FEATURE SELECTION METHODS FOR PREDICTING PRE-CLINICAL STAGE IN ALZHEIMER'S DISEASE

A Degree Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Marcel Catà Villà

In partial fulfilment

of the requirements for the degree in

Science and Telecommunications Technologies Engineering

Advisor: Verónica Vilaplana Besler

Co-advisor: Adrià Casamitjana Díaz

Barcelona, June 2014

Abstract

Alzheimer's disease is still an incurable disease. Nevertheless, some of its biomarkers suffer changes in the early stages of the disease, long before clinical symptoms appear. In order to determine how biomarkers obtained from magnetic resonance (MRI) techniques affect the disease's evolution, machine learning techniques have been used to design and implement a classification system so as to predict the stages in which several patients belong. One of the main objectives of this project is reducing the number of data to manage, since MRI provide a large volume of data for each patient. As a result, we will focus on the stage of reduction and extraction of characteristics of the classifier which may be relevant for the mentioned problem. We will carry out an exhaustive analysis of different methods of selection of features to apply to biomedical data related to Alzheimer's disease. Results obtained will also be applicable to other fields. Finally, we will assess these methods with a multimodal data base provided by the collaboration agreement with Pasqual Maragall Foundation (FPM).

Resum

La malaltia de l'Alzheimer és encara una malaltia incurable. Tanmateix, alguns dels seus biomarcadors es pateixen canvis durant les primeres etapes de la malaltia, molt abans de presentar símptomes clínics. Per a determinar com afecten a l'evolució de la malaltia els biomarcadors obtinguts a partir de tècniques de ressonància magnètica (MRI), s'han utilitzat tècniques de machine learning per a dissenyar i implementar un sistema de classificació per a predir les etapes en què es troben diversos pacients. Un dels principals objectius d'aquest projecte és reduir el nombre de dades a tractar, ja que les MRI proporcionen un gran volum de dades de cada pacient. En conseqüència, ens centrarem en l'etapa de reducció i extracció de característiques del classificador que poden ser rellevants per al problema esmentat. Realitzarem una anàlisi exhaustiva de diferents mètodes de selecció de característiques per a aplicar-los a dades biomèdiques relacionades amb la malaltia de l'Alzheimer. Els resultats obtinguts també podran aplicar-se en altres camps. Finalment, avaluarem els mètodes amb una base de dades multimodal proporcionada pel conveni de col·laboració amb la Fundació Pasqual Maragall (FPM).

Resumen

La enfermedad del Alzheimer es aún una enfermedad incurable. Sin embargo, algunos de sus biomarcadores sufren cambios durante las primeras etapas de la enfermedad, mucho antes de presentar síntomas clínicos. Para determinar cómo estos afectan a la evolución de la enfermedad los biomarcadores obtenidos a partir de técnicas de resonancia magnética (MRI), se han utilizado técnicas de machine learning para diseñar e implementar un sistema de clasificación con el fin de predecir las etapas en las que se encuentran distintos pacientes. Uno de los principales objetivos de este proyecto es reducir el número de datos a tratar, ya que las MRI proporcionan un gran volumen de datos de cada paciente. En consecuencia, nos centraremos en la etapa de reducción y extracción de características del clasificador que pueden ser relevantes para el problema mencionado. Realizaremos un análisis exhaustivo de distintos métodos de selección de características para aplicarlos a datos biomédicos relacionados con la enfermedad del Alzheimer. Los resultados obtenidos también podrán aplicarse en otros campos. Finalmente, evaluaremos los métodos con una base de datos multimodal proporcionada por el convenio de colaboración con la Fundació Pasqual Maragall (FPM).

Index

1. Context of the project

1.1 Introduction

Alzheimer's disease currently affects more than 36 million people in the world. A patient's brain already suffers changes during the earliest stages of the disease, long before showing any clinical symptoms. That is why researchers are focused in finding out what these changes are and where they take place, in order to determine indicators that help predicting the development of the Alzheimer’s disease.

Unfortunately, nowadays there is no cure for this disease. In fact, when this disease is diagnosed, the pain in brain is irreversible. However, scientists affirm that during earliest stages of the Alzheimer’s disease, when still there are no clinic symptoms, certain areas in the brain already suffer changes that might help to detect Alzheimer’s disease. In fact, the earliest pathological changes occur about 20 years before the onset of the first symptoms. That is the main reason to try to predict Alzheimer’s disease based on changes in different parts of the brain.

Figure 1 shows the evolution of different biomarkers along the different clinical stages of the disease. Depending on which stages we want to discriminate, some will be more useful than others. Among them, we are focusing on the blue curve, which refers to neuroimaging biomarkers. As these neuroimaging biomarkers appear on pre-clinical stage of the disease, we will be using them in order to discriminate between premature stages of the disease.

