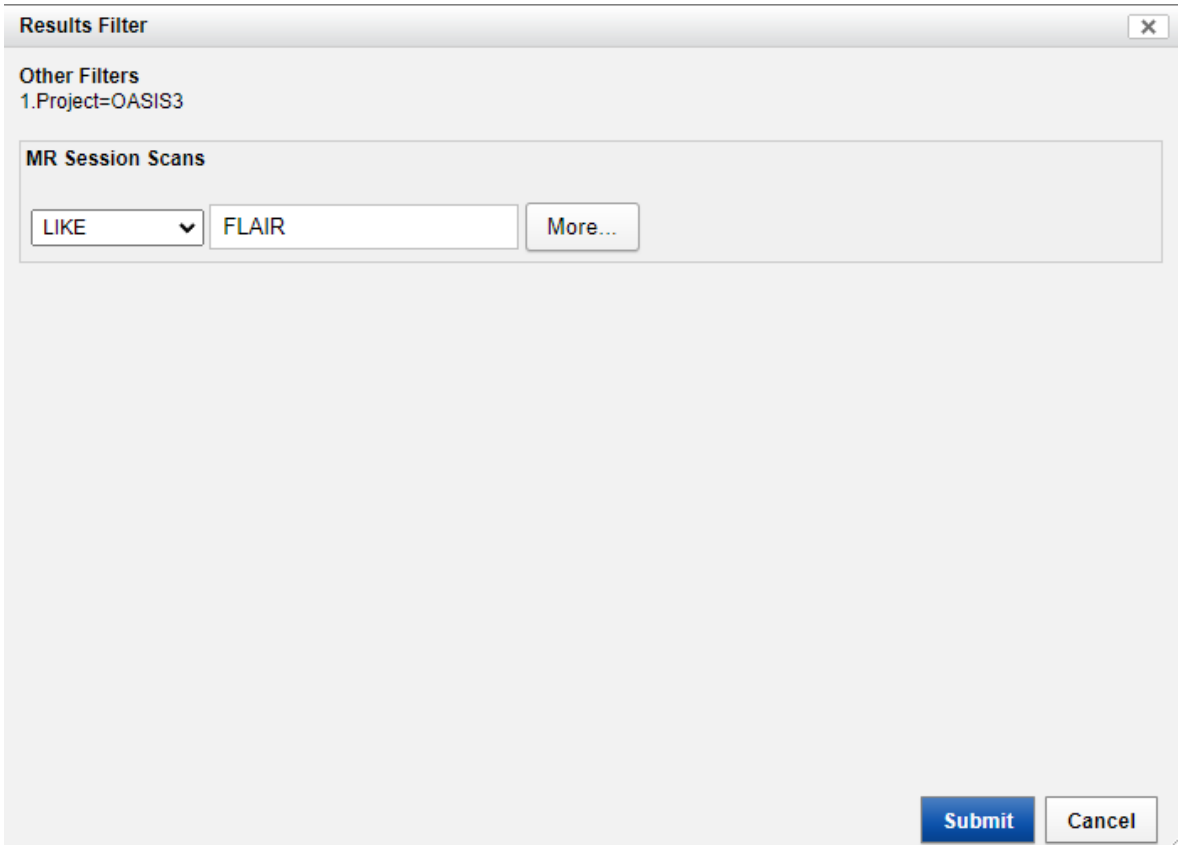


Figure 3.13. Filtering option window from XNAT.

At that point, data had to be downloaded. By clicking on "Download Images" in the "Actions" panel, it was possible to select the desired scans from the "Scan Types" section of the dedicated window. FLAIR and T1-weighted images were specified and, after submitting the procedure, their imaging files we obtained.

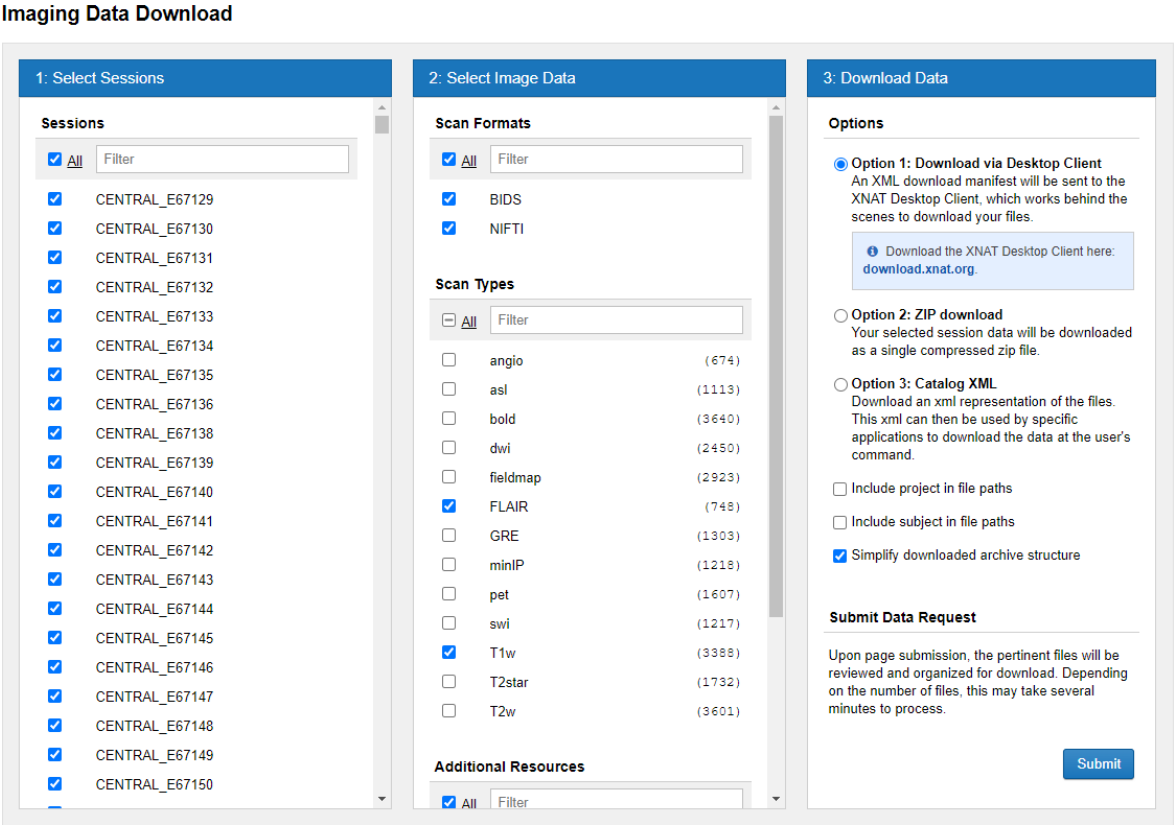


Figure 3.14. The XNAT download page. All files and modalities are already selected by default, but the user can modify according to his own needs.

On the other hand, for clinical records we followed a different strategy. Data were downloaded entirely from the repository and selected only afterwards.

Unfortunately, the download procedure was not as straightforward as the one of the imaging data, requiring multiple steps. First, we had to load the main Clinical Data tab – namely "ADRC Clinical Data" – from the OASIS3 repository using the "Add Tab" scroll down menu. After that we had to join the ADRC data frame to the other available ones trying to add as much information as possible. This step was performed through the "Option" and "Join to ..." buttons which prompted the window presented in Fig. 3.15. From there we selected, one at a time, all the available spreadsheets – from "UDS A1: Sub Demos" to "UDS D1: Clinical Diagnosis" – and checked the "detailed" option. Eventually, we clicked the "Add" button and repeated the procedure.

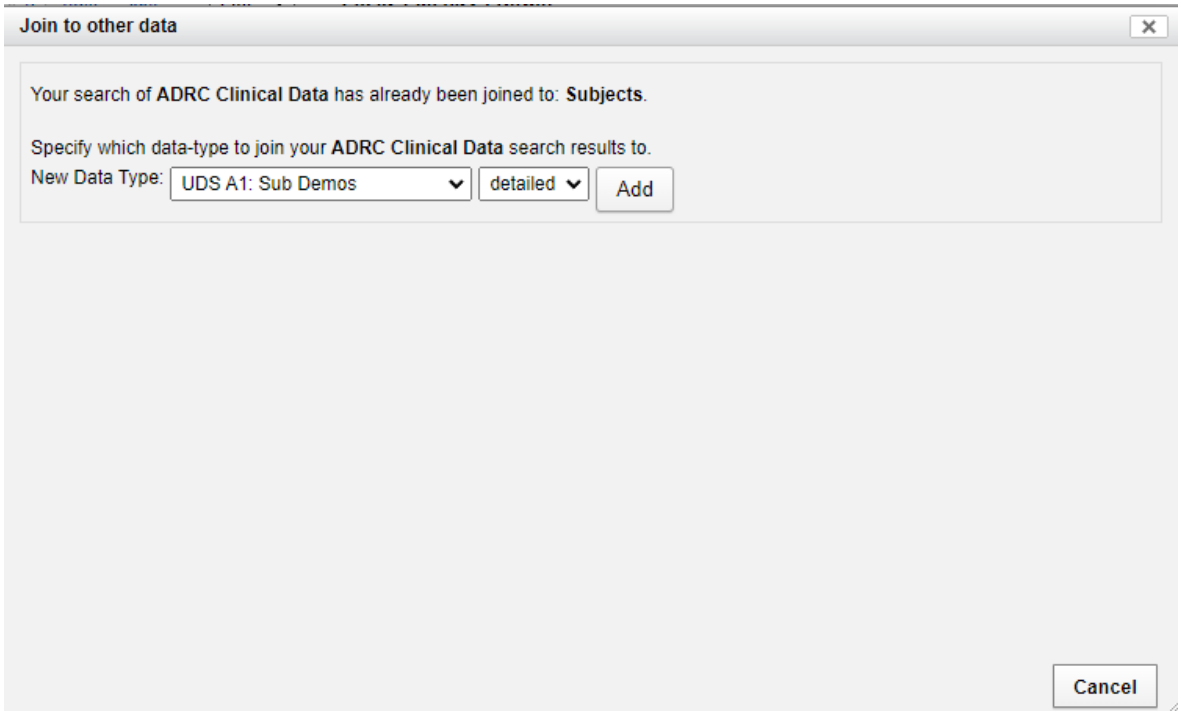


Figure 3.15. "Join to other data" window from XNAT.

Once completed the operation, we saved the resulting file in csv format using the "Options" and "Spreadsheet" buttons.

At this point, we had to select from the available spreadsheet only the records that matched the downloaded imaging sessions. Indeed, as already mentioned, MRI scans and Clinical visits were not always conducted during the same day, thus raising the need to establish a coupling. In an attempt to accomplish this task, we developed a Python code using the "Pandas" library, which sorted the MR Sessions trying to find their corresponding Clinical visits according to a time gap minimization criterion.

Once we identified the clinical records having an imaging counterpart, the data selection process was eventually completed. At that point, for patients with multiple MR session we also eliminated all the available records other than the most recent. This was done in order to avoid having multiple instances from the same subject, that could create biases when fed to a ML model, affecting both the obtained results and further statistics analysis.

In the Appendix, section 5.12 "Dataset join" there is an example on how to create the link between imaging and clinical records.

Both the imaging and clinical variables were then fully available and ready to be filtered and processed.

3.2.1.2. Imaging variable extraction

The imaging session downloaded from the OASIS3 repository were used to extract one of the main variables involved in classification: the WMH volumes. To derive it, we used BIANCA pre-trained with the WH and UK Biobank dataset from (Bordin et al). We obtained results after testing on the 735 subjects selected from OASIS3. We chose the mixed-training strategy instead of the single-site one (both training and testing on OASIS3), in order to avoid some of the bias that could have been introduced when including training and testing subjects in the same dataset. The 40 labeled subjects could have been discarded, but we wanted to avoid any possible data loss.

The procedure used to derive lesion probability maps from BIANCA is identical to the one presented in Par. 3.1. After being renamed, the imaging files from the test set went through the pre-processing steps seen before: brain extraction, biasfield, correction, registration and bianca masking. A masterfile was eventually created allowing us to run BIANCA with separate training and testing procedures.

When we finally obtained the lesion probability maps, WMH volumes were extracted using "bianca_cluster_stats". This FSL script is launched via Linux or MacOS terminal using the following syntax:

bianca_cluster_stats <bianca_output_map> <threshold> <min_cluster_size> [<mask>]

which is very similar to that of "bianca_overlap_measures". The call asks four arguments in order to work – three compulsory and one optional. They are all detailed below:

- *<bianca_output_map>* is the lesion probability map, output from BIANCA after the testing procedure;
- *<threshold>* represents the threshold value applied to the lesion probability map in order to binarise it (set to 0.9 in our case);
- *<min_cluster_size>* is the minimum dimension a cluster should have in order to be considered as a WMH lesion;
- *<mask>* is a parameter that, if specified, prompts the script to additionally calculate the number of clusters within the specified manual mask.

The outputs provided by this script are: i) the number of WMH clusters found for a certain subject; ii) the total WMH volume.

We are clearly interested in the latter one. However, the value needs to be normalized over the whole brain volume before being fed to the classification algorithm. This step is required since patients may have different brain dimensions. A higher absolute lesion volume could thus be attributed to differences in the anatomical sizes rather than to more extended WMHs.

The normalization term can be derived, utilizing the WM, GM and CSF maps output by the "fsl_anat" script. Then, mean values can be calculated for the three volumes by means of "fslstats". Summing the obtained results, we get an estimation of the whole brain volume. Eventually normalization can be performed by simply using this quantity to divide the total WMH volume output from the "bianca_cluster_stats" script. Results were finally saved into a "csv" file for all the different subjects included within our dataset.

Once again, the full list of utilized codes can be found in the Appendix, section 5.11 "FLAIR T1w run".

3.2.1.3. Clinical variables extraction

After being collected, the clinical variables underwent a thorough filtering procedure. The aim was to reduce the dimensionality of the dataset, that initially counted 807 variables. To pursue this goal, we used the following strategies: i) we tried to remove the enormous amount of missing data from variables by retaining only the first 10 with the lowest number of NaNs. The remaining variables were instead discarded; ii) eventually we used the study from (Bordin et al.) as reference to check the results of our data selection process. Indeed, that research built its own prediction model including the most relevant clinical variables for the Whitehall and UK Biobank datasets. Therefore, we wanted to check whether the ones identified for OASIS3 matched somehow those of larger populations.

From the selection procedure the following variables were extracted:

- Age
- Height
- Weight
- Sex
- Diabetes (if a patient is affected or not)
- Hand class (right- or left-handed)
- Tobac 30 (if the patient smoked in the last 30 days)
- Tobac 100 (if the patient smoked in the last 100 days)
- Alcohol (if the patient regularly drinks alcohol)

- Age at Entry (the time (in years) that the patient has been part of the study inserted in OASIS3)

Among them all had a match with variables from the reference study, except for "Age at Entry".

3.2.2. Model creation

In this section we present the main steps involved in the creation of our models. We start by processing the imaging and clinical variables obtained after the data selection process (see Par. 3.2.1). Then we carry on with an explanation of the main strategies used for building the SVM, RFC, and ANN classifiers. We conclude presenting the evaluation metrics used to assess their performance. All the discussed procedures were carried out via Google Colab (see Par. 2.2.6).

3.2.2.1. Data pre-processing

Data-processing was initially applied to the target variable, that is CDR. The different classes representing its values highlighted the presence of a strong unbalance. Indeed, we had the following numerosity:

- CDR = 0 → 588 subjects
- CDR = 0.5 → 47 subjects
- CDR = 1 → 27 subjects
- CDR = 2 → 1 subject

In an attempt to limit the disparity, we decided to binarize the variable using a threshold equal to 0. Therefore, we had:

- CDR = 0 if the old CDR was equal to 0
- CDR = 1 if the old CDR was higher than 0

To perform this change, the populations characterized by a CDR > 0 had to be united. However, the procedure was carried out only after comparing the WMH volumes to check that no significant disparity was present between the group with CDR = 0 and the remaining ones (with CDR > 0). The analysis was conducted using an ANOVA test (see Par. 2.3.1 for details).

As for the input variables, some pre-processing was also required to. In particular, the following operations were performed:

- All records containing missing values were eliminated.
- All the categorical records encoding missing or unknown values were eliminated (an example is represented by Diabetes, in which 0 means "no diabetes", 1 means "presence of diabetes", 8 means "unknown" and 9 means "no data"; in this case all the records containing 8 or 9 were eliminated).
- The variable "Age at Entry" was modified: it originally represented the patient's age when he first was enrolled within the study; then we arranged it to contain the number of years since its entry (i.e., "Age" – "Age at Entry")
- Categorical variables were transformed into binary variables (an example is represented by the "M/F" variable, whose values could either be: M indicating Males or F, indicating Females; a new variable – called 'sex' – was therefore created, and substituted to the former; its values were set to 1 for Male subject and 0 for Females). This step, alongside the next one, was embedded in the ANN model we built. On the other hand, it was performed prior to the training procedure for both RFC and SVM.
- Continuous variables were normalized (i.e., transformed to have values of mean and variance respectively equal to 0 and 1).

After these steps were finally completed, variables were ready to feed the ML models.

3.2.2.2. Model building strategy

When building a ML model, the first step that needs to be performed is the dataset split. A *training* and *testing sets* must be derived. However, when hyperparameters have to be tuned the former takes the name of *real training set* and is further split in the *training* and *evaluation* (alias, *validation*) *subsets*. The idea is to iteratively train the model with different combination of parameters (using the *training subsets*) and select the one providing the best performance (evaluated according to a pre-defined criterion) on the *evaluation subset.* That optimal combination is eventually selected to conduct the final training of the model, which uses the *real training set.* Instead, performance are evaluated on the original *testing subsets,* previously left aside.

To avoid results of the hyperparameters tuning step being dependent on the split procedure performed to derive the *training* and *evaluation subsets* from *real training*, a cross-validation (CV) approach is generally used. In this case the *real training* is k-foldsplit in k sub-sets, each taking turns in being the hold-out validation set. The model is therefore trained on (k-1) folds and measured on the remaining held-out one. The overall performance is then evaluated by averaging the results given by the k different folds. In this way every possible bias affecting results is removed. As

an example, a 5-fold CV means that the *real training* is divided 5 times in two parts: one having 20% of the original data for *evaluation*, while the remaining 80% is used for *training*. In case of a 10-fold CV the *real training* is divided 10 times, each using 10% of data for *evaluation* and 90% for *training*. This option is extremely useful in case of large datasets, where it allows for a fine tuning of the hyperparameters. However, it can significantly impair performance for small and/or unbalanced datasets. As for the latter, the reason is a significantly higher possibility of having only examples from the majority class in many of the k-folds (or to have a strongly unbalanced distribution anyway) used to tune parameters. This is even more true when the low dataset numerosity reduces the dimension of the *real training set* and of the fold themselves. In such situations, the label of the majority class will be output by the model regardless of the input data received, since it was the one mainly experienced during training. Results of the hyperparameters tuning are therefore not reliable.

In the context of our thesis, this training strategy was either performed 'manually' or using a Grid Search approach. As for the latter, we implemented it using a python script from the "Sklearn" library, called "GridSearchCV". To properly run, it needs the parameters we want to tune, the model type, and the metric used to evaluate performance ("scoring" parameter), which – in our case – will always be set to "balanced_accuracy". This to instruct the model on the lack of balance existing between different data classes. The issue is thereby adjusted by assigning higher scores to the correct classification of the minority samples. Information on the type of cross validation we want to use is also fed to the "GridSearchCV" script. The default value is set to "5-fold CV" and was never changed as for the purposes of our thesis.

The GridSearchCV variable contains information on the parameters we want to tune and on the selected evaluation strategy. However, it's the command "fit" – after being fed with the actual data – that start the overall procedure. The function applies a k-fold CV to the *real training set* and finds the optimal combination of hyperparameters, which is saved. Afterwards it uses it to carry out the actual training and testing procedure on the investigated model.