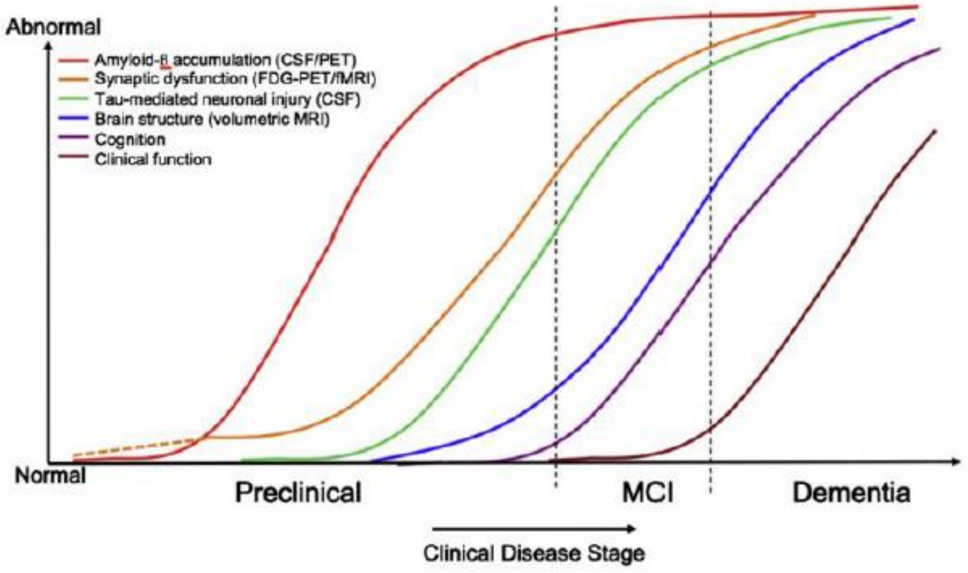


Figure 1: Hypothetical model of the progress of stages in Alzheimer's disease against different changes observed in the pacient (Sperling, Aisen et al; 2011).

Four different clinical stages can be identified when referring to Alzheimer’s disease. Subjects from the HCB (Hospital Clínic de Barcelona) database are labeled to belong to these stages:

- Normal or normal control (NC): healthy subjects.

- Pre-clinical (PC): they do not suffer any changes in cognition or clinic trials, but their brain already suffers damage.
- Mild cognitive impairment (MCI): they have less cognition capabilities but it may not be due to Alzheimer’s disease.
- Alzheimer’s disease (AD): they do have the disease.

As a matter of fact, in this project the focus is on the differences between pre-clinical and normal stages. In the Alzheimer’s Disease continuum, we observe that neuroimaging can provide valuable information regarding the development of the preclinical phase of AD. Several alterations are found to be relevant during this phase, which are measurable structural, functional and diffusion MR imaging.

1.2 State of the art

During the last years, research related to brain diseases have been focused into clinical analyses and biomedical experiments. As a result of that, there is plenty of literature about how to acquire and preprocess data from MRI scans. Those processes have evolved as a consequence of the advances in technology. That had made the investigation move towards experiments that used those methods on behalf of methods that used other biomedical markers, which were sometimes difficult to obtain.

Therefore, recent investigations have shown a great interest towards machine learning techniques, which can be applied to the prediction of brain diseases. In (Shao), there is an example of machine learning techniques applied to Alzheimer’s disease. However, there can be seen that PC stage is not considered and also that results are only given using accuracy metric, which is not fair in case of unbalanced classes, as we found that recall or f1-score are more relevant.

These techniques allowed investigators to obtain quite good results that have helped to multiply the investigation in that field. Firstly, and focusing on Alzheimer’s disease, only three different stages were considered: NC, MCI and AD [3]. That is because it is difficult to determine the difference between normal (NC) and pre-clinical (PC) stages of the disease. In fact, some recent publications do consider that pre-clinical stage [6] although they do not use it using machine learning.

Given all these points, there has been some obstacles when dealing with MRI data. One of the most important is the huge amount of features compared to the little amount of subjects (or samples). That leads to the problem known as the curse of dimensionality, which is due to the high dimensional space. With small and fixed number of training samples, the distance between them is huge due to the great amount of dimensions. As a result, everything is far from others, losing predictive power.

Another common problem when dealing with high dimensional datasets is overfitting, which means that a trained classifier will be well adapted to the training data but will obtain poor results on an independent test set. That is why data reduction helps the improvement of machine learning algorithms where there are lots of features.

Feature selection methods are used to face these issues. Although no feature selection analysis is found in literature regarding AD, in [4], there are many kinds of methods that can be applied to our problem.

1.3 Previous work

This is a project carried out at the Image and Video Processing Group (GPI) from the Signal Theory and Communications Department (TSC) at the Technical University of Catalonia (UPC) in collaboration with the Pasqual Maragall Foundation. In fact, this project's goal is to improve one part of a previous project also carried out at the GPI.

In this project we used Python language to develop the software used to make experiments. This software consists on a classifier that loads data from the HCB database, which will be explained in section 1.4, and works with it.

The structure of the software follows the typical pipeline of a machine learning problem: it is divided in pre-processing data, feature selection, classification and performance assessment. These different main stages were already implemented before the start of this part of the project. However, analyzing the parts of the project and the nature of the problem, the feature selection stage was thought as one of the important steps that could lead to further improvement, so this project was focused in that part.

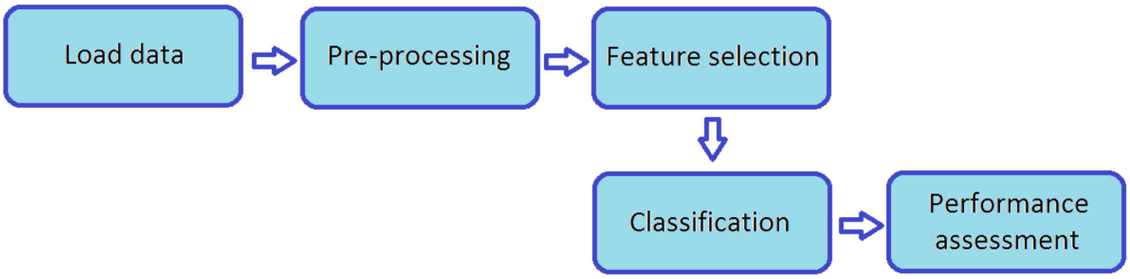


Figure 2: Pipeline of the classification system

First of all, data must be loaded to the classifier. As data has different ranges and values, therefore a pre-processing stage is needed. Then, most relevant features are selected and classification is done. Finally, in order to evaluate the performances of our classification system a leave-one-out (LOO) strategy is chosen.