Eventually, it must be pointed out that – for each of the investigated ML models – three different dataset splits were performed, according to the ratio between target variable classes. We had, accordingly:

- One split with a 1:1 ratio of CDR = 0 and CDR = 1 subjects (including 96 records of training and 33 records for testing)
- One split with a 2:1 ratio of CDR = 0 and CDR = 1 subjects (including 135 records of training and 46 records for testing)

- One split with a 3:1 ratio of CDR = 0 and CDR = 1 subjects (including 181records of training and 61 records for testing)

Performance was evaluated separately for each case.

3.2.2.3. Support Vector Machines

Among the investigated ML models, SVM was the very first. The analysis was carried out entirely using the "Sklearn" library and followed the procedure described in Par. 2.2.2.2.

The dataset was split three times, thus creating different *sets* of *(real) training* and *testing* according to the ratio between samples with CDR = 0 and samples with CDR = 1. Performance were then evaluated for every case to assess whether class balance had a significant impact on results.

As for the hyperparameter tuning, different combinations of the following values were considered:

- C: [0.1, 1, 10, 100, 1000]
  default=1.0
  Regularization parameter. The strength of the regularization is inversely proportional to C. Its value must be strictly positive.

- gamma: [1, 0.1, 0.01, 0.001, 0.0001]
  Kernel coefficient for 'rbf', 'poly' and 'sigmoid'.

- kernel: ['rbf', 'sigmoid']
  Specifies the kernel type to be used in the algorithm.

- class_weight: [None, 'balanced']
  If not given, all classes are supposed to have weight one. The "balanced" mode uses the values of the *target* to automatically adjust weights inversely proportional to class frequencies in the input data.

The best combination was found through the Grid Search approach and was eventually used for training. Finally, performance was derived from the *testing set* using the evaluation metrics described in Par. 3.2.3.

3.2.2.4. Random Forest Classifier

As for the RFC, we followed the same procedure except for one step: the introduction of PCA. This ML strategy was adopted not to perform dimensionality reduction but rather to support the RFC in classification, thus improving results. Indeed, the major pitfall of Regression Trees is that they can split the dataset at every node using only one variable at a time. To overcome this issue a linear combination of variables can be used. This strategy, called "oblique splitting", is quite recent and hasn't been implemented in the "Sklearn" library for RFC yet (Katuwal, Suganthan, and Zhang 2020). Therefore, we proceeded applying PCA on the 5 continuous variables of our Dataset (Age, Age at Entry, Height, Weight and Bianca Volume). They were replaced with 5 new variables obtained as a linear combination of the originals. The RFC model was initially trained using PCA. However, a second round of analysis was also conducted without this strategy.

After PCA, the analysis steps remained unaltered. The dataset was again split three times, according to the class balance of samples. Performance were then evaluated for every pair of *(real) training* and *testing set*.

The investigated hyperparameters were the following:

- 'n_estimators': [50, 100, 200, 500, 1000]
  Number of trees in the forest.

- 'max_features': ['auto', 'sqrt', 'log2']
  Number of features to consider when looking for the best split

- 'max_depth' : [4,5,6,7,8]
  Maximum depth of the tree. If set to *None*, nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.

- 'criterion' :['gini', 'entropy']
  Function measuring the quality of a split. Supported criteria are "gini" for the Gini impurity and "entropy" for the information gain. Note: this parameter is tree-specific.

- 'class_weight' :['balanced', 'balanced_subsample', None]
  Weights associated with *target* classes. If not given, all classes are supposed to have weight one.

Also in this case, the optimal combination was found through the Grid Search approach and was eventually used for training. Again, performance was calculated on the *testing set* using the evaluation metrics described in Par. 3.2.3.

3.2.2.5. Artificial Neural Network

An ANN was also built. However, these models are characterized by a very low explainability which is caused by the numerous sequential operations performed on every variable and by the impossibility of accessing information from the hidden layers. Therefore, in the context of our project, the ANN was only used as reference to evaluate performance of the SVM and RFC classifiers and to confirm that a certain degree of accuracy can be reached for this dataset.

The analysis was carried out using the same data preparation approach of the former cases. The dataset splitting procedure was also maintained identical (3 different pair of *(real) training* and *testing sets* were evaluated). However, since "Sklearn" does not have any pre-defined model to build ANNs, we had to implement its structure from scratch using the library "Keras". The creation of the neural network was quite straightforward and required:

1. Input neurons (11, as there are 11 variables);
2. An encoding and normalization layer (one neuron for each variable);
3. A dense, fully connected layer with 32 neurons having a Rectifier Linear Unit activation function;
4. A dropout layer, that helps prevent overfitting, diminishing the complexity of the net;
5. A single layer with sigmoid activation function to output results.

In Fig. 3.16. we can observe a graph summarizing the overall network design.

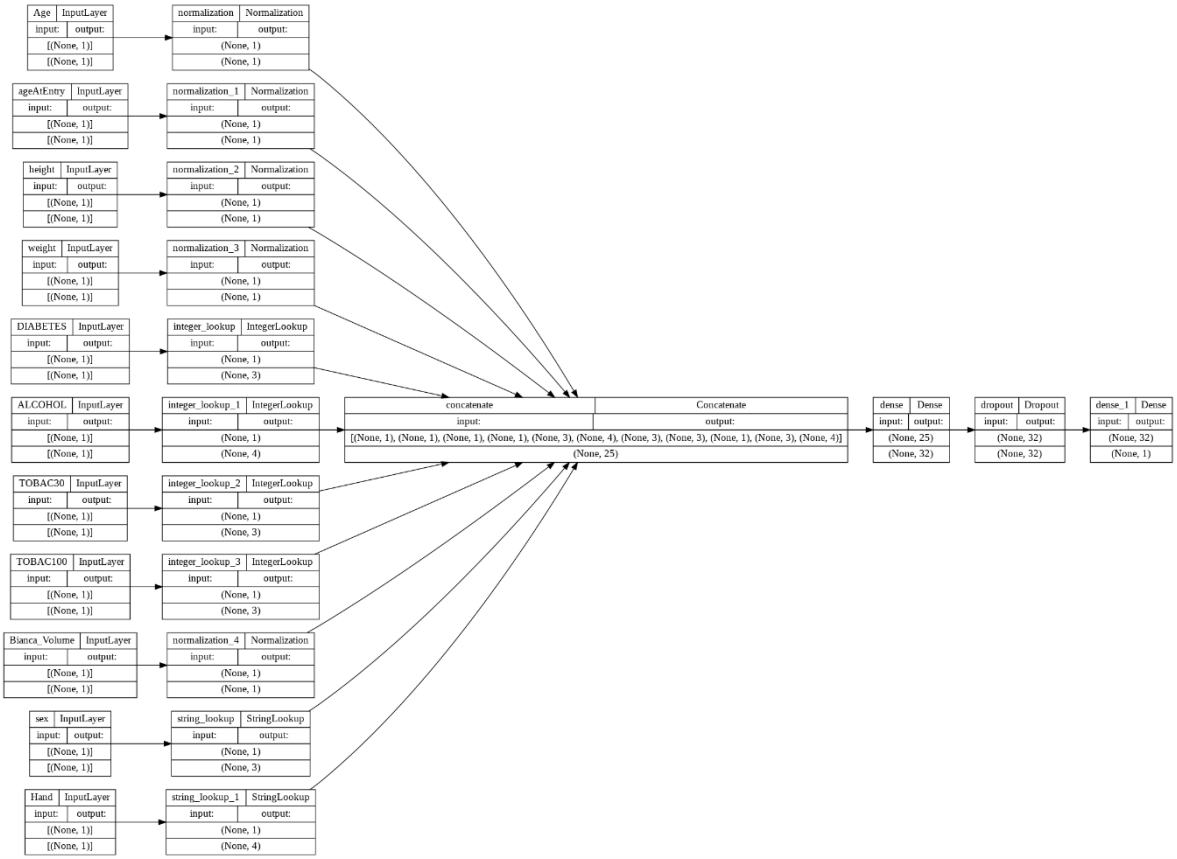


Figure 3.16. Overall graph of the designed ANN.

The lack of ANN models in "Sklearn" also prevented us from tuning hyperparameters using the "GridsearchCV" function. For this reason, we decided to manually train the ANN models multiple times, each evaluating a different combination of the following parameters.

- The presence or absence of initial bias (a pre-set value for neurons)
- The presence or absence of class weights

If we consider that this procedure had to be repeated for every dataset split involved within the analysis (they were 3) we run a total of 12 computations.

3.2.3. Machine Learning metrics

There are many indexes and metrics to evaluate the performance of ML algorithms but, as for binary classifiers, the easiest and most immediate one is represented by confusion matrixes. They are specific table layout reporting in four different panels the amount of testing samples that fall in each of the following categories: True

Positive, False Positive, False Negative and True Negative. Their characteristics are explained hereafter:

- The True Positive (TP) group contains samples whose actual and predicted class are both equal to 1. The classification is thus correct.
- The False Positive (FP) group contains samples whose actual and predicted class are equal to 0 and 1, respectively. The classification is thus incorrect.
- The False Negative (FN) group contains samples whose actual and predicted class are equal to 1 and 0, respectively. The classification is thus incorrect.
- The True Negative (TN) group contains samples whose actual and predicted class are both equal to 0. The classification is thus correct.

An example of confusion matrix is presented in in Fig. 3.17.

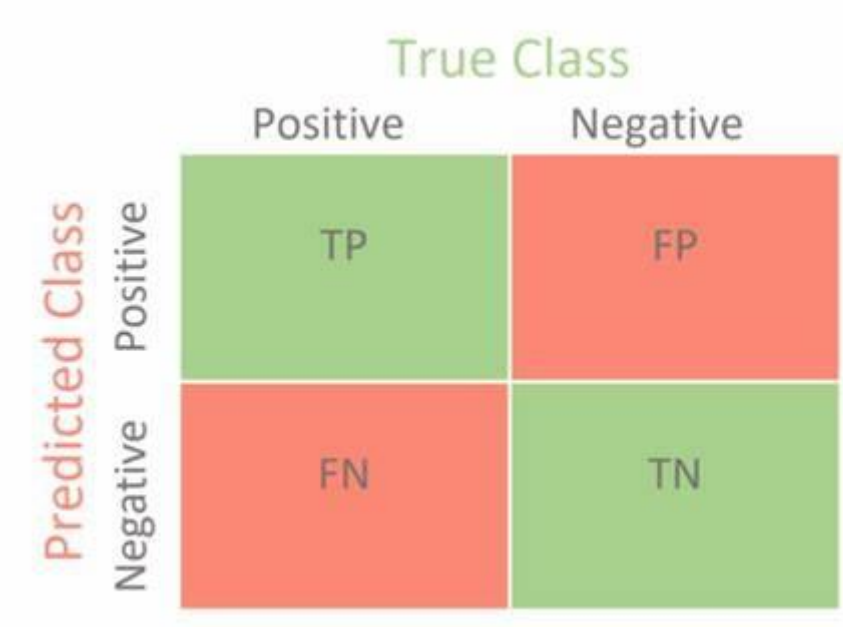


Figure 3.17. Example of a Confusion Matrix.

Aside such basic indexes, derived ones are also used to carry out the assessment. Among the most common, we find: accuracy, precision and recall.

Accuracy is defined as the number of correct predictions divided by the total amount of predictions performed. It provides an easy and fast way to acknowledge the quality of results. However, since it does not distinguish between the numbers of correctly classified examples belonging to different classes, when the dataset is heavily unbalanced, it is no longer a proper measure to evaluate performance. As for our thesis, the ratio between classes reaches a maximum value of 3:1.

Precision, on the other hand, is more appropriate when dealing with unbalanced datasets, as it expresses a ratio between the number of True Positives and the amount of Total Predicted Positives (True Positive + False Positive). This index explains how capable the model is to correctly classify Positive values while maintaining a low False Positive discovery rate and its value approaches 1 when the amount of the latter decreases.
Eventually, Recall represents the ratio between the number of True Positives and the amount of Total Actual Positives (True Positive + False Negative). It is very similar to Precision and explains how capable is the model to correctly classify samples belonging to the positive class. Its value approaches 1 when the model is able to correctly classify most of the Positive samples.

A last important metric is represented by the F1-Score, which is the harmonic combination of the last two indexes. It is defined as:

$$F_1 = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}}$$

The F1-score gives a proper evaluation even for unbalanced classification problems. This evaluation metric can be used for imbalanced Classification tasks alongside the Accuracy score for a better overall evaluation. Its value is close to 1 when performance of the model tends to a perfect classification.

All of the indexes here described were used to assess results from the different ML models implemented during our thesis.

3.3. Statistical analysis

In this section we present the statistical test used to draw comparisons between results from both the segmentation and the prediction model parts.

3.3.1. ANOVA test

ANOVA stands for Analysis of Variance. It's a parametrical test developed by Ronald Fisher in 1918 that has been widely used ever since in many different applications. Its ability is to find statistical differences between the means of two or

more independent groups, thus determining whether they come from the same population or not.

There are many different types of ANOVA but, during this thesis, only two will be used: the standard ANOVA (also known as one-way ANOVA) and the Repeated Measure ANOVA (also known also as RANOVA). The first is used to test differences between independent (unrelated) populations whereas the second is used when the compared populations contain measures of the same subjects taken at multiple time points (thus being related).

ANOVAs are hypothesis tests, which means they work by either confirming or rejecting a pair of alternative hypotheses – respectively called "null" and "alternative". The first is accepted when mean values from the compared populations present no significant difference. Data can thus be considered comparable. Conversely, the "alternative" hypothesis is accepted when the mean value of at least one population is different from the others.

The value which determines the presence or absence of the actual difference is called p-value and represents the main outcome of ANOVA. Its cut-off is usually set to 0.05 (5%): when its value is higher, we reject the alternative hypothesis (thus concluding there is no significant difference between populations); on the other hand, when its value is lower, we reject the null hypothesis (thus concluding a significant difference is present).

Another important parameter for ANOVA is the F-Statistic value, which can be expressed as:

F-statistic: Variation between sample means / Variation within samples

The larger the F-statistic, the greater the variation between sample means over the variation within samples means. This provides a greater evidence of the difference existing between groups.

4 Results

In this chapter we present the results obtained from all the analysis steps described in the "Materials and Methods". Similarly, this section is divided in two parts: the first is focused on the evaluation of BIANCA performance according to different training conditions, while the second is focused on the classification model we built aiming to predict CDR.

4.1. Evaluation of BIANCA performance

In this section we present the results obtained from BIANCA after carrying the training with the conditions presented in Par. 3.1.5.2 – 3.1.5.4: an increasing number of training subjects, a variable combination of MRI contrasts, and a pre-defined training set built using data from Whitehall and UK Biobank.

Two examples of visual assessment of segmentation are presented in Fig. 4.1.

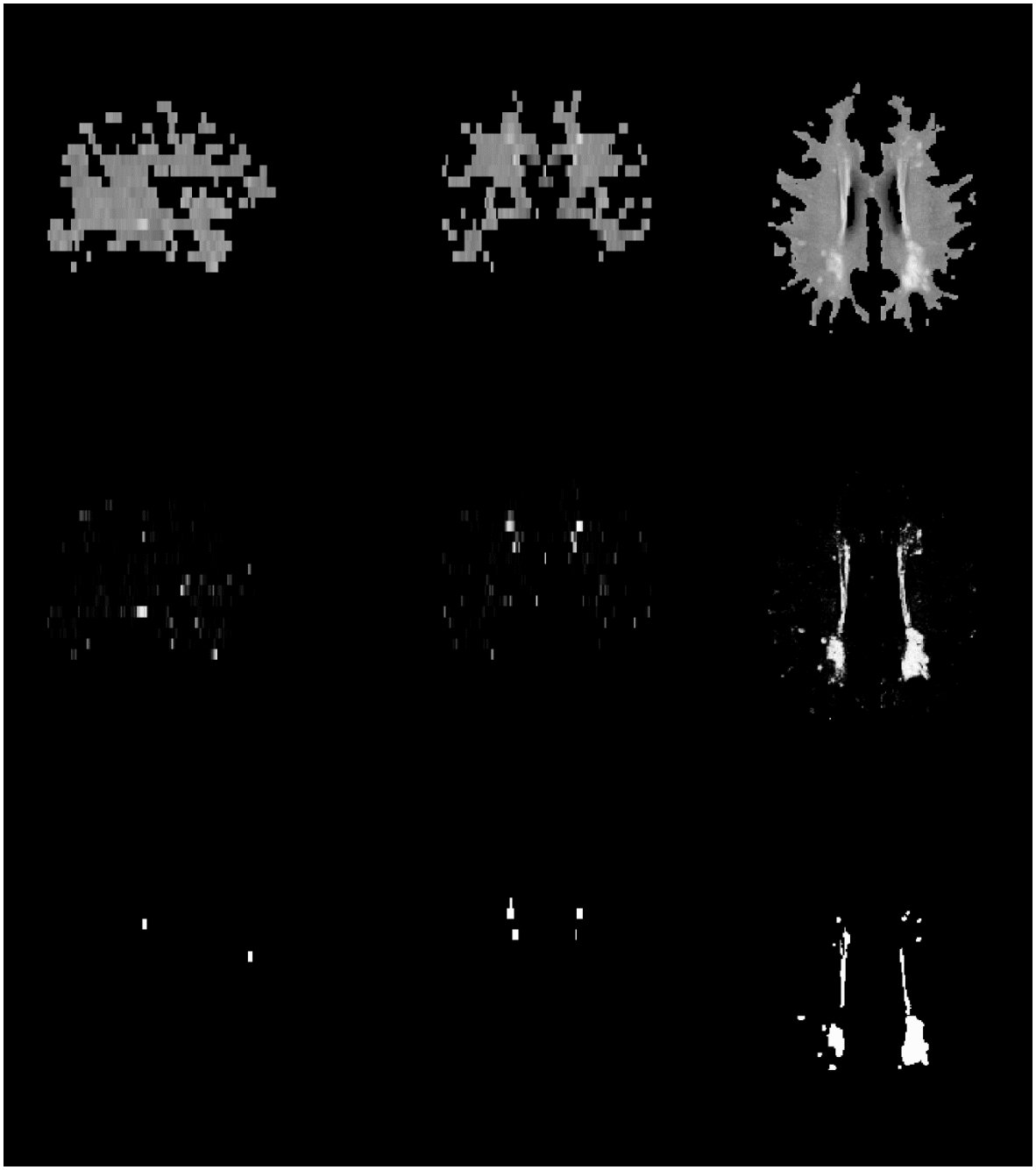


Figure 4.1. Visual assessment of the results: WM extracted from FLAIR (top panel); output from BIANCA (middle panel); manual segmentation mask (bottom panel).

4.1.1. Incremental training analysis

In this paragraph we present results relevant to the incremental training analysis. We repeated the procedure for each combination of MRI modalities described in Par 2.1.5.3 (FLAIR only, FLAIR + T1w, FLAIR + SWI, FLAIR + T1w + SWI, SWI

only), therefore we display 5 panels. For each of them we compare the training results obtained with an incremental number of subjects (10, 15, 20, 25, 30 – color-code: see legend) and next tested on the whole dataset. Testing was based on a 4-fold cross validation, with random subsets, which is known to reduce statistical biases (see Par 3.1.5.2 for further details). Results are evaluated using both "preliminary" and "expert" manual masks and are presented in terms of DICE in Fig. 4.2 and Fig 4.3, respectively.

As for the "preliminary" manual masks, a noticeable increase in DICE can be observed in correspondence to higher number of subjects involved in the training procedure. This is valid for all the combinations of MRI modalities used within the analysis except for the "SWI only" case which, however, did not undergo any statistical analysis due to the extremely low performance it obtained. The presence of differences is supported by results from a RANOVA tests that gives a p-value < 0.05. Post-hoc tests are also run for adjacent pairs of training subjects (N vs N+5 subjects – with N ranging from 10 to 25).

Results acknowledge a statistical difference for all cases except for the "25 vs 30". Its p-value, according to the combination of MRI modality used, is reported as follows: 0.786 for the "FLAIR only" case, 0.667 for the "FLAIR + T1w" case, 0.568 for the "FLAIR + SWI" case and 0.799 for the "FLAIR + T1w + SWI" case. The remaining pairs have, on the other hand, a p-value < 0.001. For such cases, black lines are drawn between the median value of each boxplot, to make differences more noticeable.

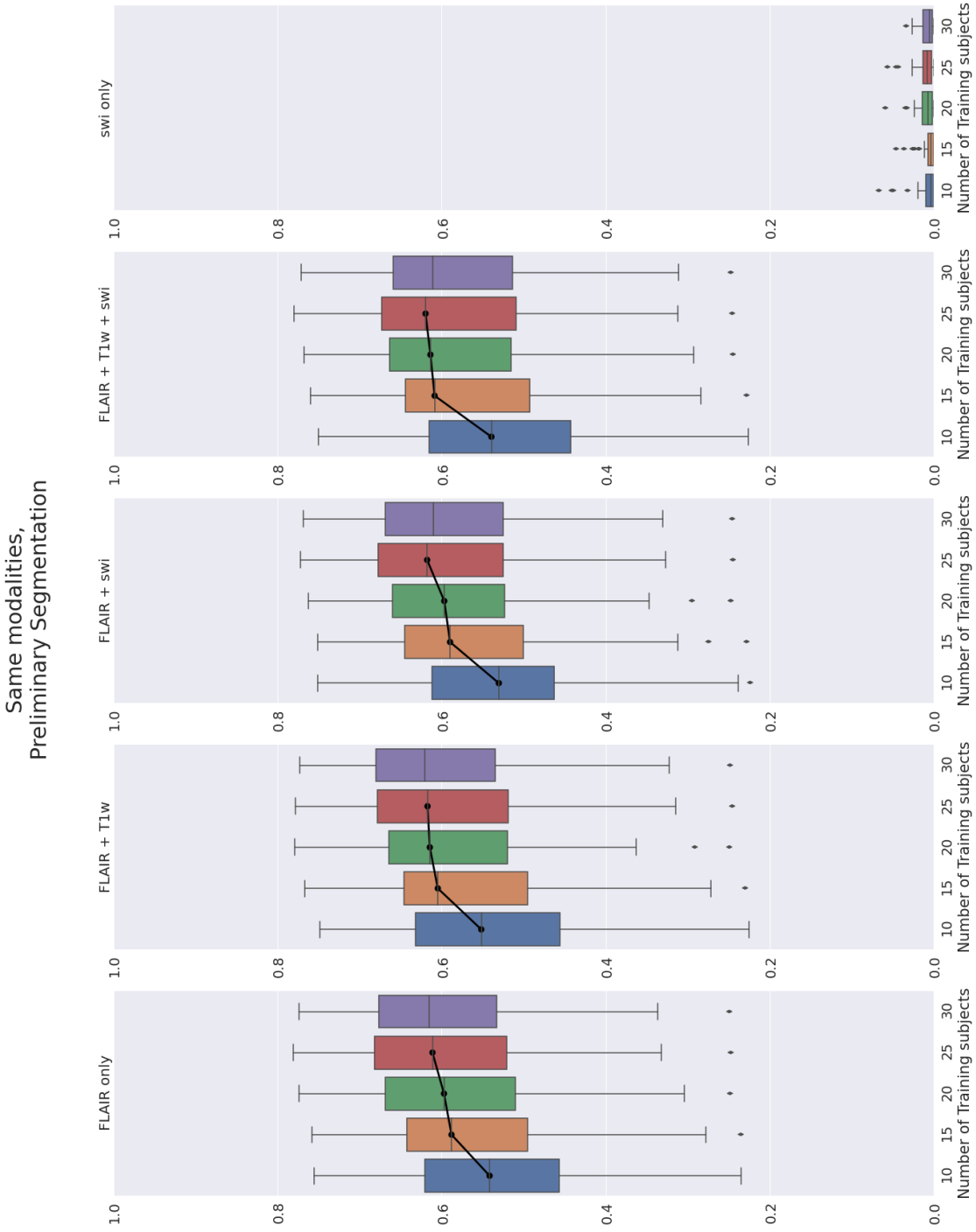


Figure 4.2. Boxplot of the DICE index (represented on the y axis) between BIANCA outputs obtained training with an increasing number of subjects (represented on the x axis) and the corresponding "preliminary" segmentations. Results are relative to the whole dataset (40 subjects). Black lines indicate significant differences (p-value < 0.05) between adjacent groups, calculated using a RANOVA test. The analysis is repeated in every panel for the different combinations of MRI modalities used for training.

As for the "expert" manual masks, results are very similar to the ones displayed above. Therefore, all the previous considerations remain valid.

Only one difference can be noticed, and it regards the results obtained from post-hoc comparisons of the RANOVA tests. As shown in Fig. 4.3, the black lines which marked the 10 vs 15, 15 vs 20 and 20 vs 25 comparisons in the "preliminary" segmentations case are now present only for the first two pairs (10 vs 15 and 15 vs 20). Such differences are demonstrated by p-values < 0.001 for every combination of MRI modality involved, except for the "SWI only" which, again, was not involved in the statistical analysis due to its scarce performance.

On the other hand, the non-significant comparisons ("20 vs 25" and "25 vs 30") have the p-values reported as follows:

- 20 vs 25: 0.991 for the "FLAIR only" case, 0.152 for the "FLAIR + T1w" case, 0.153 for the "FLAIR + SWI" case and 0.251 for the "FLAIR + T1w + SWI" case.

- 25 vs 30: 0.486 for the "FLAIR only" case, 0.546 for the "FLAIR + T1w" case, 0.791 for the "FLAIR + SWI" case and 0.849 for the "FLAIR + T1w + SWI" case.

A final consideration can be drawn as for the difference between "preliminary" and "expert" segmentations. Higher DICE values are reported across all combination of MRI modalities (except for the "SWI only" case) when "expert" segmentations are used, with median values > 0.6 for the first boxplot of every panel (10 training subjects case). On the other hand, the median values relative to "preliminary" segmentations are all < 0.6 for the first boxplot of every panel (10 subjects training segmentation).

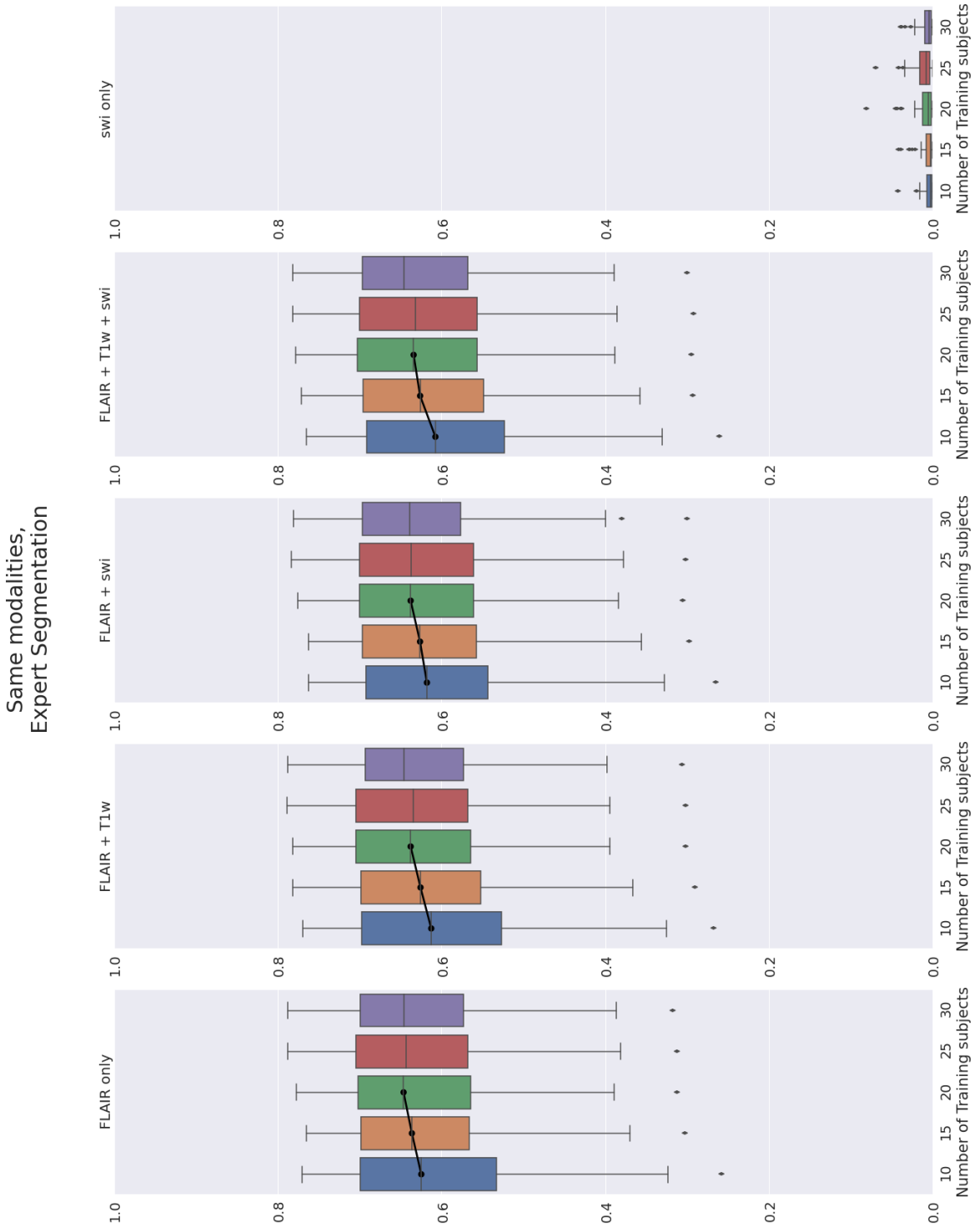


Figure 4.3. Boxplot of the DICE index (represented on the y axis) between BIANCA outputs obtained training with an increasing number of subjects (represented on the x axis) and the corresponding "expert" segmentations. Results are relative to the whole dataset (40 subjects). Black lines indicate significant differences (p-value < 0.05) between adjacent groups, calculated using a RANOVA test. The analysis is repeated in every panel for the different combinations of MRI modalities used for training.

4.1.2. Intra-rater variability

At this point we present results of the intra-rater variability analysis. Manual masks from the "preliminary" and "expert" rounds were compared by means of DICE for the 40 subjects involved in our study. Results are presented in the boxplot of Fig. 4.4, that displayed a median value of 0.79.

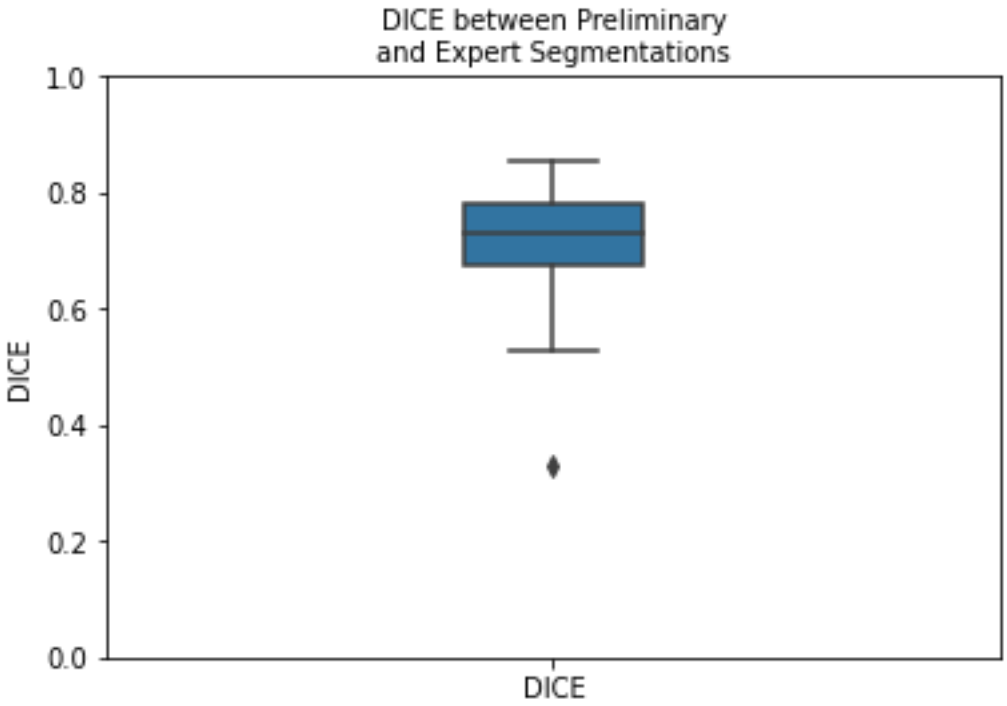


Figure 4.4. Boxplot of the DICE between "preliminary" and "expert" manual masks for the whole dataset involved in the study.

The visual inspection process highlighted the presence of slight imprecisions in the first round of segmentations, that missed small clusters of WMH lesions. An example is reported in Fig. 4.5, where "preliminary" and "expert" manual masks are reported in yellow and red on the central and right panel, respectively. The lack of correspondence for certain periventricular regions in clearly visible and resulted in a DICE of 0.59.

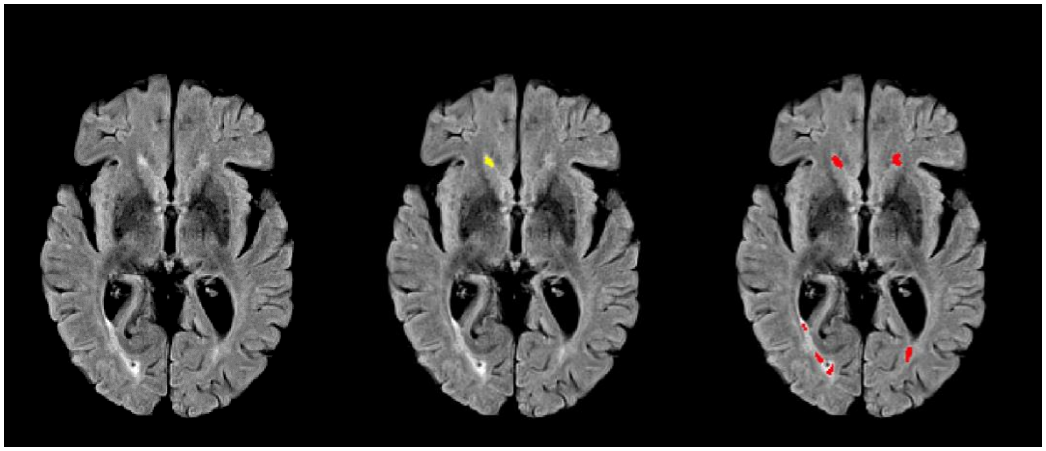


Figure 4.5. Visual example of the difference existing between "preliminary" (central panel – yellow area) and "expert" (right panel) segmentations, outlined on a FLAIR image (left panel – red area).

After the current analysis, subsequent steps were carried out using solely the "expert" manual masks to evaluate BIANCA performance.

4.1.3. Multimodality analysis

In this paragraph we present results relative to multimodality. We carried out the analysis by fixing the number of training subject to its highest value. Indeed, the entire dataset was involved through a leave-one-out validation. We then varied the combination of MRI contrasts assessing the options described in the previous chapter: FLAIR, FLAIR + T1w, FLAIR + SWI, FLAIR + T1w + SWI. Results were evaluated using "expert" manual masks. The distribution of DICE for every option is presented in Fig. 4.6.

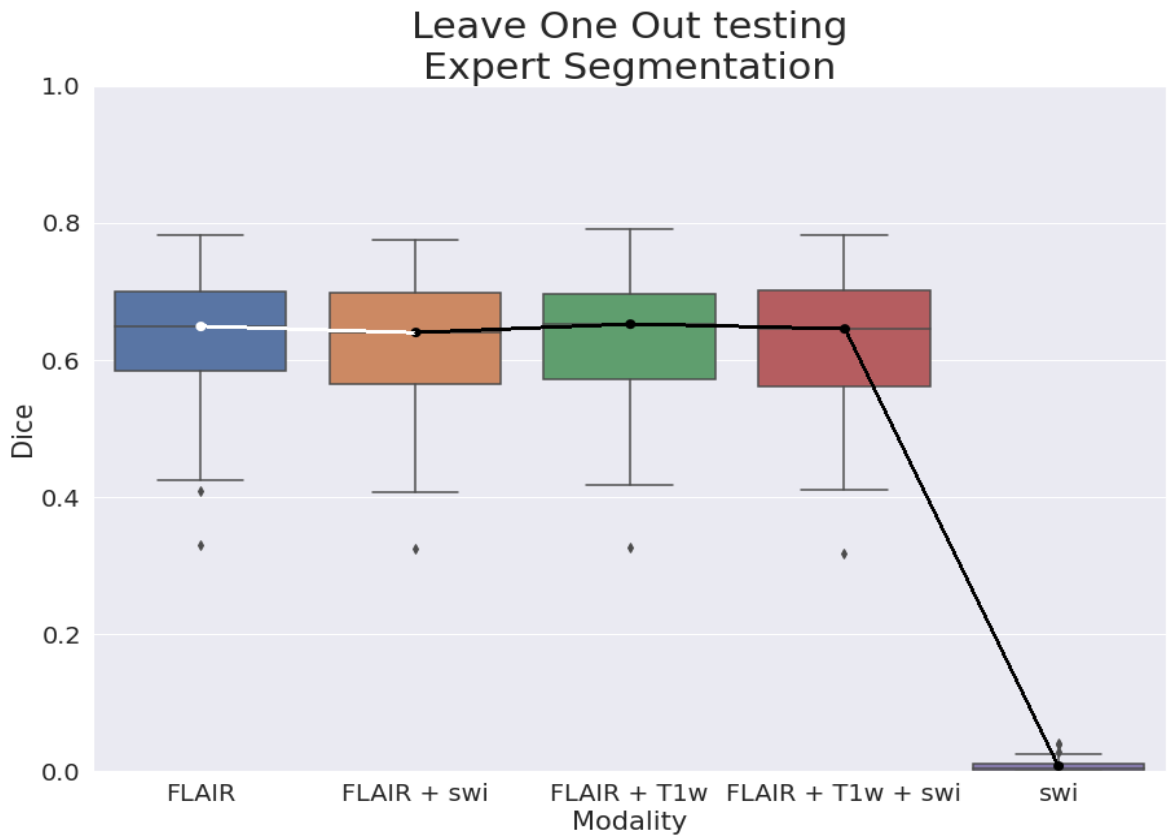


Figure 4.6. Boxplot of the DICE index (represented on the y axis) between BIANCA outputs obtained training and testing with leave-one-out validation on the whole dataset (40 subjects). Results are relative to the different combination of MRI modalities used (represented on the x axis): FLAIR (in blue), FLAIR + SWI (in orange), FLAIR + T1w (in green), FLAIR + T1w + SWI (in red), SWI (in purple). Full lines indicate significant differences (p-value < 0.05) between adjacent distributions, calculated using a RANOVA test.

The first thing that can be noticed is the high difference in performance between the compared distributions, especially for the "SWI" case. This is further confirmed by the RANOVA test that provides a p-value < 0.001.