That LOO strategy consists on using all samples except one to train the classifier and the remaining one to test. It is computationally expensive but can be afforded as our database has little amount of samples. This strategy is known to have low bias and, assuming that in every iteration the model is stable, we can compute the confusion matrix (formed by TP, TN, FP, FN) gathering all predictions from every iteration.

We proposed several choices for our classification algorithm: finally, a Regularized Logistic Regression was preferred among the others. For that, we had to optimize the hyperparameter value for our input data. We, again, use a LOO strategy to cross-validate this hyperparameter, splitting the training data into train and validation.

As a result of those strategies, that nested leave-one-out has increased the computational cost of the software, as the number of operations grows exponentially: in every iteration of the outer LOO, we have an internal LOO to cross-validate.

Finally, the performance metrics used are precision, recall (or sensitivity), specificity, accuracy and the f1-score. These metrics are computed using the values from the confusion matrix as shown below:

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{FP + TN} \qquad Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$f1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

In addition, the cost function to optimize the regularization parameter of the LR classifier is the f1-score.

1.4 Data

The dataset we will be working with was elaborated in the Hospital Clínic de Barcelona (HCB) and provided by the Pasqual Maragall Foundation (FPM). This dataset contains subjects that are divided into 4 different stages of Alzheimer's Disease, which are normal control (NC), pre-clinical (PC), mild cognitive impairment (MCI) and Alzheimer's Disease (AD). Therefore, as the focus of this project is the classification between pre-clinical (PC) and normal control (NC) stages, subjects from other stages will not be used.

The study that we are overtaking is transversal, which means that data is not gathered from same subject along time but from different subjects that have been labeled into these different stages. Thus, the intrinsic variability of biological data between different subjects makes the analysis more complex. In other words, when comparing subjects from different stages, the comparison will be between two different people, one from each class, whereas it would be more interesting to compare the same subject once its disease has evolved into a higher stage, as in a longitudinal study. Even though, it is hard to find volunteers that can be labeled into those classes. As a consequence, the amount of subjects is small.

To sum up, in table 1 there are the numbers about NC and PC subjects in the HCB database. As it can be seen in the table, classes are unbalanced, which will also be a fact to be taken into consideration when assessing the results.

	Grey Matter volume	**Structural connectivity**
NC	69	44
PC	19	12
Total	88	56

Table 1: Number of subjects in HCB database.

After acquisition and standard preprocessing, the MRI scans are registered to an atlas (AAL) that divides the brain into 90 anatomical regions (ROI: regions of interest) that indicate macroscopic brain structures. From different MRI modalities we get different kind of data ready to process.

Data needs to be pre-processed to better fit our problem, as different scales in feature values do not help, but make classification harder. It aims all samples to be comparable without losing discriminative power. In fact, there are different pre-processing methods depending on the modality.

- <u>Grey matter volume (GMV) (90 features):</u> as its names indicates, the value accounts for the volume of grey matter in the ROI.
  That value is normalized by dividing each grey matter volume in each region by the volume of the subject's brain. Then, is escalated in the range [0,1].

- <u>Structural connectivity matrices (SCM) (8100 features):</u> they contain an estimation of the number of fibers that connect two ROI's of the brain. That connectivity matrix is obtained by using diffusion MRI (Figure 3). As their values somehow illustrate connections between areas of the brain, there are 8100 features (90x90).

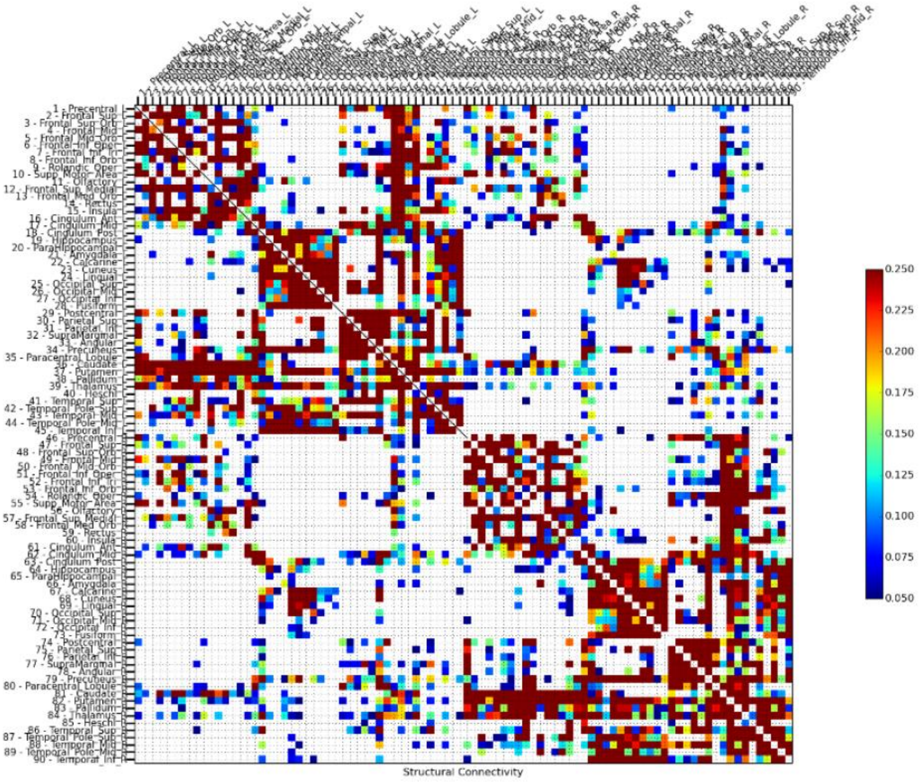


Figure 3: Example of a structural connectivity matrix. Example from [7]

As we are dealing with a great amount of features, a first manual reduction is done. First, we consider only half of the matrix as connections between same areas are only considered once. Secondly, the diagonal is nullified as areas are not self-connected. Finally, as well as in grey matter volume, the data is escalated in the range [0,1].
Nonetheless, there is also an extra step when talking about structural matrices: we are keeping the top 10% percentile of the features.

2. Feature Selection

2.1 Introduction

The feature selection stage of the classifier is an essential part due to the composition of our database. As it has been said before, we are working with a database that has a great amount of features in comparison to its small number of samples, which leads us to the problem known as curse of dimensionality, which is caused by the fact of having many dimensions (features) and not many samples. That distribution outlines the need for data reduction on features used in classification; otherwise the classifier will be prone to overfit.

In addition to that, feature selection also helps us gaining understanding of the process, as knowing the most important features of the dataset will allow us to visualize where the changes between Alzheimer's stages are.

Among the different methods that exist to select the best set of features, feature selection methods are usually grouped in three: filters, wrappers and ensemble methods [4].

2.2 Filter methods

The first set of methods that we are going to present is filter methods. These methods, which are also known as ranking methods, are based on assessing each feature and assigning it a score. After that, features are ordered using that metric and the best ones in the ranking are selected. That score is generally assigned using a metric that takes into account the value of the features across subjects in comparison to their label (the class they belong).

1. Evaluate features according to a method
2. Order them by importance
3. Select K top features

Eventually, there are some remarkable advantages when using them, such as computational cost, as only one calculation per feature is needed in most of the cases and there are no classifiers implied in the selection process. This reason makes those methods to be the most used or, at least, the first to be tried when facing a problem where there is no other related work on this stage.

There has to be mentioned that after the computation of scores, the majority of these methods below have the possibility to perform a hypothesis test to evaluate the result and compute the correspondent p-value.

2.2.1 Pearson correlation coefficient

The Pearson Correlation coefficient considers the relation between features (X) and labels (Y) based on the expression (1). The $\mathrm{cov}(X, Y)$ means the covariance between those variables and σ_X, σ_Y are the variances of each variable.

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} \qquad (1)$$

This metric measures the linear relationship between two variables, or in other words, how the data can be fitted by a straight line. However, Pearson coefficient is not useful to detect other relationships between variables, such as quadratic.

Like other correlation coefficients, its value varies between -1 and +1 and indicates how well a regression line fits the data. These extreme values imply an exact linear relationship, either positive or negative, and 0 implies that there is no linear relation between both analyzed variables. As we are only interested in finding strong relations and we do not care about them being positive or negative, we use the absolute value of the correlation to rank features.

As we are working with groups of samples and not random variables, we will compute the Pearson coefficient (r) using the estimation of covariance and variances as in (2). That must be taking into account that classification is binary, so only two classes are used.

$$r_{XY} = \frac{\sum_{i=1}^{N}(x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_i - \bar{x})^2 \cdot \sum_{i=1}^{N}(y_i - \bar{y})^2}} \qquad (2)$$

Here, we assume that $\bar{y}$ is the estimated mean value for elements for y and $\bar{x}$ the estimated mean value for x.

In our case of study, Pearson correlation coefficient might be useful as it has been detected (using medical tests) that there exists a dependency between stages of the Alzheimer Disease and neuroimaging biomarkers. However, the main drawback of Pearson correlation is that is very sensitive to outliers, which might appear due to intrinsic variability, as data is obtained from different people. Even all of that, Pearson correlation coefficient is one of the most used coefficients to select important features from a feature set.

2.2.2 Kendall's tau coefficient

The Kendall Tau correlation coefficient measures the correlation between the order of two different variables X and Y. These variables will be ranked based on their value, so this method cannot be used with categorical variables. Then, Kendall Tau correlation assesses the similarity between orders of variables.

1. Substitute the values of samples in X and Y by their rank number if values were ordered (if it is repeated, use the mean value of the positions)
2. Compare contiguous features and assess their values according to their order.
3. Compute τ.

Ordering samples based on their value is simple. However, when comparing contiguous features (as in step 2.), there can be several possibilities. Considering x_i and x_j as two consecutive samples, they can be considered concordant pairs (P) if they meet one of the following conditions:

$$x_i < x_j \;\; and \;\; y_i < y_j$$

$$x_i > x_j \;\; and \;\; y_i > y_j$$

They are considered discordant pairs (Q) if they meet one of the following:

$$x_i > x_j \;\; and \;\; y_i < y_j$$

$$x_i < x_j \;\; and \;\; y_i > y_j$$

Otherwise, there will be a tie. Those ties are not taken into consideration when using the Kendall tau correlation as in (3), being N the number of samples.

$$\tau = \frac{P-Q}{M} \;\; where \;\; M = \frac{1}{2}N \cdot (N-1) \qquad (3)$$

However, if there is a case where lots of ties happen, the Kendall tau-b correlation is computed, as in expression (4). That method takes ties in *X* or in *y* into consideration, although it does not account for ties in both variables at the same time. We will name T the number of ties only in *X* and U the number of ties only in *y*.

$$\tau_B = \frac{P-Q}{\sqrt{(P+Q+T)\cdot(P+Q+U)}} \qquad (4)$$

The scores obtained after computing τ_B between features and labels have values between [-1, +1]. Therefore, +1 will indicate that variables are ordered in the same way, -1 that they are ordered inversely and 0 that there is no relation between both variable orders.