Post-hoc comparisons are also evaluated for every pair of MRI modalities involved in the study. For an easier visualization, they are displayed in six separate panels (one for every pair) along with information about their p-values. The "SWI" case is excluded from this statistical test due to the clear existence of a significant differences with respect to the remaining distributions. As regards p-values, it is worth mentioning that the full lines are drawn across boxplots to indicate the presence of a statistical difference between certain pairs (pvalue < 0.05). The level of significance is furthermore indicated using the line color:

- Black: p-value < 0.001
- Red: 0.001 < p-value < 0.01

- White: 0.01 < p-value < 0.05

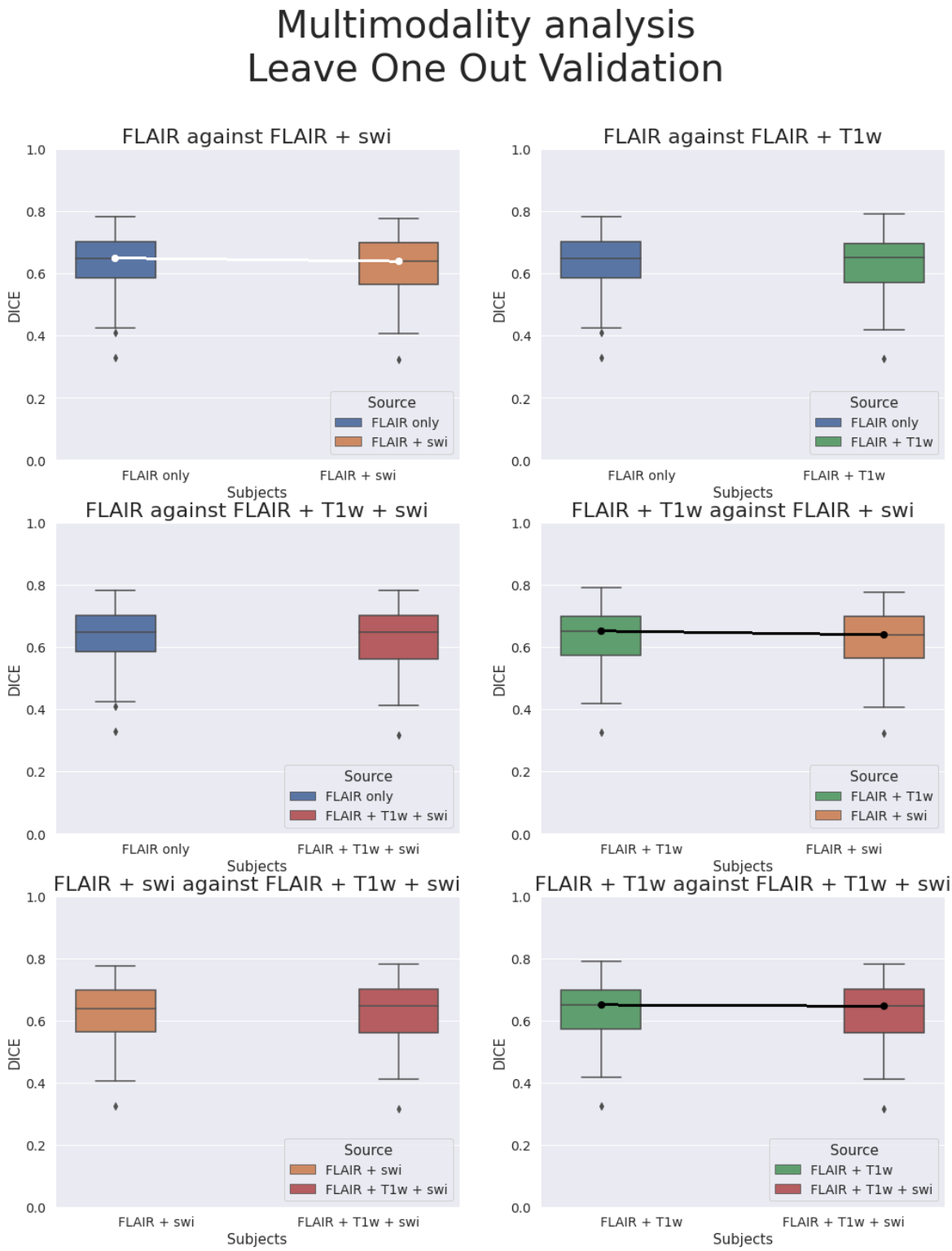


Figure 4.7. Pairwise comparisons of the distributions present in Fig. 4.6. Full lines indicate significant differences (p-value < 0.05) between distributions, calculated using a RANOVA test.

Results highlighted no significant difference between the following pairs:

- “FLAIR” against “FLAIR + T1w” (p-value = 0.121);

- “FLAIR” against “FLAIR + T1w + SWI” (p-value = 0.514);

- “FLAIR + SWI” against “FLAIR + T1w + SWI” (p-value = 0.150);

On the other hand, a significant difference was demonstrated for the following:

- “FLAIR” against “FLAIR + SWI” ($0.01 < p\text{-value} < 0.05$);

- “FLAIR + T1w” against “FLAIR + SWI” ($p\text{-value} < 0.001$);

- “FLAIR + T1w” against “FLAIR + T1w + SWI” ($p\text{-value} < 0.001$);

In particular, as for these latter, some results highlight the presence of a favorable trend (in terms of higher median values for the DICE) towards one modality with respect to the other. This is the case for “FLAIR” and “FLAIR + T1w”, when compared to “FLAIR + SWI”. Accordingly, higher median values are also found when comparing “FLAIR + T1w” with the “FLAIR + T1w + SWI” case.

In addition, it must be noted that the results from the LOO are comparable to the ones of BIANCA trained with 30 subjects, as the RANOVA test does not acknowledge any statistical difference.

4.1.4. Harmonization pipeline

Here we present the results obtained after training BIANCA with data from the Whitehall and UK Biobank populations. The existing training set, derived from the study of Bordin and colleagues(Bordin et al. 2021), involved 60 patients with “FLAIR + T1w” contrasts. Its aim was to maximize the generalization capabilities of the segmentation tool thus providing harmonization.

The trained BIANCA was thoroughly validated on different subjects, however that was limited to the same database. This thesis, for the first time, quantifies the performance of the trained BIANCA on subject from the different OASIS3 dataset, in view of testing the general validity or the previous training and its capability of harmonized processing on any database.

Once utilized for training, the testing phase was conducted on the entire pool of data at our disposition (40 subjects from the OASIS3 dataset) and results were evaluated by means of DICE using "expert" manual masks.

A comparison was finally drawn with results from the existing literature, in which the harmonization training set was both trained and tested on data from the Whitehall and UK Biobank populations in the previous study (Bordin et al. 2021). The main descriptive statistics of both cases are summarized in Table 4.1.

	Testing on OASIS3	Testing on Whitehall and UK Biobank
Mean	0.61	0.52 (for Whitehall Scanner1); 0.47 (for Whitehall Scanner2); 0.63 (for UK Biobank);
Median	0.63	0.54 (for Whitehall Scanner1); 0.47 (for Whitehall Scanner2); 0.65 (for UK Biobank);
Std	0.12	0.10 (for Whitehall Scanner1); 0.05 (for Whitehall Scanner2); 0.10 (for UK Biobank);

Table 4.1. Segmentation performance obtained testing with: i) data from the OASIS3; ii) data from the Whitehall and UK Biobank datasets. The training phase was conducted in both cases using the Whitehall + UK Biobank dataset from(Bordin et al. 2021).

The results obtained when applying the training set from (Bordin et al. 2021)to data from OASIS3 seem perfectly comparable in terms of mean, median and standard deviation values to that obtained applying it to data from Whitehall and UK Biobank.

Additionally, we report results obtained after comparing the mixed training strategy (i.e., training on Whitehall + UK Biobank and testing on OASIS3) with the single-site approach represented by the incremental training analysis (i.e., training and testing on OASIS3 using a 4-fold cross validation). The testing subset is represented, in both cases, by the pool of 40 manually labeled subjects from the OASIS3 dataset. "Expert" manual masks were considered for the evaluation and the resulting DICE are reported in Fig. 4.8.

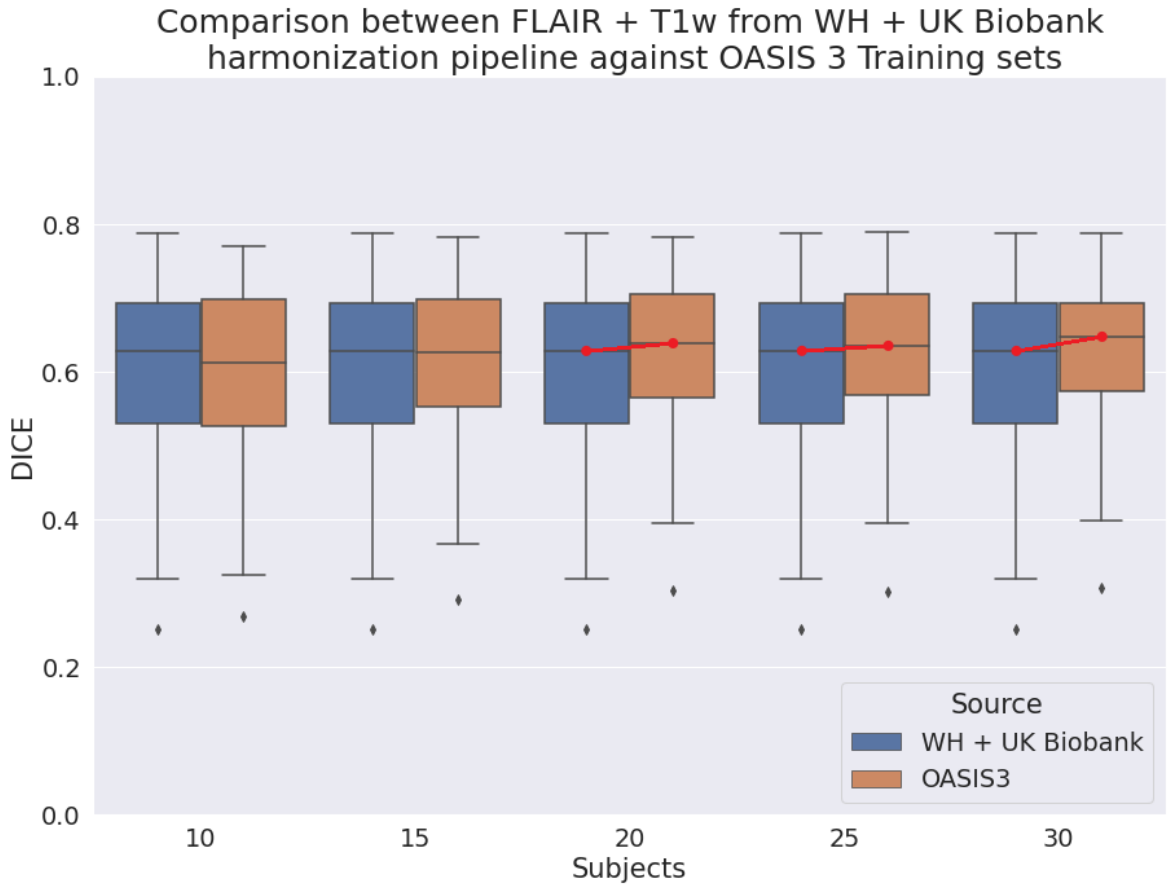


Figure 4.8. Boxplot of the DICE index (represented on the y axis) between BIANCA outputs obtained with both a mixed training strategy (training on Whitehall + UK Biobank and testing on OASIS3 – represented in blue) and a single-site training strategy (training and testing on OASIS3 using a 4-fold cross validation – represented in orange). Full lines indicate significant differences (p-value < 0.05) between adjacent distributions, calculated using a paired t-test.

Results highlight the presence of a difference between the mixed and the single-site training approaches only when the latter uses an amount of training subjects that is higher than 20. For those cases, results from the paired t-test gave indeed a p-value < 0.01. On the other hand, the remaining distributions have a p-values of 0.139 and 0.149, respectively.

4.2. Classification Model

In this section we present the results obtained from the classification models trying to predict CDR from both clinical data and WMH volumes. The ML techniques employed in the analysis are the ones described in Par. 3.2.2.3 – 3.2.2.5:

- SVM
- RFC with PCA
- RFC without PCA
- ANN

For each technique, the best parameters obtained after the tuning procedure will be presented, together with the evaluation metrics of Par. 2.2.2.6 (calculated on the *testing set*).

However, prior to these information, we present results for the analysis carried out on CDR before its pre-processing step.

4.2.1. Data pre-processing

In this paragraph we present the results obtained when comparing WMH volumes relative to different CDR groups. Our aim was to merge the CDR = 0.5 and CDR = 1 groups to binarize the target variable. But, before carrying on with this task, we plotted their WMH volumes distributions (using boxplots) along with the one from the CDR = 0 group, to check for the presence of potential differences. Results are reported in Fig. 4.9. A one-way ANOVA test was also run.

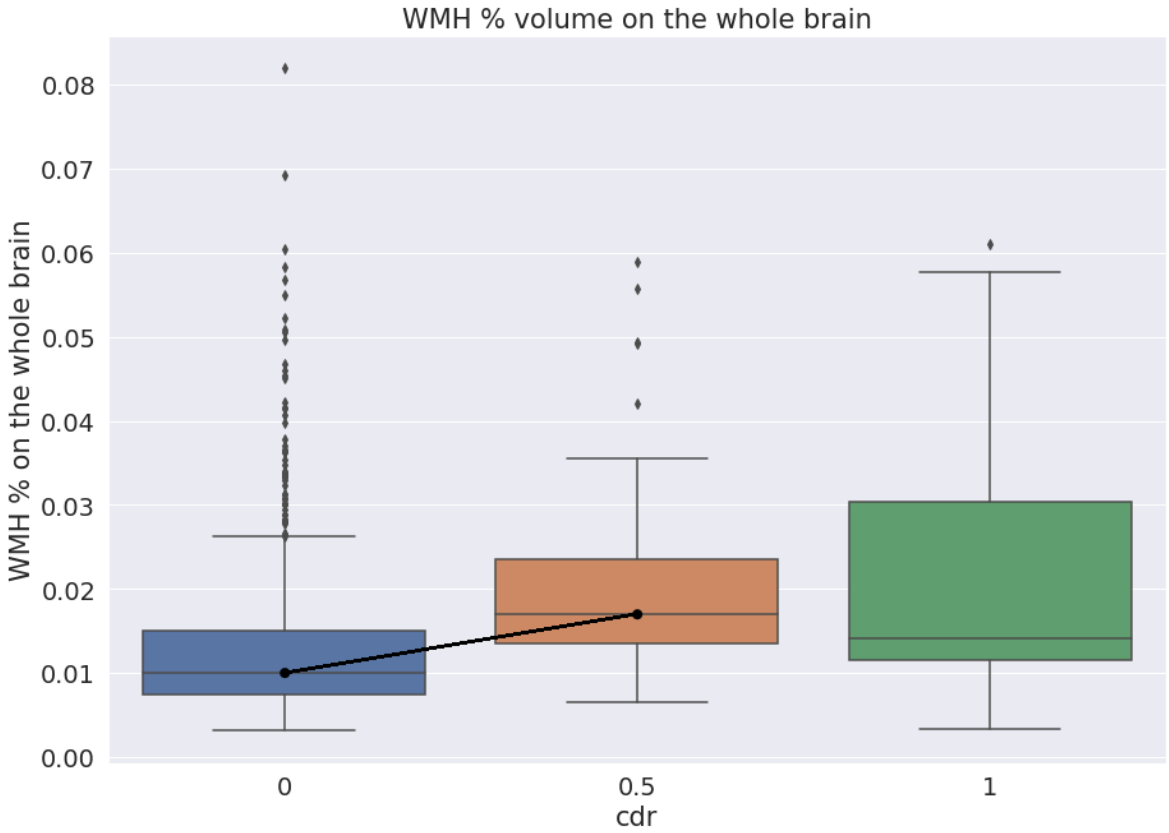


Figure 4.9. Boxplots of WMH percentage volumes (calculated as % of the total brain volume – represented on the y axis) for three different CDR groups: CDR = 0 (blue), CDR = 0.5 (orange), CDR = 1 (green). Full lines indicate significant differences (p-value < 0.05) between adjacent distributions, calculated using a one-way ANOVA test.

Results highlight the presence of a significative differences between distributions (p-value < 0.001). As for the post-hoc comparisons, a strong difference between the CDR = 0 group and the remaining ones was demonstrated (p-value < 0.001 in both cases). On the other hand, the ANOVA test provides a p-value of 0.99 when comparing the CDR = 0.5 and CDR = 1 groups. This result shows the insufficient statistical power of the current data set in order to separate different degrees of AD severity. Accordingly, the choice to limit to a binary classification at the present stage was justified.

At this point we present the results obtained after binarizing our target variable. The relationship between WMH volumes and age was compared for the couple of CDR groups obtained after merging: CDR = 0; CDR = 1 (obtained from all CDR>0 groups). Results are presented in Fig. 4.10 using a scatter plot. A linear regression with 95% confidence intervals is displayed.

Figure 4.10. Scatterplot between WMH volumes (calculated as % of the total brain volume – represented on the y axis) and age (represented on the x axis) for CDR = 0 (blue) and

CDR > 0 (orange – indicated as CDR = 1 in the legend) populations. Regression lines with 95% confidence interval are also displayed.

4.2.1.1. Support Vector Machine

As regards the implemented models, here we report results obtained using SVM. The training and testing procedures were conducted following the indications of Par. 3.2.2.2. Hyperparameter tuning was conducted using a Grid Search approach, while the optimal ratio between the target variable classes (CDR = 0 and CDR = 1) was identified through a trial-and-error approach.

As for the latter aspect, a 3:1 ratio between the CDR = 0 and CDR = 1 classes allows to obtain the best classification performance on *testing set*. Meanwhile, the optimal parameter values – identified on the *(real) training set* – are the following:

'C': 1; 'class_weight': 'balanced'; 'gamma': 0.1; 'kernel': 'rbf'

The quantitative evaluation metrics described in Par. 2.2.3 report the following values as for the classification performance:

- 80% accuracy
- 54% precision
- 85% recall
- 66% F1-score

Thanks to those results and its easily reading of the importance of the inputs variables, it will be the main focus for the next section, illustrating how each of those influence the Clinical dementia rating.

Firstly, as for all the other models, those are the outputs from the GridsearchCV, thus the best parameters found:

'C': 1, 'class_weight': 'balanced', 'gamma': 0.1, 'kernel': 'rbf'

This has been trained with a 3:1 ratio of negative and positive classes, thus implying that each model handles class balances differently.

A confusion matrix with the amount of TP, FP, TN and FN is also reported in Fig. 4.11.