Kendall's tau correlation does not only detect linear dependency but also other non linear dependencies between variables. When applying to our problem, we will use Kendall tau-b because we are comparing a feature with its labels. Those labels are binary and because of that there are a lot of draws in y that need to be computed. Therefore, the τ_B coefficients can be easily used to rank the features depending on their score and then select the K best among them.

In fact, tau-b model solves the fact that Kendall's tau correlation does not account for ties. In addition, this metric is also consistent in front of the outliers, which is a useful fact as we are dealing with clinical data.

2.2.3 T-test

The t-test method is used to compare if two populations have the same mean or not. There are two different t-test expressions possible: Student t-test and Welch t-test. The difference is in the fact that the Student t-test assume that variances of samples are equal, whereas Welch t-test does not.

Consequently, due to our data distribution, we will use Welch t-test, which also performs well if classes are unbalanced. The t-test value is computed using the expression (5), where $\overline{X_1}$ is the sample mean of the first sample, s_1^2 its sample variance and N_1 its size. Analogously, $X_2, s_2^2, \; N_2$

for the second sample. In case of Welch t-test, an unpooled variance is computed as different classes are assumed to have different variances.

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_2^2}{N_2}}} \qquad (5)$$

This test assesses the difference between the mean values of samples from each of the two classes compared. Therefore, we assume that different means, represented by the biggest values of t, indicate the most discriminative features. As a consequence, the absolute value of t-test will be used, as there it makes no difference the fact that t-test is positive or negative.

2.2.4 F-test

The F-test method is used to compare if two populations have the same variance. We are using ANOVA F-test to evaluate differences between classes. That method compares how different classes are from the assumption that they yield the same mean response. The score for the features is computed as follows:

$$F = \frac{\sum_{j=1}^{K} n_j \cdot (\overline{y_j} - \bar{y})^2 \Big/ K-1}{\sum_{i=1}^{N} (y_i - \bar{y})^2 \Big/ N-K} \qquad (6)$$

The elements of the expression are:

- n_j: the number of samples from variable y that belong to class j
- N : number of samples of y
- K : number of classes
- $\bar{y}$: mean estimated value of y
- y_j : mean estimated value of elements from y that belong to class j

Our analysis will be always binary, that is, between two different classes. Then, for each feature (y) we will compare the mean from subjects of each class (y_j for $j = 0{,}1$) to the global mean $\bar{y}$. That leads to the following expression (using K=2 as classification is binary):

$$F = \frac{n_1 \cdot (\overline{y_1} - \bar{y})^2 + n_2 \cdot (\overline{y_2} - \bar{y})^2}{\sum_{i=1}^{N} (y_i - \bar{y})^2 \Big/ N-2} \qquad (7)$$

Finally, we rank features by F-score value. The F-score metric is used to detect how much does variability from classes depend on different features. As an example, we are very interested in features that have the same variability as class labels, which will mean that they do change a lot between patients from different classes. Nonetheless, the class variability might be due to intrinsic subjects' variability instead of difference between classes.

2.2.5 mRMR - Minimum redundancy maximum relevance

The essential aspect of this method is that it assigns scores to features based on redundancy and relevance using two different metrics, in contrast to previous methods, which only used one. It basically implements a solution that tries to maximize the relevance of the set of selected features at the same time that tries to minimize its redundancy. In [2], results show an improvement when applying mRMR to gene microarray classification problems.

Both aspects are expected to be important: in Alzheimer's disease, we know that differences between stages are gathered in some areas rather than being generic. It is also thought to be redundant (that is, equivalent behavior) between several ROIs of the brain, that may be caused, as an example, by sagittal symmetry. Taking redundancy into consideration to compute the score guarantees that the selected features will not only be the most relevant ones, but also as uncorrelated as possible. The key fact about that aspect is that we will rather choose an uncorrelated and medium relevant feature rather than a very relevant but also redundant feature.

mRMR is more exhaustive than previous ranking methods, as it iteratively changes the score from each eligible feature every time a feature is selected, as the correlation between each unselected feature and the set of selected features will be different. The algorithm works as follows:

```
1. Compute F-score between features and labels
2. Compute correlations between features
3. Rank features by F-score
4. Add best feature to selected_features_set
5. while number_features_selected < number_features_wanted
    a. Rank remaining features using F-score and correlation taking into account the selected_features_set
    b. Add best feature to selected_features_set
```

The initialization of the subset of features selected is performed using relevance (F-test) and then we keep adding features to the subsect in a cost function that is a tradeoff between the relevance with respect to the class labels and the redundancy of the subset.

Each metric is computed as follows:

Relevance

- Relevance is measured using and F-score between the features and the labels:

$$F(x,h) = \frac{\sum_k \frac{n_k(\overline{x_k} - \bar{x})^2}{K-1}}{\sigma^2} \qquad (8)$$

Where x represents the feature and h is the label of the feature. Then, the F-score is computed using the information from the class label of the feature as n_k is the number of elements that belong to class k, K is is the number of classes (in our case, 2), $\bar{x}$ is the mean value of feature x and $\overline{x_k}$ the mean value of each class. Finally, σ^2 is the pooled variance, which is computed as shown below (9), where σ_k is the variance of each class:

$$\sigma^2 = \frac{\sum_k (n_k - 1)\sigma_k^2}{n - K} \qquad (9)$$

Then, as what is wanted is the maximum relevance, the most relevant feature will be selected using the following expression (being S the subset of features that we are seeking):

$$\max \mathrm{V_F} \qquad \mathrm{V_F} = \frac{1}{|\mathrm{S}|}\sum_{\mathrm{x}\in\mathrm{S}} F(x,h) \qquad (10)$$

The value of $\mathrm{V_F}$ will be computed for all the possible subsets of features. Those subsets will be formed by the already selected features plus one extra feature from the rest of them.