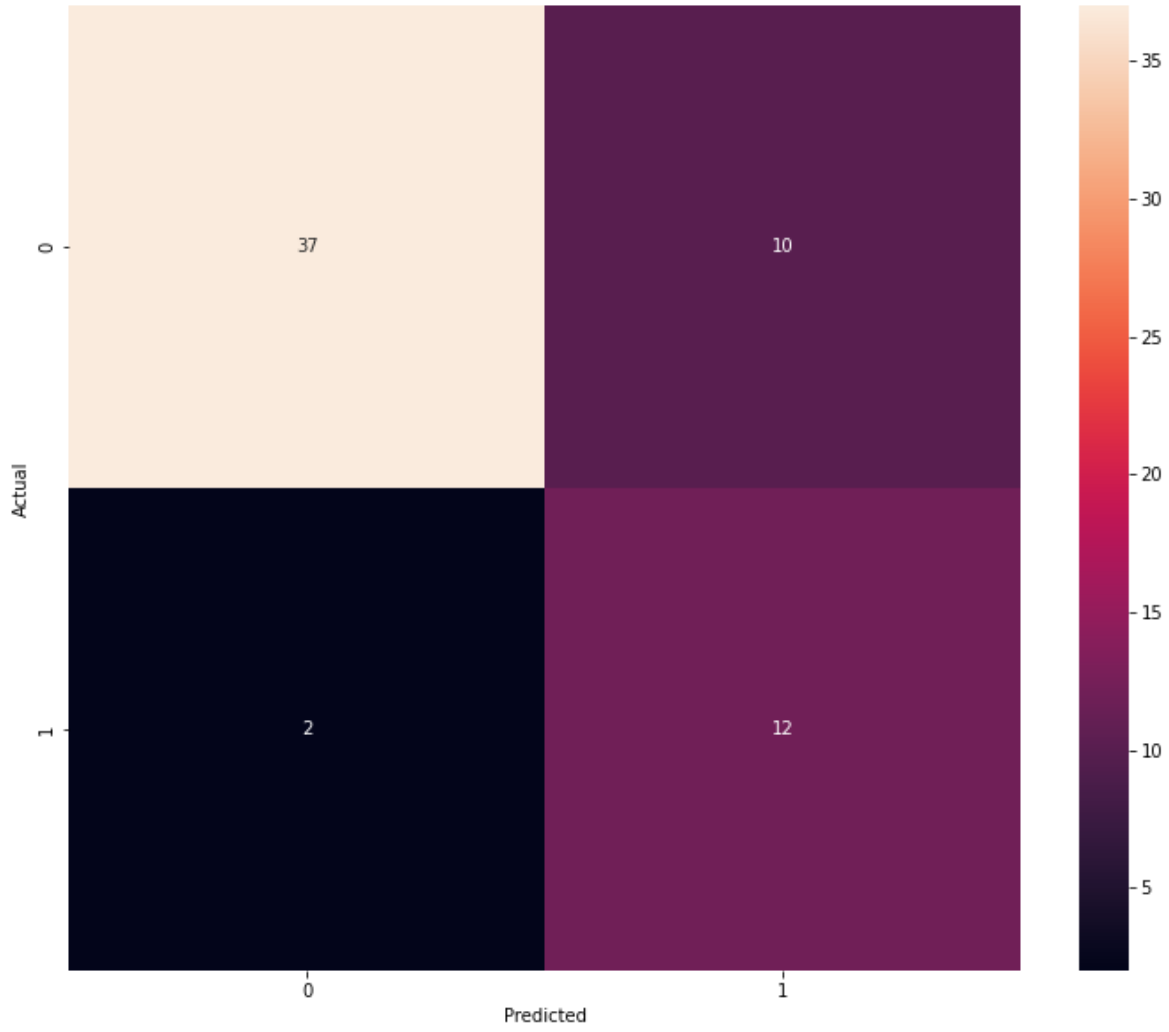


Figure 4.11. Confusion Matrix of the best SVM model.

Finally, the impact exerted on final classification by the different input variables, calculated using permutation importance, is reported in Fig 4.12.

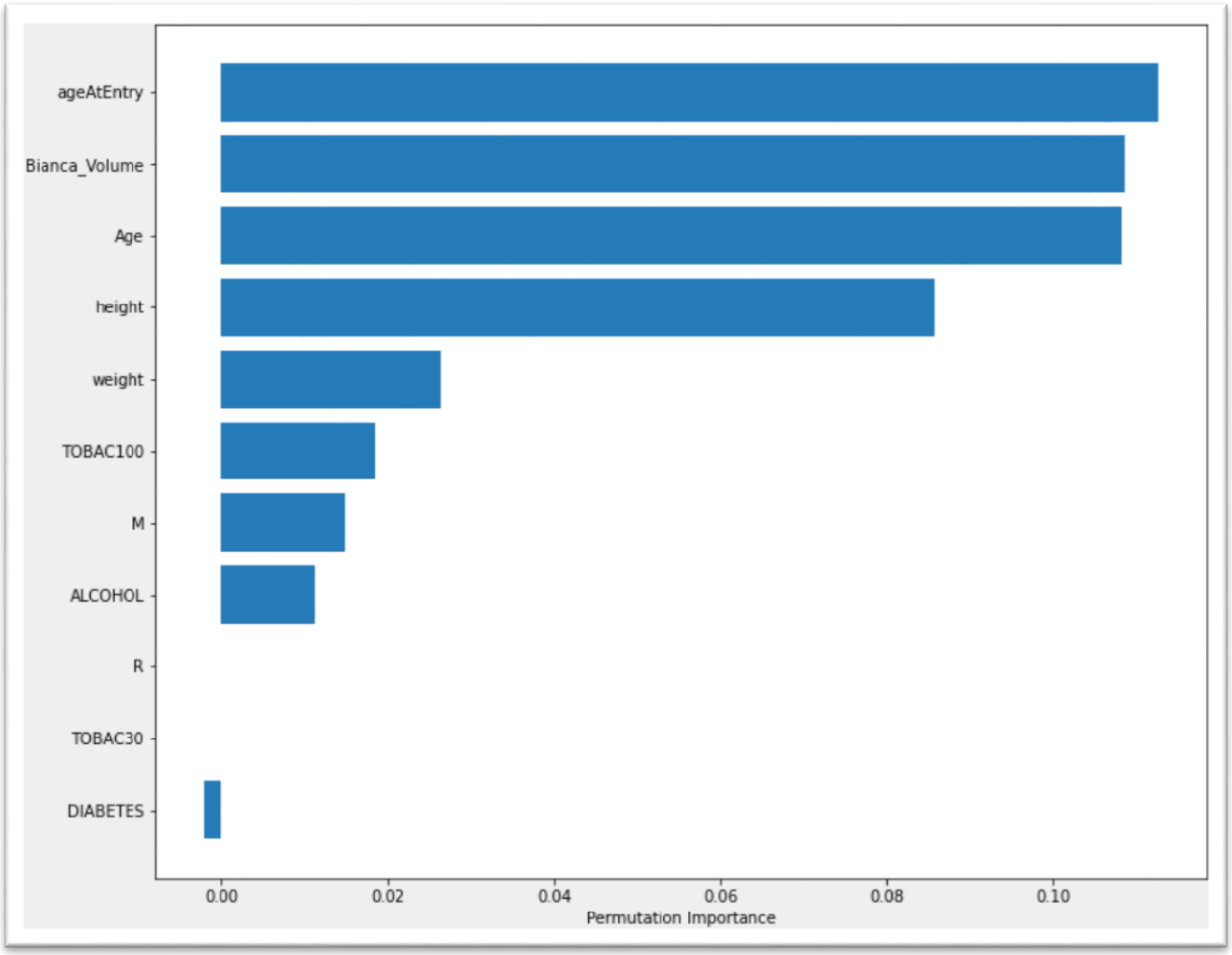


Figure 4.12. Permutation importance (reported on the x axis) of the different input variables (reported on the y axis) involved in the SVM model.

The plot reveals the presence of three variables with very high importance, followed by a fourth with slightly lower values. The remaining, on the other hand, seem scarcely relevant as for the CDR prediction. The first and most important variable is "ageAtEntry" representing the number of years passed since the subject was first included into the study. The second most important variable is instead "Bianca_Volume" representing the volume of WMH lesions segmented using BIANCA (expressed as % of the whole brain volume). This is followed by "Age" and eventually by "Height". Finally, a comment needs to be made about the least important variable, which is "DIABETES" (representing the presence or absence of such condition). Its permutation importance is negative. However, that is not a mistake. Negative values indeed indicate that the prediction obtained using shuffled (or noisy) versions of such variable is more accurate than the one carried out using the original data.

4.2.1.2. Random Forest Classifier with Principal Component Analysis

Here we present results obtained using RFC. The model was implemented following the procedure described in Par. 2.2.2.2. Hyperparameter were tuned and the optimal ratio between the target variable classes (CDR = 0 and CDR = 1) was identified.

To mitigate problems relative to the dataset splitting procedure carried out by the RFC algorithm, the model was first run using PCA. In this case, the best classification performance (evaluated on *testing set*) is obtained with a 1:1 ratio between the CDR = 0 and CDR = 1 classes. Moreover, results from the Grid Search indicated the following parameters as optimal:

'class_weight': 'balanced'; 'criterion': 'gini'; 'max_depth': 5; 'max_features': 'auto'; 'n_estimators': 200

Among them, the only questionable value is relative to the "class weight" parameter, which should be equal to "None" for a dataset with a 1:1 ratio between target classes. However, since "balanced" and "None" in this case would provide identical results, we decided to leave it unaltered.

Nonetheless, the following classification performance were reached:

- 82% accuracy
- 80% precision
- 80% recall
- 80% F1-score

A confusion matrix with is also reported in Fig. 4.13.

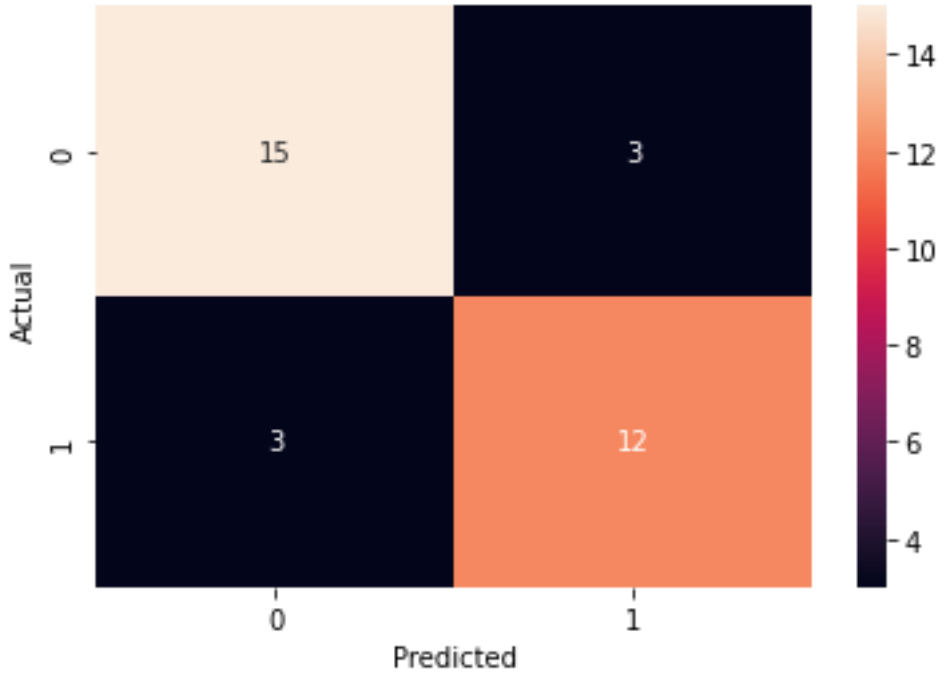


Figure 4.13. Confusion Matrix of the best RFC model with PCA.

As for the explainability, this model suffers limitations due to the introduction of PCA. The permutation importance (displayed in Fig 4.16) indicates some of the PCA components as being more relevant than other variables. But PCA components are themselves a linear combination of variables, and therefore it becomes hard to extract information. The composition of every component is presented in Fig. 4.14.

	PCA	Age	ageAtEntry	height	weight	Bianca_Volume
0	PCA 1	0.497554	0.353134	-0.462682	-0.502803	0.401063
1	PCA 2	0.439539	0.117242	0.529299	0.464820	0.544835
2	PCA 3	-0.125565	0.915600	0.118032	0.144220	-0.333435
3	PCA 4	-0.632147	0.133652	-0.375129	0.290419	0.597881
4	PCA 5	0.379301	-0.073241	-0.592549	0.652678	-0.271409

Figure 4.14. Linear combination coefficients for the different principal components. Each value has to be multiplied by its corresponding continuous variable (being either Age, ageAtEntry, height, weight or Bianca_Volume) and eventually sum up to the others to get the final result.

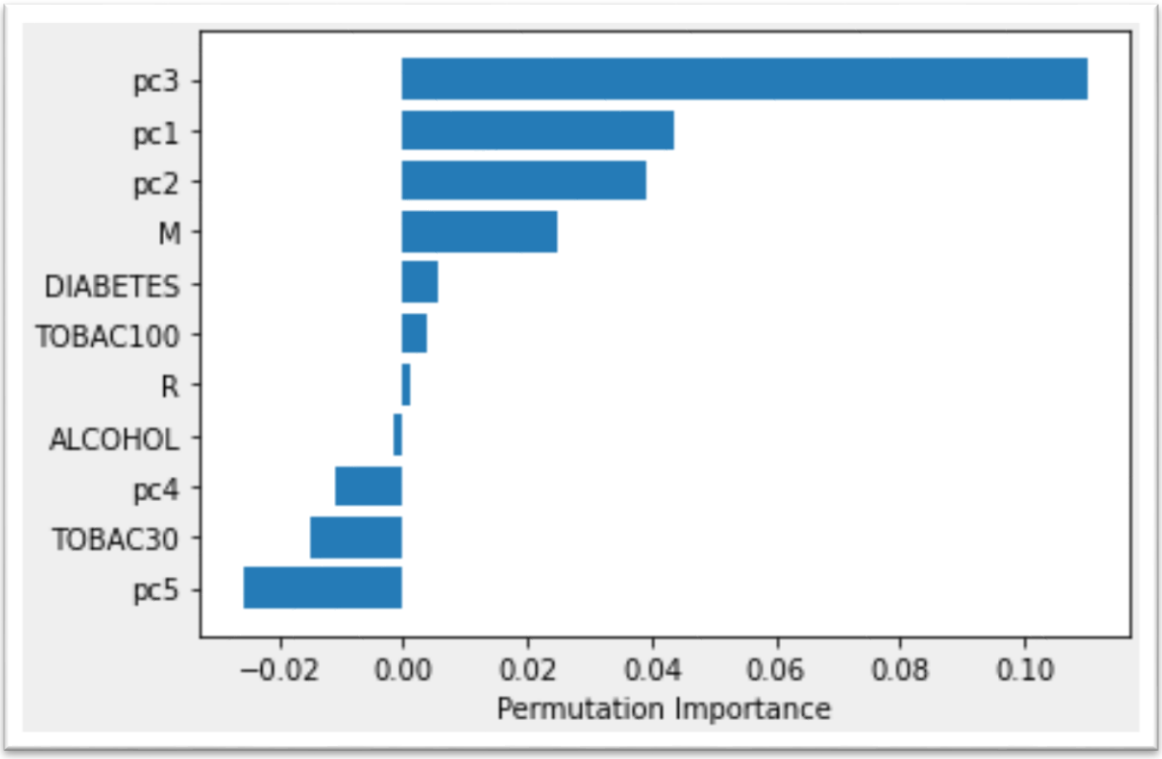


Figure 4.15. Permutation importance (reported on the x axis) of the different input variables (reported on the y axis) involved in the RFC model with prior PCA.

On the other hand, Fig. 4.15 shows that results are indeed difficult to interpret, requiring a constant check over Fig. 4.14 for every principal component. The most important variable is represented by PC3, which combines information on both the number of years for which the subject has been part of the study (ageAtEntry) and his WMH volume extracted using BIANCA (Bianca_Volume). The remaining 3 variables (Age, height and weight) on the other hand do contribute scarcely. The second most important variable is PC1 which combines almost equally the 5 continuous variables. PC2 follows immediately after combining them all except for the ageAtEntry.

4.2.1.3. Random Forest Classifier without Principal Component Analysis

Here we present results from the second RFC model, run without using PCA. A 1:1 ratio between the CDR = 0 and CDR = 1 classes allows to obtain the best classification performance on the *testing set.* Meanwhile, the optimal hyperparameters tuned on the *(real) training set* using a Grid Search approach, reveal to be the following:

'class_weight': 'balanced_subsample', 'criterion': 'entropy', 'max_depth': 4, 'max_features': 'auto', 'n_estimators': 50

The quantitative metrics used to evaluate classification give the following results:

- 76% accuracy
- 80% precision
- 60% recall
- 68% F1-score

The outcomes from the confusion matrix instead are represented in Fig. 4.16.

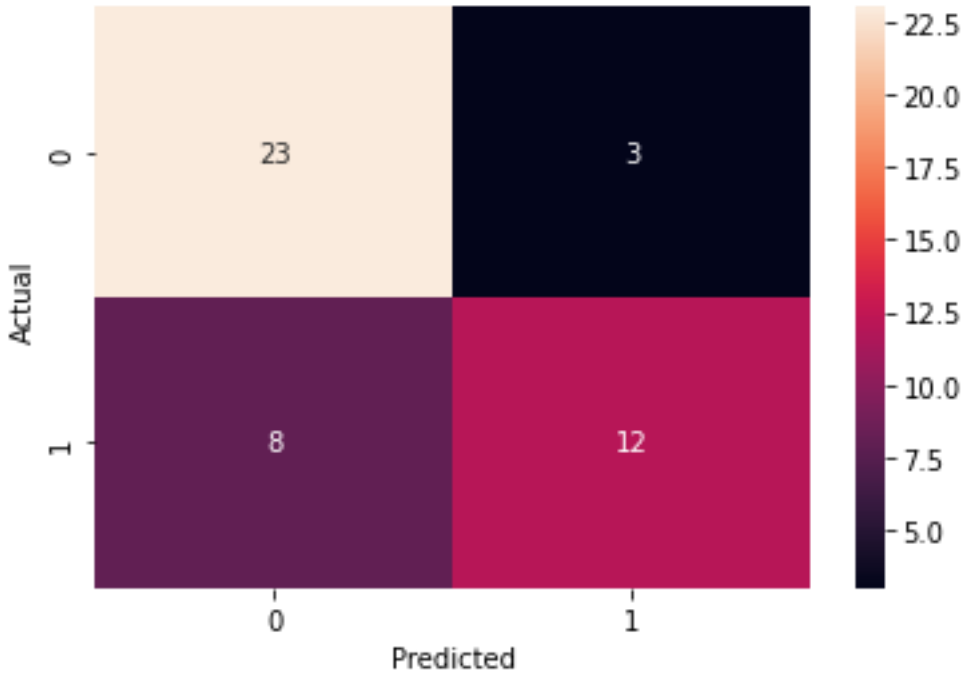


Figure 4.16. Confusion Matrix of the best RFC model without PCA.

Results from the permutation importance analysis are impacted by the working principle of this method. To explain how, in Fig. 4.17 we present results from two different RFC models, run with a different initial split of the dataset (carried out at the root node considering only one variable due to PCA absence). Despite achieving comparable performance (76% and 74% of accuracy) they show a very different permutation importance.