Redundancy

- Redundancy score is computed by the correlation between features. The correlation coefficient that is used is the Pearson correlation coefficient. However, in comparison to the previous method that also uses that coefficient to compare features and labels, here it is used to compute correlation between features.

$$\rho_{\mathrm{X,Y}} = \frac{\mathrm{cov(X,Y)}}{\sigma_{\mathrm{X}}\sigma_{\mathrm{Y}}} \qquad (11)$$

Therefore, since we want to minimize the redundancy between features, we will proceed to minimize the following expression:

$$\min \mathrm{W}_j \qquad \mathrm{W_j} = \frac{1}{|\mathrm{S}|^2}\sum_{\mathrm{x,y}\in\mathrm{S}} |c(x,y)| \qquad (12)$$

The redundancy for each candidate feature j is computed as a sum of the absolute value of the correlations between this feature and the features in S (the subset of selected features). The absolute value is used assuming that it does not matter whether the features are correlated positive or negatively.

As it is said before, the behavior of this method is slightly different from the methods above shown, as it ranks features one by one using a score based on the ones that have been selected before (while on previous methods we computed the score and ranked the features only once). More concretely, we would like to maximize the relevance of the subset in the ranking, as we want the most relevant feature to be added, and minimize its redundancy with previous features, as a measure to minimize its similarity with the features present in the selected set. As the selection must use both metrics, a final score using relevance and redundancy is computed. Consequently, this method is selecting the best set of m features instead of the best m features.

That final score can be computed in two different ways:

- FCD - F-test correlation difference: $\max\ (\mathrm{V_F} - \mathrm{W_C})$

- FCQ - F-test correlation quotient: $\max\ (\mathrm{V_F}/\mathrm{W_C})$

Finally, in terms of our project, both methods are implemented. There has to be said that the result of an F-test has a range from 0 to $+\infty$ whereas correlation scores are fitted between [0,1] (absolute value is taken). Results seem to vary from one to the other, as the FCD emphasizes the relevance in front of the redundancy, because of the different score order, and the FCQ gives more importance to redundancy.

2.3 Wrapper methods

Wrapper methods are methods that analyze different subsets of variables to choose the best one. Moreover, they also use a classifier during the feature selection, so they usually are computationally expensive. That fact is emphasized when using a LOO strategy, as the computational cost grows as another loop is included inside the workflow.

The main aspect about wrapper methods is that the evaluation of feature subsets is done based on the performance achieved by using them with a determined classifier. Consequently, there are some degrees of freedom to build a wrapper method as wanted:

1. The performance metric that will be assessed
2. The classifier to be used
3. The way subsets will be selected and its size

In order to prevent overfitting, the performance metric will be evaluated on unseen data. That is, we will use cross-validation to split between train and validation data. That points out, even more, the small number of samples that we have. To correct that, as it has been explained before, a leave-one-out algorithm is used to avoid possible deviations.

In this project there are two different types of wrappers that have been implemented: forward selection (using two versions: SFS and SFFS) and L1-selection.

Note: we will call subset the set of features that is going to be assessed. Therefore, we will be assessing many subsets in order to find the best one.

2.3.1 Forward selection

The main idea of this method is very simple: at each step, the feature subset that have obtained better results according to the f1-score in the performance is selected. F1-score is selected to be the metric to analyze because it takes into account recall and precision, which are the indicators of the classification performance that we want to optimize.

Thus, each step will be adding one feature to the subset. Then, our subset will be empty at the beginning of the algorithm and will end having, at most, the number of features requested.

1. Empty subset
2. Select best feature according to f1-score
3. while number of features selected < number of features requested
    a. Try new subset
    b. Update the best subset (if there is an improvement)
4. return the best subset

Forward selection algorithm iteratively tries feature subsets for each number of features possible. The range of the length of possible subsets is [1, number_of_features_requested].

However, the algorithm will finally select the best subset (according to a determined metric) among those that have been tried. That means that the length of best set does not have to be the number of requested features, as a subset with less features might obtain better assessment. In fact, in case of a tie, the smaller subset is selected.

There are two different versions of the algorithm: SFS (Sequential Forward Selection) and SFFS (Sequential Floating point Forward Selection). These two versions of the algorithm are exhaustive, as they try many different subsets of features before choosing the best one. That is another reason that makes this method very computationally demanding (and, depending on the size of the dataset or the classifier, unfeasible).

In these forward selection methods, sometimes there are different subsets which obtain the same f1-score. As a consequence, we looked for a method to solve ties when trying to add a new feature. Finally, ties are solved using Pearson selection. In other words, when a tie occurs, the feature added to the subset is the one that has higher Pearson ranking. The reason to use Pearson is that it has proven to be the most efficient filter method and here we needed a fast and useful metric to solve ties.