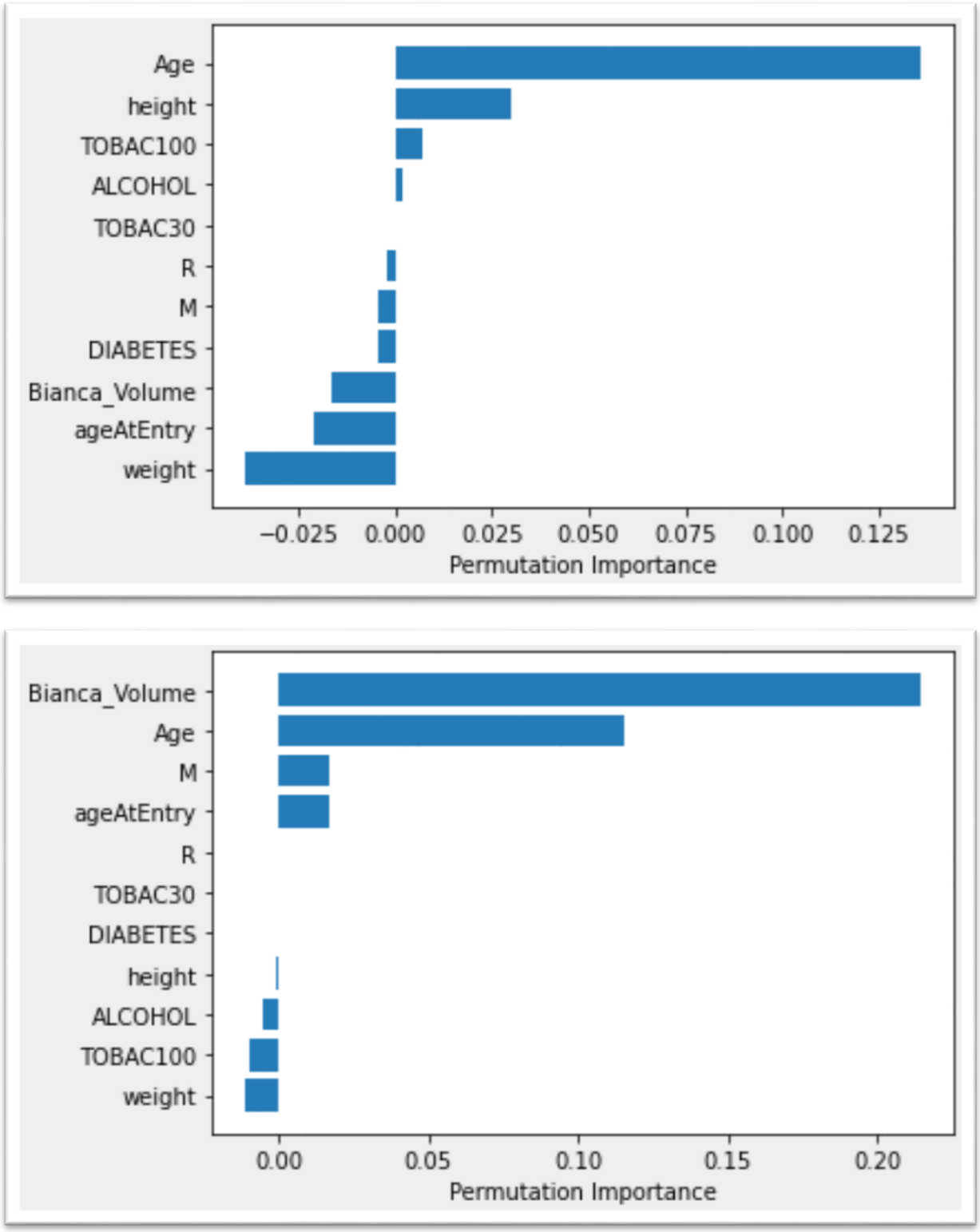


Figure 4.17. Permutation importance (reported on the x axis) of the different input variables (reported on the y axis) involved in two similarly performing RFC models. The upper and lower panel showed the model with 76% and 74% of accuracy, respectively.

In the first case "Bianca_Volume shows a negative and very low value, while in the second one the same variable appears among the most important features. A similar pattern is also seen for ageAtEntry. The permutation results are thus highly variable.

4.2.1.4. Artificial Neural Network

Finally, we present the results obtained from the implemented ANN model. The optimal classification (evaluated on *validation set*) is achieved through the following: i) a 2:1 ratio between the CDR = 0 and CDR = 1 classes; ii) the presence of class weight (the error relative to a mislabeling of the CDR > 0 class is "heavier" than its counterpart); iii) no bias initializer.

The resulting performance provides the following values:

- 84% accuracy
- 76% precision
- 81% recall
- 78% F1-score

The confusion matrix for the present method is also displayed in Fig. 4.18.

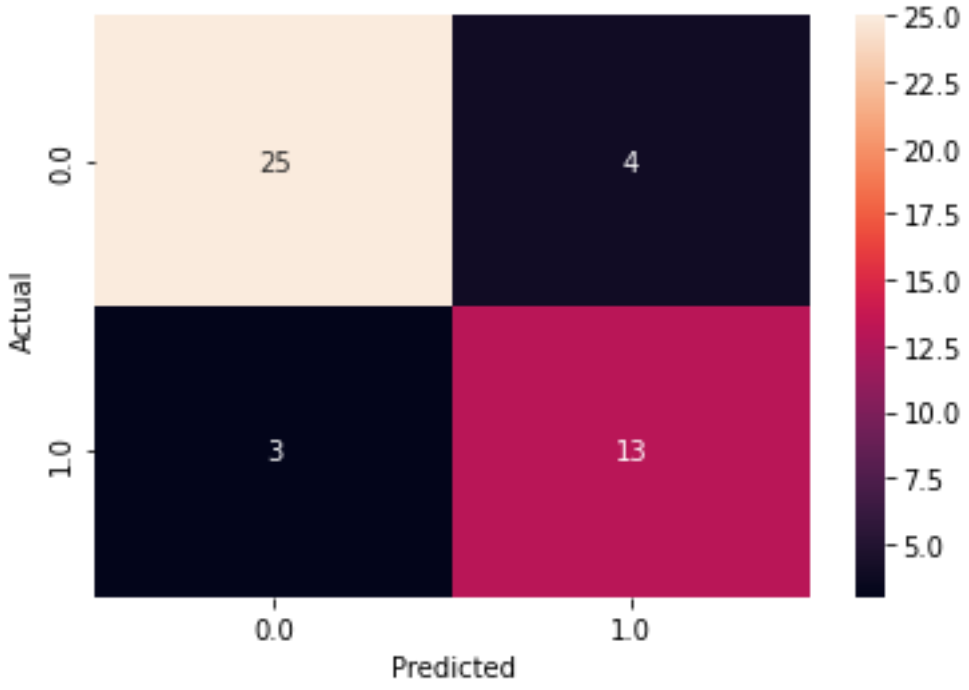


Figure 4.18. Confusion Matrix of the best ANN model.

5 Discussion and conclusions

In this chapter we discuss the results obtained throughout the analysis and carefully highlight the acquired information. The project focused on the optimization of automatic methods to segment WMHs. It also tried to confirm their role as neuroimaging biomarkers for the diagnosis of clinical dementia conditions associated to AD.

The segmentation tool under investigation was BIANCA, a fully automated approach developed by the by the Oxford Centre for Functional MRI of the Brain (FMRIB) based on the k-NN algorithm. BIANCA requires the choice of several parameters in order to carry out the WMH segmentation, some of which have a direct impact on the final performance. However, this relationship has not been fully investigated yet. First, no prior study has ever tried to determine the minimum number of subjects required in the training set to provide accurate segmentation outcomes. The work by Griffanti and colleagues evaluated this aspect only preliminarily primarily focusing its dependence from the WMH load of the involved participants(Griffanti et al. 2016). Secondly, a well-rounded view over multimodality is missing. Having a deeper level understanding of the impact exerted by single MRI contrasts or by specific combination of MRI contrasts over the final WMH segmentation would be highly valuable. This is even more true when it comes to harmonization contexts where performance is often hindered by the integration of heterogeneous data. The results here presented could indeed offer counterbalances by suggesting the best combination of modalities possible within the available datasets.

Alongside segmentation performance, harmonization of the WMH measures (i.e., the WMH volumes extracted using BIANCA) is another important aspect that this project tried to work on. The recent study from Bordin and colleagues (Bordin et al. 2021) has provided a harmonization strategy in the form of a training set derived from the Whitehall and UK Biobank populations. The result obtained on their testing set were overall satisfying. However, such training set was never validated on external datasets (with respect to Whitehall and UK Biobank) except for a minor project that counted only 10 subjects. A broader validation would thus prepare the

ground for its widespread applicability thus guaranteeing to the clinical community a proper tool for data integration between heterogeneous cohorts.

The first part of our analysis, tried therefore to evaluate BIANCA performance according to: i) an increasing number of training subjects; ii) the use of different MRI modalities; iii) the use of an external training set provided by the study of Bordin and colleagues. To do so, we extracted and manually labeled 40 subjects from the OASIS3 study, a longitudinal cohort of neuroimaging and clinical data focused on normal aging and AD subjects affected by cognitive decline. The manual labeling procedure was repeated twice (at the beginning ("preliminary") and half of the way ("expert") into the project, respectively) and the intra-rater variability was evaluated. Results from the first and last step were eventually compared to each other. This gave us the possibility to compare a mixed training strategy (represented by the external training set applied to our population) against a single-site training strategy (in which training and testing were both performed on the OASIS3).

Results of the different analysis steps, are summarised and discussed as follows:

- Involving an increasing number of subjects within the training process initially increased performance. However, after a certain threshold was reached, we encountered a plateau, and no further optimization was gained from the introduction of additional data. This analysis was carried out using every combination of MRI modalities explored during the second step and gave similar performance in every case. Likewise, the segmentation outcomes were evaluated using both "preliminary" and "expert" manual segmentations. Their use led us to the achievement of different threshold values – 25 and 20 – the first being higher than the second. Thus, despite the high value of intra-rater variability reach by the manual operator, this difference led us to believe that the "expert" segmentations were more accurate and therefore suitable to evaluate performance. This was further suggested by the visual inspection process carried out on both manual masks, that highlighted some missing areas around the periventricular region in the "non-expert" round. For this reason, the "expert" labels were used from that moment onwards to carry out evaluations.

- Using SWI as the only MRI modality to carry out the automatic segmentation process led to very poor performance (median DICE value close to zero).

In addition, several combinations including it displayed worse outcomes with respect to the counterpart that lacked SWI. In particular: FLAIR performed better than FLAIR + SWI (p-value < 0.05) and FLAIR + T1w performed better than FLAIR + T1w + SWI (p-value < 0.001). Moreover, even though FLAIR + T1w showed no difference with respect to FLAIR, when we added SWI to this, latter results were significantly worsened (p-value < 0.001 when comparing FLAIR + T1w with

FLAIR + SWI). This confirms that SWI alone carries no useful information as for the WMH segmentation. Instead, this contrast introduces noise that can significantly worsen BIANCA performance. As regards the remaining options, they obtained quite similar outcomes. Therefore, it was impossible to identify the optimal case. Nonetheless, since FLAIR and T1w are among the most common sequences performed in clinical contexts, we suggest using their combination whenever possible.

- The harmonization training set proposed by Bordin and colleagues (Bordin et al. 2021) provides good segmentation outcomes even when applied to an entirely new population with respect to the ones involved in its development (Whitehall and UK Biobank). The performance obtained in terms of DICE had indeed a mean, median and standard deviation values comparable with the ones presented in the corresponding literature. This, together with the numerosity of the pool of subjects used to conduct the analysis (40), proved the achievement of a more robust validation, previously conducted on a narrow dataset of only 10 subjects. As for the comparison of the obtained results with that of the incremental training set analysis, we found that the single-site training (usually considered as the optimal strategy) was superior with respect to the mixed approach (training on Whitehall + UK Biobank and testing on OASIS3) only if a minimum number of subjects were involved within the analysis. This value was equal to 20, which indeed corresponded to the threshold for performance optimization found during the incremental training analysis. Therefore, we can conclude that, when manual masks from an expert rater are used, a minimum number of 20 subjects is generally sufficient to obtain good segmentation outcomes. At the same time, 20 is also the minimum number of subjects needed to guarantee the superiority of the single-site training with respect to the mixed approach.

After we dealt with segmentation performance using the imaging part of the OASIS3 dataset, we extended the analysis to the clinical variables. Our aim was to build a ML model merging both type of data (i.e., imaging and clinical) to predict a well-known clinical score for dementia: the CDR. To do that, we included in the study 471 subjects, among which there were the 40 manually labeled used for the former evaluations. In addition, we selected the following 11 clinical variables: Age, ageAtEntry, height, weight, DIABETES, ALCOHOL, TOBAC30, TOBAC100, sex, Hand, CDR. Finally, as for the imaging variable, we calculated the WMH volume using BIANCA and expressed its value as percentage of the total brain volume.

It must be clarified that the ultimate goal of this analysis was not the classification task by itself but instead to evaluate and quantify the impact exerted by WMHs on the final prediction of AD. This, in an attempt to validate their role as neuroimaging biomarker for the early-stage diagnosis of AD dementia. Different ML models were

used to reach our goal, among which: SVM, RFC and ANN. Results obtained from each one, are summarised and discussed as follows:

- The SVM model reached a lower performance with respect to the others, especially when it comes for precision (80% accuracy, 54% precision, 85% recall, 66% F1-score). However, it had the property of being simple and easy to discuss as for the input variables relevance. In this regard, the analysis identified three variables with almost identical importance, followed by a fourth which displayed slightly lower values. The first was "ageAtEntry", indicating for how many years a subject have been part of the study. As the OASIS3 collects MR Sessions and clinical visits from studies mainly focused on AD, it is possible that many patients had their initial visit at the first appearance of symptoms. This implies that the longer a patient does stay in the study, the worse his disease can get. For this reason, the relationship between "ageAtEntry" and CDR seemed reasonable. The second most important variable was instead represented by the WMH volume extracted using BIANCA. Its presence among the features that better explain the classification outcome confirms our hypothesis about the great importance of this neuroimaging sign in the context of CDR prediction. Afterwards, the third position was occupied by age, which is already well-known to be strictly correlated with the development of AD dementia (Guerreiro and Bras 2015). Height had also a significant role. Unfortunately, the metric we used to evaluate features importance did not specify whether a variable was influencing results in a positive or negative way. Thus, we could not be certain if taller patients were more or less subject to AD dementia. Nonetheless, we hypothesize the nature of that relationship as being inverse. Indeed, WMHs have been correlated with high blood pressure values (Dufouil et al. 1991; Maillard et al. 2012) and these latter have proved to be in an inverse relationship with height (Cochran et al. 2021). This may suggest height is inversely correlated with the presence of WMHs, which in turn are correlated to CDR. The reason behind the high importance exerted by this variable would thus the explained. However, since these finding are extremely early, further research is mandatory to evaluate and possibly confirm this speculation.

- RFC was evaluated both using PCA and not. With PCA it provided good classification outcomes (82% accuracy and 80% for precision, recall and F1-score). Results were higher with respect to the SVM case. However, differently from SVM, this model suffered from the lack of explainability introduced by PCA as regards the impact exerted by input variables on the final classification. Indeed, among the most important variables we found PCA components, that were a linear combination of the 5 continuous variables present within our study (Age, ageAtEntry, Bianca_Volume, height, weight). Hence, extracting information became hard. The variable with highest relevance, PC3, was a combination in which

age at entry and BIANCA volume (whose meaning has already been discussed) played the most important role. The remaining 3 variables, on the other hand, gave a scarce contribution. This confirmed the results highlighted by the SVM model. However, the presence of other principal components among the most important variables made it very difficult to draw further conclusions.

RFC without PCA, provided the lowest performance. This was probably due the algorithm working principle that proceeds by splitting the datasets at each node using only one variable at a time. This strategy additionally affected our chances to have a good indicator of permutation importance, since the obtained result did vary a lot depending on the initial split of the dataset. Indeed, when we compared two RFC models built with a different split, we obtained opposite results in terms not only of bianca volumes and ageAtEntry, but also of TOBAC30 (which indicates if the subject has smoked in the last 30 days) and ALCOHOL (which indicates if the patients drinks alcohol). Their importance was extremely high in one case and very low in the other. Therefore, we conclude that RFC without PCA is unreliable to draw conclusions.

- The ANN achieved the best classification results (84% accuracy, 76% precision, 81% recall, 78% F1-score). However, as already explained in the previous chapters, this algorithm had a very low explainability as for the input variables relevance. Therefore, results were only kept as a gold standard to compare the performance of other classifiers.

To conclude, when we consider the tradeoff between performance and explainability the SVM model was the ones providing the best solution. His results indicated the WMH volume among the most relevant features as for the classification of CDR, thus confirming their importance as neuroimaging hallmarks for AD dementia.

5.1. Conclusion

With this work we gained important insights to help optimizing the automatic segmentation of WMHs. We acknowledged how BIANCA performance varies when using different: i) number of training subjects; ii) MRI contrasts; iii) training strategies (i.e., single-site or mixed). We discovered the presence of a plateau in the performance growth experienced when adding more subjects to the training. We identified 20 as the minimum value to get optimal results. In addition, we

highlighted the total lack of information carried by the SWI contrast and recognized its presence as potential source of noise for the segmentation. Finally, we found that the well-known superiority of the single-site training with respect to the mixed, is held only if a minimum number of subjects are used for training. This was again represented by 20. Furthermore, the results obtained when applying an external training set – developed with harmonization purposes – to our population gave performance comparable with that of the corresponding literature. This proved its widespread applicability. Finally, the ML models we built, successfully confirmed the importance of WMHs in the assessment of AD Clinical Dementia.

With these findings we have strengthen our former knowledge on the automatic segmentation strategy represented by BIANCA. In addition, we have validated a harmonization pipeline to derive integrated measures of WMH volumes. Eventually, we have demonstrated the role of WMHs as early-stage hallmark of AD dementia.

Our experiments were partially limited by the amount of data we could access. Future developments may address the use of biggest population focused on AD such as ADNI (https://adni.loni.usc.edu/). By providing numerous records and additional clinical variables with respect to our study, they could allow us to explore more robust classification techniques. A possibility could be represented by deep learning strategies, that combined with more advanced explainability techniques as for the role of WMHs, could help us deepening our knowledge about this neurological sign.

Bibliogaphy

Altmann-Schneider, Irmhild et al. 2013. "Lower Susceptibility to Cerebral Small Vessel Disease in Human Familial Longevity: The Leiden Longevity Study." *Stroke* 44(1): 9–14.

Balakrishnan, Ramya, Maria del C. Valdés Hernández, and Andrew J. Farrall. 2021. "Automatic Segmentation of White Matter Hyperintensities from Brain Magnetic Resonance Images in the Era of Deep Learning and Big Data – A Systematic Review." *Computerized Medical Imaging and Graphics* 88: 101867.

Bastin, Mark E. et al. 2009. "Diffusion Tensor and Magnetization Transfer MRI Measurements of Periventricular White Matter Hyperintensities in Old Age." *Neurobiology of Aging* 30(1): 125–36.

Black, Sandra, Fuqiang Gao, and Juan Bilbao. 2009. "Understanding White Matter Disease Imaging-Pathological Correlations in Vascular Cognitive Impairment Measuring Small Vessel Disease in Vivo: Challenges and Opportunities." http://stroke.ahajournals.org.

Bordin, Valentina et al. 2021. "Integrating Large-Scale Neuroimaging Research Datasets: Harmonisation of White Matter Hyperintensity Measurements across Whitehall and UK Biobank Datasets." *NeuroImage* 237.

Cochran, John Michael et al. 2021. "The Relationship between Adult Height and Blood Pressure." *Cardiology* 146(3): 345–50.

Cover, T., and P. Hart. 1967. "Nearest Neighbor Pattern Classification." *IEEE Transactions on Information Theory* 13(1): 21–27.

Debette, Stéphanie, and H. S. Markus. 2010. "The Clinical Importance of White Matter Hyperintensities on Brain Magnetic Resonance Imaging: Systematic Review and Meta-Analysis." *BMJ (Online)* 341(7767): 288.

Dong, Chao, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2016. "Image Super-Resolution Using Deep Convolutional Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(2): 295–307.

Dufouil, C et al. 1991. *Longitudinal Study of Blood Pressure and White Matter Hyperintensities The EVA MRI Cohort*.

Evgeniou, Theodoros, and Massimiliano Pontil. 2001. "Support Vector Machines: Theory and Applications." In , 249–57.

Farrall, Andrew J., and Joanna M. Wardlaw. 2009. "Blood–Brain Barrier: Ageing and Microvascular Disease – Systematic Review and Meta-Analysis." *Neurobiology of Aging* 30(3): 337–52.

Fazekas, F. et al. 1993a. "Pathologic Correlates of Incidental Mri White Matter Signal Hyperintensities." *Neurology* 43(9): 1683–89.

———. 1993b. "Pathologic Correlates of Incidental Mri White Matter Signal Hyperintensities." *Neurology* 43(9): 1683–89.

Fazekas, Franz et al. 2005. "MTI of White Matter Hyperintensities." *Brain* 128(12): 2926–32.

Ferguson, Stewart C et al. *Cognitive Ability and Brain Structure in Type 1 Diabetes Relation to Microangiopathy and Preceding Severe Hypoglycemia*.

Gons, Rob A R et al. "Cigarette Smoking Is Associated with Reduced Microstructural Integrity of Cerebral White Matter." *A JOURNAL OF NEUROLOGY*. http://sourceforge.net/.