2.3.1.1 SFS - Sequential Forward Selection

The behavior of this method is simple: at each step, a new feature is added to the current subset. To select the best features, we build up several feature subsets, formed by the previously selected subset plus each of the candidates (in that case, the remaining features) and perform the classification step to assess its performance. Then, the one with the best f1-score is added to the subset.

```
1. Empty subset
2. Select best feature according to f1-score
3. while number of features selected < number of features requested
    a. Add a new feature to the set
        i. Classify using subsets formed by selected features + new feature
        ii. Select best subset and add new feature
    b. Update the best subset (if there is an improvement)
4. Return best subset
```

There is no doubt that this process is computationally expensive, as the algorithm is classifying for every possible feature to add and, in addition to that, a leave-one-out cross validation is used. That reason motivated the implementation to be done using two different classifiers to select features: Logistic regression and LDA (Linear Discriminant Analysis).

The advantages of using LDA as a classifier is that it has no hyperparameters to optimize in the cross validation stage, so in the inner loop from feature selection stage, no validation is needed.

On the other hand, logistic regression needs to optimize its regularization parameter C that regularizes the classifier, so every feature must be tested with every possible value of C, which makes the computational cost grow. That is the reason why we do not use this classifier here, as after making several experiments we saw that using logistic regression as a classifier was computationally unfeasible.

As a result, only two of the three expected combinations of classifiers will be assessed, as in table 2.

Classifier used to assess performance	Classifier used to select features
LDA	LDA
Logistic Regression	LDA

Table 2: Classifiers used in Forward selection methods.

2.3.1.2 SFFS - Sequential Floating point Forward Selection

The SFFS algorithm is one step forward the SFS. This method incorporates a new stage in the selection process: the possibility to remove a feature from the subset of selected features. The algorithm works as follows:

```
1. Empty subset
2. Select best feature according to f1-score
3. while number of features selected < number of features requested
    a. Add a new feature to the set
        i. Classify using subsets formed by selected features + new feature
        ii. Select best subset and add new feature
    b. Try removing one feature
        i. Classify using a subsets formed by selected features except one.
        ii. Check if f1-score obtained is better than before
        iii. In case there is an improvement, remove the feature
    c. Update the best subset (if there is an improvement)
4. Return best subset
```

It can be seen that the new part is 3.b in the algorithm. That step is used in order to prevent the subset from redundancy, as new features might be redundant with selected ones or even better, so the optimal subset is supposed to be better than when using SFS.

As said before, the classifier and the cross-validation algorithm make the algorithm very slow. Moreover, the fact that an extra loop to try to remove features is added also rises the computational time.

2.3.2 L1 selection

The aim of using L1-norm selection is because of the sparse solutions that we obtain. In other words, it tries to find the smallest subset of features that optimize a certain cost function. Here we use L1-norm as an approximate solution for the optimal L0-norm, because we are dealing with an NP-hard problem as all different combinations of features need to be tried.

In logistic regression, the probability to belong to each class is computed as in (13), where $g(z)$ is a sigmoid function which is applied to a linear combination of independent variables x_i.

$$\hat{p} = g\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{x} + w_0\right) \quad \textit{where} \quad g(z) = \frac{1}{1-e^{-z}} \qquad (13)$$

Among other ways to solve this problem, we will use a classifier regularized by an L1-penalty, following the expression in (14), where $\|W\|_1$ is the norm for a vector that contains the weights to be estimated. The penalty parameter C is the tradeoff between the prediction error and the amount of features in the subset selected.

$$argmin\ \|\boldsymbol{W}\|_1 + C \cdot \|\boldsymbol{y} - g(\boldsymbol{Wx})\|^2 \quad where\ \|\boldsymbol{W}\|_1 = \sum_{i=1}^{m} |w_i| \qquad (14)$$

The restriction on the weights affects the classification. Therefore, features that have weights different from zero are the ones selected. As a result, for a big value of C there will be more weights different from zero (as the weights will not affect the cost function) whereas for small values of C there will be a small amount of features with weighs different from zero. The algorithm followed is simple:

1. Classify using logistic regression with L1 regularization
2. Compute feature importance
3. Select features with importance above zero

The main drawback of this method in order to do the feature selection analysis is that we cannot decide the number of selected features but only the value for the penalization parameter C. They are correlated, as the bigger the penalization parameter, the more features are selected. However, that dependency does not follow a pattern, as the number of selected features might not rise while the parameter does.

2.4 Ensemble methods

Ensemble methods are those that use different classifiers and reach a decision based on all of them. In fact, the final decision is typically reached by majority voting, which means that the most repeated class will be the decided one. As an algorithm, an ensemble method should behave as follows:

1. Create C classifiers
2. Decide the number of features to use in every classifier
3. Classify (each classifier) the input sample
4. Decide the class by majority voting

Because of that, feature importance has to be extracted from classifiers after the classification is done. In our project, Random Forests are implemented as an example of ensemble feature selector.

2.4.1 Random Forests

As its name indicates, a random forest (RF) is formed by a group of decision trees that attempt to classify samples. These trees will be created using a determined number of features randomly selected. After classifying, the most important features will be selected in order to be used posteriorly.

Decision Trees

A decision tree classifier predicts the class of a variable by learning rules inferred from the data features. Decision trees are very useful as they can treat either numerical or categorical data indistinctively.