Gouw, Alida A. et al. 2011. "Heterogeneity of Small Vessel Disease: A Systematic Review of MRI and Histopathology Correlations." *Journal of Neurology, Neurosurgery and Psychiatry* 82(2): 126–35.

Greve, Douglas N., and Bruce Fischl. 2009. "Accurate and Robust Brain Image Alignment Using Boundary-Based Registration." *NeuroImage* 48(1): 63–72.

Griffanti, Ludovica et al. 2016. "BIANCA (Brain Intensity AbNormality Classification Algorithm): A New Tool for Automated Segmentation of White Matter Hyperintensities." *NeuroImage* 141: 191–205.

Grossi, Enzo, and Massimo Buscema. 2007. "Introduction to Artificial Neural Networks." *European Journal of Gastroenterology & Hepatology* 19(12): 1046–54.

Guerreiro, Rita, and Jose Bras. 2015. "The Age Factor in Alzheimer's Disease." *Genome Medicine* 7(1): 106.

Hachinski, Vladimir C, Paul Potter, and Harold Merskey. *Leuko-Araiosis*. http://archneur.jamanetwork.com/.

Jenkinson, M. 2002. "Improved Optimization for the Robust and Accurate Linear Registration and Motion Correction of Brain Images." *NeuroImage* 17(2): 825–41.

Jenkinson, Mark et al. 2012. "FSL." *NeuroImage* 62(2): 782–90.

Jenkinson, Mark, and Stephen Smith. 2001. "A Global Optimisation Method for Robust Affine Registration of Brain Images." *Medical Image Analysis* 5(2): 143–56.

Katuwal, Rakesh, P. N. Suganthan, and Le Zhang. 2020. "Heterogeneous Oblique Random Forest." *Pattern Recognition* 99: 107078.

LaMontagne, Pamela J et al. 2019. "OASIS-3: Longitudinal Neuroimaging, Clinical, and Cognitive Dataset for Normal Aging and Alzheimer Disease." *medRxiv*: 2019.12.13.19014902. http://medrxiv.org/content/early/2019/12/15/2019.12.13.19014902.abstract.

Lepetit, V., and P. Fua. 2006. "Keypoint Recognition Using Randomized Trees." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(9): 1465–79.

Maillard, Pauline et al. 2012. "Effects of Systolic Blood Pressure on White-Matter Integrity in Young Adults in the Framingham Heart Study: A Cross-Sectional Study." *The Lancet Neurology* 11(12): 1039–47.

Maniega, Susana Muñoz et al. 2015. "White Matter Hyperintensities and Normal-Appearing White Matter Integrity in the Aging Brain." *Neurobiology of Aging* 36(2): 909–18.

Markus, H S. 2000. "Reduced Cerebral Blood Flow in White Matter in Ischaemic Leukoaraiosis Demonstrated Using Quantitative Exogenous Contrast Based Perfusion MRI." *Journal of Neurology, Neurosurgery & Psychiatry* 69(1): 48–53.

McCarthy, Paul. 2021. "FSLeyes." https://zenodo.org/record/5576035 (March 7, 2022).

Moran, Chris, Thanh G. Phan, and Velandai K. Srikanth. 2012. "Cerebral Small Vessel Disease: A Review of Clinical, Radiological, and Histopathological Phenotypes." *International Journal of Stroke* 7(1): 36–46.

Morris, Zoe et al. 2009. "Incidental Findings on Brain Magnetic Resonance Imaging: Systematic Review and Meta-Analysis." *BMJ (Online)* 339(7720): 547–50.

Munoz, David G. 2003. "Small Vessel Disease: Neuropathology." *International Psychogeriatrics* 15(S1): 67–69.

O'Sullivan, M. et al. 2002. "Patterns of Cerebral Blood Flow Reduction in Patients with Ischemic Leukoaraiosis." *Neurology* 59(3): 321–26.

O'Sullivan, M. 2004. "Diffusion Tensor MRI Correlates with Executive Dysfunction in Patients with Ischaemic Leukoaraiosis." *Journal of Neurology, Neurosurgery & Psychiatry* 75(3): 441–47.

Parker Jones, O, F Alfaro-Almagro, and S Jbabdi. 2018. "An Empirical, 21st Century Evaluation of Phrenology." *Cortex; a journal devoted to the study of the nervous system and behavior* 106: 26–35. https://pubmed.ncbi.nlm.nih.gov/29864593.

Qu, Yongming et al. 2002. *Principal Component Analysis for Dimension Reduction in Massive Distributed Data Sets SDS: A Framework for Scientific Data Services View Project Understanding Climate Change: A Data Driven Approach View Project Principal Component Analysis for Dimension Reduction in Massive Distributed Data Sets* *. https://www.researchgate.net/publication/232063041.

Shin, Hoo-Chang et al. 2016. "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning." *IEEE Transactions on Medical Imaging* 35(5): 1285–98.

Shoamanesh, A., C. S. Kwok, and O. Benavente. 2011. "Cerebral Microbleeds: Histopathological Correlation of Neuroimaging." *Cerebrovascular Diseases* 32(6): 528–34.

Smith, Stephen M. 2002. "Fast Robust Automated Brain Extraction." *Human Brain Mapping* 17(3): 143–55.

———. 2004. "Advances in Functional and Structural MR Image Analysis and Implementation as FSL." *NeuroImage* 23: S208–19.

Staals, Julie et al. 2014. *Stroke Subtype, Vascular Risk Factors, and Total MRI Brain Small-Vessel Disease Burden*. www.equatornetwork.org.

Tomimoto, Hidekazu et al. 1996. "Alterations of the Blood-Brain Barrier and Glial Cells in White-Matter Lesions in Cerebrovascular and Alzheimer's Disease Patients." *Stroke* 27(11): 2069–74.

Turner, Stephen T. et al. 2004. "Heritability of Leukoaraiosis in Hypertensive Sibships." *Hypertension* 43(2): 483–87.

Valdés Hernández, Maria del C. et al. 2013. "Brain White Matter Damage in Aging and Cognitive Ability in Youth and Older Age." *Neurobiology of Aging* 34(12): 2740–47.

Wardlaw, Joanna M. et al. 2013. "Neuroimaging Standards for Research into Small Vessel Disease and Its Contribution to Ageing and Neurodegeneration." *The Lancet Neurology* 12(8): 822–38.

Wardlaw, Joanna M., Colin Smith, and Martin Dichgans. 2013. "Mechanisms of Sporadic Cerebral Small Vessel Disease: Insights from Neuroimaging." *The Lancet Neurology* 12(5): 483–97.

Wardlaw, Joanna M., Maria C. Valdés Hernández, and Susana Muñoz-Maniega. 2015. "What Are White Matter Hyperintensities Made of? Relevance to Vascular Cognitive Impairment." *Journal of the American Heart Association* 4(6): 001140.

Woolrich, Mark W. et al. 2009. "Bayesian Analysis of Neuroimaging Data in FSL." *NeuroImage* 45(1): S173–86.

Yano, Niai, and Hiroshi Watanabe. 2020. "Image Quality Enhancement with Machine Learning Based Multi-Step Super-Resolution." In *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, IEEE, 141–45.

Zhang, Y., M. Brady, and S. Smith. 2001. "Segmentation of Brain MR Images through a Hidden Markov Random Field Model and the Expectation-Maximization Algorithm." *IEEE Transactions on Medical Imaging* 20(1): 45–57.

A Appendix A

5.2. Spreadsheet filtering

```
#!/usr/bin/python
import pandas as pd
name = “” #insert the csv file name here
path = “” #insert the path to the csv file here
df = pd.read_csv(path+name, header = 0)
index =[]
for i in range(len(df)):
    string = str(df["Scans"][i])
    if "swi" in string:
        if "FLAIR" in string:
            if "T1w" in string:
                if "GRE" in string:
                    index.append(i)
                else:
                    pass

new_df = []
new_df = df.iloc[index]
new_df = df.loc[:, new_df.columns = "MR ID"]
new_df.to_csv(path+’FLAIR_T1w_SWI.csv')
```

5.3. Renaming external

```
#!/bin/bash
path="" #insert the path to the OASIS3 downloaded folders file here
cd path

for i in {00..39}
do
files=(*)
mv ${files[0]} sub${i}
done

for i in *
do
for j in $i/*
do
mv $j/*FLAIR.nii.gz $i/${i}_FLAIR.nii.gz
mv $j/*FLAIR_roi.nii.gz $i/${i}_FLAIR_roi1.nii.gz
mv $j/*FLAIR_roi.nii.gz $i/${i}_FLAIR_roi2.nii.gz
mv $j/*T1w.nii.gz $i/${i}_T1w.nii.gz
mv $j/*swi.nii.gz $i/${i}_swi.nii.gz

done

mkdir $i/FLAIR
mkdir $i/T1w
mkdir $i/SWI
```

```
mv $i/*FLAIR.nii.gz $i/FLAIR/
mv $i/*FLAIR_roi1.nii.gz $i/FLAIR/
mv $i/*FLAIR_roi2.nii.gz $i/FLAIR/
mv $i/*T1w.nii.gz $i/T1w/
mv $i/*swi.nii.gz $i/SWI/


for k in {1..5}
do
rm -rf anat$k
rm -rf swi$k
done

done
```

5.4. BET run

```
#!/bin/bash
for i in {01..40}
```

```
do

path="" #insert the path to the OASIS3 downloaded folders file here

(bet ${path}sub${i}/FLAIR/sub${i}_FLAIR.nii.gz
${path}sub${i}/FLAIR/sub${i}_FLAIR_brain.nii.gz -R)

(bet ${path}sub${i}/T1w/sub${i}_T1w.nii.gz
${path}sub${i}/T1w/sub${i}_T1w_brain.nii.gz -R)

(bet ${path}sub${i}/SWI/sub${i}_swi.nii.gz
${path}sub${i}/SWI/sub${i}_swi_brain.nii.gz -R)

(bet ${path}sub${i}/SWI-MAg/sub${i}_swi_mag.nii.gz
${path}sub${i}/SWI/sub${i}_swi_mag_brain.nii.gz -R)

done
```

5.5. Fast run

```
#!/bin/bash

for i in {01..40}

do

path="" #insert the path to the OASIS3 downloaded folders file here

(/usr/local/fsl/bin/fast -B ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain)

(/usr/local/fsl/bin/fast -B ${path}sub${i}/T1w/sub${i}_T1w_brain)

(/usr/local/fsl/bin/fast -B ${path}sub${i}/SWI/sub${i}_swi_brain)

(/usr/local/fsl/bin/fast -B ${path}sub${i}/SWI_MAG/sub${i}_swi_mag_brain)


done
```

5.6. FLIRT run

```
#!/bin/bash

for i in {01..40}

do

path="" #insert the path to the OASIS3 downloaded folders file here

(/usr/local/fsl/bin/flirt -in ${path}sub${i}/SWI/sub${i}_swi_brain_restore.nii -ref ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain_restore.nii.gz -out ${path}sub${i}/SWI/sub${i}_swi_reg.nii.gz -omat ${path}sub${i}/SWI/sub${i}_swi_reg.mat -dof 6)

(/usr/local/fsl/bin/flirt -in ${path}sub${i}/SWI_MAG/sub${i}_swi_mag_brain_restore.nii -ref ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain_restore.nii.gz -out ${path}sub${i}/SWI_MAG/sub${i}_swi_mag_reg.nii.gz -omat ${path}sub${i}/SWI_MAG/sub${i}_swi_mag_reg.mat -dof 6)

(/usr/local/fsl/bin/flirt -in ${path}sub${i}/T1w/sub${i}_T1w_brain_restore.nii -ref ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain_restore.nii.gz -out ${path}sub${i}/T1w/sub${i}_T1w_reg.nii.gz -omat ${path}sub${i}/T1w/sub${i}_T1w_reg.mat -dof 6)


done
```

5.7. BIANCA masking

```
#!/bin/bash

for i in {01..40}

do

path="" #insert the path to the OASIS3 downloaded folders file here

mkdir ${path}sub${i}/T1w/fsl_anat

cp ${path}sub${i}/T1w/sub${i}_T1w.nii.gz ${path}sub${i}/T1w/fsl_anat

fsl_anat -i ${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.nii.gz

(make_bianca_mask ${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_biascorr
${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_fast_pve_0
${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/MNI_to_T1_nonlin_field.nii.gz)

(/usr/local/fsl/bin/flirt -in
${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_biascorr_bianca_mask.nii.gz -ref
${path}sub${i}/FLAIR/sub${i}_FLAIR_brain_restore.nii.gz -out
${path}sub${i}/T1w/sub${i}_T1w_bianca_mask_reg.nii -applyxfm -init
${path}sub${i}/T1w/sub${i}_T1w_reg.mat)

(fslmaths ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain_restore.nii.gz -mas
${path}sub${i}/T1w/sub${i}_T1w_bianca_mask_reg.nii
${path}sub${i}/FLAIR/sub${i}_FLAIR_masked.nii.gz)

(fslmaths ${path}sub${i}/SWI_MAG/sub${i}_swi_mag_reg.nii.gz -mas
${path}sub${i}/T1w/sub${i}_T1w_bianca_mask_reg.nii
${path}sub${i}/SWI_MAG/sub${i}_swi_mag_masked.nii.gz)


done
```

5.8. Masterfile creation

```
#!/bin/bash

path="" #insert the path to the OASIS3 downloaded folders file here

for i in {01..40}
do
convert_xfm -omat ${path}sub${i}/T1w/sub${i}_FLAIR_to_T1w_reg.mat -inverse ${path}sub${i}/T1w/sub${i}_T1w_reg.mat
convert_xfm -omat ${path}sub${i}/FLAIR/sub${i}_FLAIR_to_MNI.mat -concat ${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_to_MNI_lin.mat ${path}sub${i}/T1w/sub${i}_FLAIR_to_T1w_reg.mat.mat
done

> ${path}masterfile.txt
for i in {01..40}
do

echo "${path}sub${i}/FLAIR/sub${i}_FLAIR_masked.nii.gz ${path}sub${i}/T1w/sub${i}_T1w_reg.nii.gz ${path}sub${i}/SWI_MAG/sub${i}_swi_reg.nii.gz ${path}sub${i}/FLAIR/sub${i}_roi1.nii.gz ${path}sub${i}/FLAIR/sub${i}_roi2.nii.gz ${path}sub${i}/FLAIR/sub${i}_FLAIR_to_MNI.mat ${path}sub${i}/SWI_MAG/sub${i}_FLAIR_masked.nii.gz" >> ${path}masterfile.txt
done
```

5.9. BIANCA training

```
#!/bin/bash

for i in {4..5}
do
((ROI=i-3))
path="" #insert the path to the OASIS3 downloaded folders file here
path_copy=”${path}”
path_temp="${path}${ROI}"

mkdir ${path_temp}
cp ${path}masterfile.txt ${path_temp}/
path="${path_temp}/"



echo "FLAIR T1w swi"
echo "a"
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_10a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_15a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_20a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_25a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_30a -v

echo "b"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_10b -v
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_15b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_20b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_25b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_30b -v

echo "c"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_10c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_15c -v
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_20c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_25c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_30c -v

echo "d"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_10d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_15d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_20d -v
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_25d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_swi_30d -v



echo "FLAIR T1w"

echo "a"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_10a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_15a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_20a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --
```

```
trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_25a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_30a -v

echo "b"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_10b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_15b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_20b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_25b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --
```

```
trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_30b -v

echo "c"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_10c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_15c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_20c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_25c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_30c -v

echo "d"
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_10d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_15d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_20d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_25d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_T1w_30d -v



echo "FLAIR swi"

echo "a"
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_10a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_15a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_20a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_25a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_30a -v

echo "b"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_10b -v
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_15b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_20b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_25b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_30b -v

echo "c"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_10c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_15c -v
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_20c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_25c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_30c -v

echo "d"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_10d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_15d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_20d -v
```

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_25d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_swi_30d -v



echo "FLAIR"

echo "a"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_10a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_15a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_20a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --
```

```
trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_25a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_30a -v

echo "b"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_10b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_15b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_20b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_25b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35,36,37,38,39,40 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_30b -v
```

```
echo "c"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_10c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_15c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_20c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_25c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_30c -v

echo "d"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_10d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --
```

trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_15d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_20d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_25d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}FLAIR_30d -v

echo "swi"

echo "a"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_10a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_15a -v

```
bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_20a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_25a -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=31 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_30a -v

echo "b"


bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_10b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_15b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 --featuresubset=7 --matfeaturenum=7 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_20b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=21 --
```

```
trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_25b -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=21 --trainingnums=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,31,32,33,34,35,36,37,38,39,40 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_30b -v

echo "c"


bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_10c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_15c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_20c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_25c -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=11 --trainingnums=1,2,3,4,5,6,7,8,9,10,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,3
```

8,39,40 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_30c -v

echo "d"

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_10d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_15d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_20d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_25d -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=1 --trainingnums=10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40 --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder --saveclassifierdata ${path}swi_30d -v

```
echo "Leave One Out"

for j in {01..40}

do

mkdir ${path_copy}sub${j}/Bianca_incr

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1,2,3 --querysubjectnum=${j} --trainingnums=all --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder -o ${path}sub${j}/Bianca_incr/sub${j}_FLAIR_T1w_swi_all_roi$((ROI)).nii.gz -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1,2 --querysubjectnum=${j} --trainingnums=all --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder -o ${path}sub${j}/Bianca_incr/sub${j}_FLAIR_T1w_all_roi$((ROI)).nii.gz -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1,3 --querysubjectnum=${j} --trainingnums=all --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder -o ${path}sub${j}/Bianca_incr/sub${j}_FLAIR_swi_all_roi$((ROI)).nii.gz -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=1 --querysubjectnum=${j} --trainingnums=all --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder -o ${path}sub${j}/Bianca_incr/sub${j}_FLAIR_all_roi$((ROI)).nii.gz -v

bianca --singlefile=${path}masterfile.txt --labelfeaturenum=$((ROI)) --brainmaskfeaturenum=7 --querysubjectnum=${j} --trainingnums=all --featuresubset=7 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --trainingpts=2000 --nonlespts=10000 --selectpts=noborder -o ${path}sub${j}/Bianca_incr/sub${j}_FLAIR_T1w_swi_all_roi$((ROI)).nii.gz -v

done

done
```

5.10. BIANCA testing

```
#!/bin/bash


path_B=”” #insert path to BIANCA’s trained with WH and UK Biobank


for k in {4..5}
do
((ROI=k-3))
path="" #insert the path to the OASIS3 downloaded folders file here
path_copy=”${path}”
path_temp="${path}${ROI}"

path="${path_temp}/"
```

```
for i in {31..40}
do

echo ${i}
for n in {10..30..5}
do
bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --querysubjectnum=${i} --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --loadclassifierdata=${path}FLAIR_T1w_swi_${n}a  -o ${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_T1w_swi_${n}_roi$((ROI)).nii.gz
bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --querysubjectnum=${i} --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --loadclassifierdata=${path}FLAIR_T1w_${n}a  -o ${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_T1w_${n}_roi$((ROI)).nii.gz
bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --querysubjectnum=${i} --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --loadclassifierdata=${path}FLAIR_swi_${n}a  -o ${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_swi_${n}_roi$((ROI)).nii.gz
bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --querysubjectnum=${i} --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --loadclassifierdata=${path}FLAIR_${n}a  -o ${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_${n}_roi$((ROI)).nii.gz
bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=6 --querysubjectnum=${i} --featuresubset=6 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --loadclassifierdata=${path}swi_${n}a  -o ${path_copy}sub${i}/Bianca_incr/sub${i}_swi_${n}_roi$((ROI)).nii.gz

done
done

for i in {21..30}
```

```
do


echo ${i}

for n in {10..30..5}

do

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 -
-patchsizes=3 --loadclassifierdata=${path}FLAIR_T1w_swi_${n}b  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_T1w_swi_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_T1w_${n}b  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_T1w_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_swi_${n}b  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_swi_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_${n}b  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=6 --
querysubjectnum=${i} --featuresubset=6 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}swi_${n}b  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_swi_${n}_roi$((ROI)).nii.gz


done

done


for i in {11..20}

do
```

```
echo ${i}

for n in {10..30..5}

do

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 -
-patchsizes=3 --loadclassifierdata=${path}FLAIR_T1w_swi_${n}c  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_T1w_swi_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_T1w_${n}c  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_T1w_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_swi_${n}c  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_swi_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_${n}c  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=6 --
querysubjectnum=${i} --featuresubset=6 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}swi_${n}c  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_swi_${n}_roi$((ROI)).nii.gz


done

done


for i in {01..10}

do
```

```
echo ${i}

for n in {10..30..5}

do

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,2,3 --matfeaturenum=6 --spatialweight=2 -
-patchsizes=3 --loadclassifierdata=${path}FLAIR_T1w_swi_${n}d  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_T1w_swi_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_T1w_${n}d  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_T1w_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,3 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_swi_${n}d  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_swi_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}FLAIR_${n}d  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_FLAIR_${n}_roi$((ROI)).nii.gz

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=6 --
querysubjectnum=${i} --featuresubset=6 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path}swi_${n}d  -o
${path_copy}sub${i}/Bianca_incr/sub${i}_swi_${n}_roi$((ROI)).nii.gz


done

done

for z in {1..2}

do

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --
querysubjectnum=${i} --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --
patchsizes=3 --loadclassifierdata=${path_B}wmh_harmonisation-master-
BIANCA_training_datasets/BIANCA_training_datasets/Mixed_WH-
```

```
UKB_FLAIR_T1 -o
${path}sub${i}/Bianca_results/sub${i}_FLAIR_T1w_V_roi${n}.nii.gz

done

done
```

5.11. BIANCA performance

```
#!/bin/bash


path="" #insert path to oasis images here

for z in {1..2}

do

for i in {01..40}

do

for n in {10..30..5}

do

bianca_overlap_measures ${path}sub${i}_FLAIR_T1w_swi_${n}_roi${z}.nii.gz 0.9
${path}sub${i}_roi${z}.nii.gz 1

bianca_overlap_measures ${path}sub${i}_FLAIR_T1w_${n}_roi${z}.nii.gz 0.9
${path}sub${i}__roi${z}.nii.gz 1

bianca_overlap_measures ${path}sub${i}_FLAIR_swi_${n}_roi${z}.nii.gz 0.9
${path}sub${i}__roi${z}.nii.gz 1

bianca_overlap_measures ${path}sub${i}_FLAIR_${n}_roi${z}.nii.gz 0.9
${path}sub${i}__roi${z}.nii.gz 1

bianca_overlap_measures ${path}sub${i}_swi_${n}_roi${z}.nii.gz 0.9
${path}sub${i}__roi${z}.nii.gz 1

done

done

done
```

```
#To load the results in the same place
for z in {1..2}
do
mkdir ${path}Bianca_incr_roi${z}

for n in {10..30..5}
do
>${path}Bianca_incr_roi2/FLAIR_T1w_swi_${n}_roi${z}_0.9.txt
>${path}Bianca_incr_roi2/FLAIR_T1w_${n}_roi${z}_0.9.txt
>${path}Bianca_incr_roi2/FLAIR_swi_${n}_roi${z}_0.9.txt
>${path}Bianca_incr_roi2/FLAIR_${n}_roi${z}_0.9.txt
>${path}Bianca_incr_roi2/swi_${n}_roi${z}_0.9.txt
done

for i in {01..40}
do
for n in {10..30..5}
do
cat ${path}sub${i}/Bianca_incr_roi2/results_roi2/Overlap_and_Volumes_sub${i}_FLAIR_T1w_swi_${n}_roi${z}_0.9.txt >> ${path}Bianca_incr_roi2/FLAIR_T1w_swi_${n}_roi${z}_0.9.txt
cat ${path}sub${i}/Bianca_incr_roi2/results_roi2/Overlap_and_Volumes_sub${i}_FLAIR_T1w_${n}_roi${z}_0.9.txt >> ${path}Bianca_incr_roi2/FLAIR_T1w_${n}_roi${z}_0.9.txt
```

```
cat ${path}sub${i}/Bianca_incr_roi2/results_roi2/Overlap_and_Volumes_sub${i}_FLAIR_swi_${n}_roi${z}_0.9.txt >> ${path}Bianca_incr_roi2/FLAIR_swi_${n}_roi${z}_0.9.txt

cat ${path}sub${i}/Bianca_incr_roi2/results_roi2/Overlap_and_Volumes_sub${i}_FLAIR_${n}_roi${z}_0.9.txt >> ${path}Bianca_incr_roi2/FLAIR_${n}_roi${z}_0.9.txt

cat ${path}sub${i}/Bianca_incr_roi2/results_roi2/Overlap_and_Volumes_sub${i}_swi_${n}_roi${z}_0.9.txt >> ${path}Bianca_incr_roi2/swi_${n}_roi${z}_0.9.txt

done

done



for n in {10..30..5}

do

 ex -s +'v/\S/d' -cwq ${path}Bianca_incr_roi2/FLAIR_T1w_swi_${n}_roi${z}_0.9.txt

 ex -s +'v/\S/d' -cwq ${path}Bianca_incr_roi2/FLAIR_T1w_${n}_roi${z}_0.9.txt

 ex -s +'v/\S/d' -cwq ${path}Bianca_incr_roi2/FLAIR_swi_${n}_roi${z}_0.9.txt

 ex -s +'v/\S/d' -cwq ${path}Bianca_incr_roi2/FLAIR_${n}_roi${z}_0.9.txt

 ex -s +'v/\S/d' -cwq ${path}Bianca_incr_roi2/swi_${n}_roi${z}_0.9.txt

 done


> ${path}results_V_roi${z}.txt

for i in {01..40}

do

bianca_overlap_measures sub${i}_FLAIR_T1w_V.nii.gz 0.9 sub${i}_roi${z}.nii.gz 1
```

```
cat
${path}sub${i}/Bianca_results/Overlap_and_Volumes_sub${i}_FLAIR_T1w_V_0.9.t
xt >> ${path}results_V_roi${z}.txt

done

done
```

B Appendix B

5.12. FLAIR T1w run

#(The reader must be wary that, due to the high amount of images processed, the code could take up to and even over two month of non-stop work to complete, this is why, if the reader wants to recreate our work, we suggest to split the subjects in equal groups and launching the code multiple times in more terminals, accordingly to the CPU cores, their performance and cooling of the components.)

```
#!/bin/bash

path="" #insert path to oasis images here

path_B="" #insert path to BIANCA's trained with WH and UK Biobank

for i in {000..734}

do

mkdir ${path}sub${i}/T1w/fsl_anat

cp ${path}sub${i}/T1w/sub${i}_T1w.nii.gz ${path}sub${i}/T1w/fsl_anat

fsl_anat -i ${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.nii.gz

make_bianca_mask ${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_biascorr
${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_fast_pve_0
${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/MNI_to_T1_nonlin_field.nii.gz
```

```
bet ${path}sub${i}/FLAIR/sub${i}_FLAIR.nii.gz ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain.nii -R

/usr/local/fsl/bin/fast -B ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain


/usr/local/fsl/bin/flirt -in ${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_biascorr_brain.nii.gz -ref ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain_restore.nii.gz -out ${path}sub${i}/T1w/sub${i}_T1w_reg.nii.gz -omat ${path}sub${i}/T1w/sub${i}_T1w_reg.mat -dof 6

/usr/local/fsl/bin/flirt -in ${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_biascorr_bianca_mask.nii.gz -ref ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain_restore.nii.gz -out ${path}sub${i}/T1w/sub${i}_T1w_bianca_mask_reg.nii -applyxfm -init ${path}sub${i}/T1w/sub${i}_T1w_reg.mat


fslmaths ${path}sub${i}/FLAIR/sub${i}_FLAIR_brain_restore.nii.gz -mas ${path}sub${i}/T1w/sub${i}_T1w_bianca_mask_reg.nii ${path}sub${i}/FLAIR/sub${i}_FLAIR_masked.nii.gz

convert_xfm -omat ${path}sub${i}/T1w/sub${i}_FLAIR_to_T1w_reg.mat -inverse ${path}sub${i}/T1w/sub${i}_T1w_reg.mat

convert_xfm -omat ${path}sub${i}/FLAIR/sub${i}_FLAIR_to_MNI.mat -concat ${path}sub${i}/T1w/fsl_anat/sub${i}_T1w.anat/T1_to_MNI_lin.mat ${path}sub${i}/T1w/sub${i}_FLAIR_to_T1w_reg.mat.mat

mkdir ${path}sub${i}/Bianca_results

bianca --singlefile=${path}masterfile.txt --brainmaskfeaturenum=1 --querysubjectnum=${i} --featuresubset=1,2 --matfeaturenum=6 --spatialweight=2 --patchsizes=3 --loadclassifierdata=${path_B}wmh_harmonisation-master-BIANCA_training_datasets/BIANCA_training_datasets/Mixed_WH-UKB_FLAIR_T1 -o ${path}sub${i}/Bianca_results/sub${i}_FLAIR_T1w_V_roi${n}.nii.gz

> ${path}sub${i}/Bianca_results/results.txt

bianca_cluster_stats ${path}sub${i}/Bianca_results/sub${i}_FLAIR_T1w_V.nii.gz 0.9 1 >> ${path}sub${i}/Bianca_results/results.txt
```

```
done

for i in {000..734}
do
grep -Eo '[0-9.]{1,15}' ${path}sub${i}/Bianca_results/results.txt | tail -1 >> ${path}results_all.txt
done
i=0
cd ${path}
>${path}volumes.csv
echo "Subject,MR_ID,Bianca_Volume" >> ${path}volumes.csv
for g in *
do

path_t="${path}sub$(printf %03d $((10#$i)) )/T1w/fsl_anat/sub$(printf %03d $((10#$i)) )_T1w.anat/"
>${path}/sub$(printf %03d $((10#$i)) )/Bianca_results/Vol_pve.txt
echo ${i}

for k in {0..2}
do
vol=`fslstats ${path_t}/T1_fast_pve_${k}.nii.gz -V | awk '{print $2}'`
mean=`fslstats ${path_t}/T1_fast_pve_${k}.nii.gz -M`
tissuevol=`echo "$mean * $vol" | bc -l`
done
echo ${tissuevol}>>${path}sub$(printf %03d $((10#$i)) )/Bianca_results/Vol_pve.txt
```

```
str1=$(grep -Eo '[0-9.]{1,15}' ${path}sub$(printf %03d $((10#$i)) )/Bianca_results/results.txt | tail -1)

i=$(printf %03d $((10#$i+1)) )

str2=$(echo $g | cut -d '_' -f 1)


res1=$(awk '{print $1/$2}' <<<"${str1} ${tissuevol}")

res="${res1/,/.}"

echo "${str2},${g},${res}" >> ${path}volumes.csv

done
```

5.13. Dataset join

```
import pandas as pd

path = "" #insert the path to the csv files

adcr = pd.read_csv(path + "ADRC.csv")

vol = pd.read_csv(path + "volumes.csv")

adcr.columns = adcr.columns.str.replace(' ', '_')

vol.columns = vol.columns.str.replace(' ', '_')

adcr.dropna(how='all', axis=1, inplace=True)


df = pd.DataFrame(columns= df_temp.columns)

df.head()

test = df_temp.MR_ID[2]

g = len(df_temp.ADRC_ADRCCLINICALDATA_ID)

str0 = df_temp.ADRC_ADRCCLINICALDATA_ID[123]

diff = abs(int(str0[len(str0) - 4:]) - int(test[len(test) - 4:]))
```

```
df = []


max = 0


scans = 0
while scans < len(df_temp.ADRC_ADRCCLINICALDATA_ID) - 1:
    scan0 = df_temp.MR_ID[scans]
    diff0 = 9999
    adv = 0
    print(scans)
    while scans + adv < len(df_temp.ADRC_ADRCCLINICALDATA_ID) and scan0
== df_temp.MR_ID[scans + adv]:
        str0 = df_temp.ADRC_ADRCCLINICALDATA_ID[scans + adv]
        diff = abs(int(str0[len(str0) - 4:]) - int(scan0[len(scan0) - 4:]))
        if diff < diff0:
            diff0 = diff
            idx = scans + adv
        adv += 1
        if adv > max:
            max = adv
    scans += adv
    if scans < len(df_temp.ADRC_ADRCCLINICALDATA_ID):
        df.append(df_temp.iloc[idx])


df = pd.DataFrame(df)
```

```
df.dropna(how='all', axis=1, inplace=True)

df_temp = df.copy()


df_temp.drop(df_temp.columns.difference(['MR_ID','Subject', 'Age', 'ageAtEntry', 'cdr', 'height', 'weight', 'M/F', 'Hand', 'ALCOHOL', 'TOBAC30', 'TOBAC100', 'DIABETES']), 1, inplace=True)

df = pd.merge(df_temp, vol, on=["MR_ID"])

df.to_excel(path+"df.xlsx")
```

List of Figures

[3.1] Wide WMH lesions clearly visible on a FLAIR image as white marks.

[4.1] A schematic of a Decision Tree

[2.2.] An example of SVM.

[2.3.] An example of an Artificial Neural Network.

[3.1.] “MR Sessions” Tab opened withing the OASIS3 dedicated webpage.

[3.2.] Script used to download images from the OASIS3 dataset. Required inputs are displayed.

[3.3.] Example of four FLAIR images. The first three are all ineligible for WMH segmentation: the top left scan presents very strong motion artifacts; the top right scan shows extremely big ventricles which prevents the shape of the brain from being considered “regular”; the bottom left scan has visible WMH lesions which, however, are very limited in numerosity and dimension. On the contrary, the last FLAIR image has heavy lesional load and is artifact-free, thus representing a valid example for the purposes of our work.

[3.4.] Jim8 startup page.

[3.5.] Jim8 ROI toolkit window.

[3.6.] Higher intensity area automatically segmented with a light blue contour by the Jim8 "ROI Analysis" Toolkit.

[3.7.] The Jim8 Masker tool window.

[3.8.] Original (left panel) and brain-extracted (right panel) FLAIR image.

[3.9.] Original FLAIR image (left panel) later BF corrected using FAST (central panel). The outline of the magnetic field inhomogeneities is also displayed (right panel).

[3.10.] Noticeable spatial resolution loss suffered from a T1-weighted image (upper panel) after being registered to the corresponding FLAIR scan (lower panel).

[3.11.] Binary image of the CSF mask (top panel); binary mask created using the "make_bianca_mask" command (central panel); bianca mask registered and applied to the FLAIR image (bottom panel).

[3.12.] Example of a possible Masterfile, directly taken from the user guide of BIANCA (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/BIANCA/Userguide).

[3.13.] Filtering option window from XNAT.

[3.14.] The XNAT download page. All files and modalities are already selected by default, but the user can modify according to his own needs.

[3.15.] "Join to other data" window from XNAT.

[3.16.] Overall graph of the designed ANN.

[3.17.] Example of a Confusion Matrix.

[4.1.] Visual assessment of the results: WM extracted from FLAIR (top panel); output from BIANCA (middle panel); manual segmentation mask (bottom panel).

[4.2.] Boxplot of the DICE index (represented on the y axis) between BIANCA outputs obtained training with an increasing number of subjects (represented on the x axis) and the corresponding "preliminary" segmentations. Results are relative to the whole dataset (40 subjects). Black lines indicate significant differences (p-value < 0.05) between adjacent groups, calculated using a RANOVA test. The

analysis is repeated in every panel for the different combinations of MRI modalities used for training.

[4.3.] Boxplot of the DICE index (represented on the y axis) between BIANCA outputs obtained training with an increasing number of subjects (represented on the x axis) and the corresponding "expert" segmentations. Results are relative to the whole dataset (40 subjects). Black lines indicate significant differences (p-value < 0.05) between adjacent groups, calculated using a RANOVA test. The analysis is repeated in every panel for the different combinations of MRI modalities used for training.

[4.4.] Boxplot of the DICE between "preliminary" and "expert" manual masks for the whole dataset involved in the study.

[4.5.] Visual example of the difference existing between "preliminary" (central panel – yellow area) and "expert" (right panel) segmentations, outlined on a FLAIR image (left panel – red area).

[4.6.] Boxplot of the DICE index (represented on the y axis) between BIANCA outputs obtained training and testing with leave-one-out validation on the whole dataset (40 subjects). Results are relative to the different combination of MRI modalities used (represented on the x axis): FLAIR (in blue), FLAIR + SWI (in orange), FLAIR + T1w (in green), FLAIR + T1w + SWI (in red), SWI (in purple). Full lines indicate significant differences (p-value < 0.05) between adjacent distributions, calculated using a RANOVA test.

[4.7.] Pairwise comparisons of the distributions present in Fig. 4.6. Full lines indicate significant differences (p-value < 0.05) between distributions, calculated using a RANOVA test.

[4.8.] Boxplot of the DICE index (represented on the y axis) between BIANCA outputs obtained with both a mixed training strategy (training on Whitehall + UK Biobank and testing on OASIS3 – represented in blue) and a single-site training strategy (training and testing on OASIS3 using a 4-fold cross validation – represented in orange). Full lines indicate significant differences (p-value < 0.05) between adjacent distributions, calculated using a paired t-test.

[4.9.] Boxplots of WMH percentage volumes (calculated as % of the total brain volume – represented on the y axis) for three different CDR groups: CDR = 0 (blue), CDR = 0.5 (orange), CDR = 1 (green). Full lines

indicate significant differences (p-value < 0.05) between adjacent distributions, calculated using a one-way ANOVA test.

[4.10.] Scatterplot between WMH volumes (calculated as % of the total brain volume – represented on the y axis) and age (represented on the x axis) for CDR = 0 (blue) and CDR > 0 (orange – indicated as CDR = 1 in the legend) populations. Regression lines with 95% confidence interval are also displayed.

[4.11.] Confusion Matrix of the best SVM model.

[4.12.] Permutation importance (reported on the x axis) of the different input variables (reported on the y axis) involved in the SVM model.

[4.13.] Confusion Matrix of the best RFC model with PCA.

[4.14.] Linear combination coefficients for the different principal components. Each value has to be multiplied by its corresponding continuous variable (being either Age, ageAtEntry, height, weight or Bianca_Volume) and eventually sum up to the others to get the final result.

[4.15.] Permutation importance (reported on the x axis) of the different input variables (reported on the y axis) involved in the RFC model with prior PCA.

[4.16.] Confusion Matrix of the best RFC model without PCA.

[4.17.] Permutation importance (reported on the x axis) of the different input variables (reported on the y axis) involved in two similarly performing RFC models. The upper and lower panel showed the model with 76% and 74% of accuracy, respectively.

[4.18.] Confusion Matrix of the best ANN model.

List if Tables

[4.1.] Segmentation performance obtained testing with: i) data from the OASIS3; ii) data from the Whitehall and UK Biobank datasets. The training phase was conducted in both cases using the Whitehall + UK Biobank dataset from(Bordin et al. 2021).

Acknowledgments

Ringrazio di cuore la mia famiglia, che mi è stata accanto durante questo importante percorso.

Un grazie sincero anche ad Arianna, che mi ha sempre supportato ed incoraggiato.

Un ringraziamento speciale alla sezione Kino di un certo gruppo di amanti dei cereali, con cui ho passato degli splendidi martedì sera durante questo cammino.

Grazie anche a tutti gli amici acquisiti e persi in questi anni.

Un ulteriore, ma non meno importante, e doveroso ringraziamento al Professor Baselli, a Valentina e Marcella, senza cui questa Tesi non sarebbe stata possibile.