Theoretically, the process of building a tree is an iterative process that can be described by the following steps:

1. Select the rule that will be applied to the node (following certain criteria)
2. Split the node
3. Decide if the node should be a leaf or not (if it is a leaf, go to the next node)
4. Decide if the tree is complete

The criteria applied to decide the most useful rule to apply at each node can be, for example, entropy impurity ($i_E(N)$) or Gini impurity ($i_G(N)$).

$$i_E(N) = -\sum_j P(\omega_j) \cdot log_2 P(\omega_j) \quad where\ P(\omega_j) = \frac{n_j}{N} \qquad (15)$$

$$i_G(N) = -\sum_j P(\omega_j) \cdot (1 - P(\omega_j)) \quad where\ P(\omega_j) = \frac{n_j}{N} \qquad (16)$$

These expressions will determine which will be the rule (that is, which feature) used in every node that has to be split. The $P(\omega_j)$ is the probability to be in class j in a node, as n_j is the number of elements from class j in a node when taking a certain rule into consideration. The feature used to split the node will be the one with highest value according to previous criteria.

The main drawback of decision trees is that they are very dependent on the data used to build them. That means that little changes on data might mean big changes on the tree structure, which is prone to lead to different classification results. In other words, if a dataset has few samples and a big amount of features, decision trees tend to overfit the data. To avoid the overfitting problem, trees need to be pruned (if they are complete), or, at least, be limited on the depth. That is why we will need to control maximum depth's parameter of the tree to avoid overfitting.

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. Samples used in each tree are drawn with replacement as there is bootstrapping.

Finally, once every tree has reached a decision, the final decision is done by majority voting, which means that the most common class will be the decided one.

Once the classification is finished, there is the key part of the process: selecting the most relevant features. That is made according to the importance that every feature has had in the building of the trees. That feature importance ranking is the key that will lead us to select the more important ones.

3. Results

Different feature selection methods are assessed in order to find out which of them has the best behavior when applied to Alzheimer's disease problems. Assessment and comparison of different methods will be done by making different experiments. These experiments will be focused on three different aspects of each method (if evaluable): order of selected features, variability of selected features among folds and performance of the classifier. Each aspect will be presented using different figures.

- Analysis of the most representative features of all dataset: to clearly show the order of selected features, we are using a list. This list allows us to compare the top features of each method and determine which ones are the most representative. Lists can be found in the annexes. These lists allow us to, as well as comparing methods, see which are the most relevant ROIs, which is useful for
  In addition, we can also plot a feature map (Figure 4), which helps to visualize coincidences in top selected features between two methods.

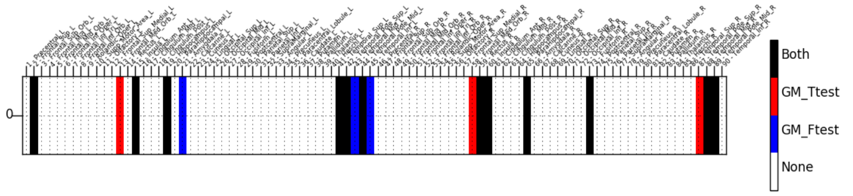


Figure 4: Figure representing coincidences between top 15 features selected by a t-test and a f-test in grey matter volume case

- Variability of selected features among folds: in order to assess the stability of models across each fold of LOO strategy, we will analyze the variability among these different partitions of the database. Those different partitions might convey some variability between the selected features (depending on the partition).
  That aspect is shown using feature maps with different intensity colors that indicate the frequency of each feature to be selected.
- Performance of the classifier: finally, the assessment of the performance is shown by using a graphic that plots performance metrics against the number of features requested to the method. As a result, the optimal number of features can be identified. Nonetheless, there are some methods, such as L1-selection, where the number of features cannot be easily obtained. Then, the plot shows performance metrics against value of a parameter.

The conducted experiments have been done under some assumptions:

- A nested LOO strategy is used to assess the results, as an internal LOO is used to cross-validate the logistic regression hyperparameter.
- Classification is binary, considering pre-clinical (PC) class as the positive one and normal control (NC) class as the negative one.
- The criteria used to evaluate the results from a feature selection method is f1-score.
  The reason is because we are interested in the detection of PC subjects correctly and avoiding false negatives. However, we found that recall was not enough, as classifying all samples as positive gave the best value possible (recall = 1). Therefore, we computed f1-score which also takes into consideration the precision, that also penalizes false positives. If necessary, we will also take accuracy and specificity into account.
- Pearson and Kendall methods have been taken as baseline methods, as they were the ones implemented in the previous classification system.

3.1 Determining optimal parameters

3.1.1 Random forests:

The main obstacle when analyzing the random forest method is the amount of parameters that is has. After analyzing their parameters, we found relevant the number of trees (estimators) taken into consideration in the forest. In addition to that, some other parameters such as the maximum depth of trees and bootstrapping were set by default as recommended in the sklearn library when facing feature selection problems in classification.

As a result, several simulations were carried in order to obtain the optimal number of estimators. To obtain that optimal number, we plotted different performance results with a fixed number of features in order to see where best results are achieved. A different optimal parameter was found for every modality.

Grey matter volume

In case of grey matter volume data, figure 5 shows the performance of the classifier along the different values for the number of estimators. The best performance is obtained when there are 190 estimators. As a consequence, performance analysis will be done using that value (190) for the number of estimators.

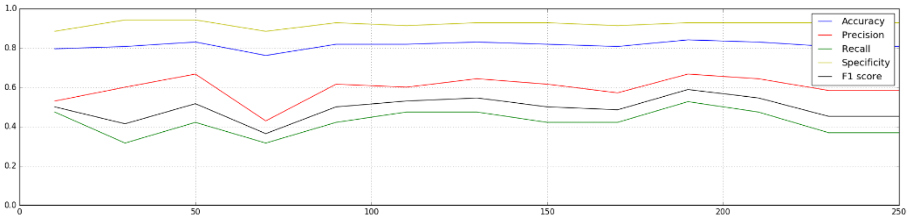


Figure 5: Performance evolution among different number of estimators. Number of selected features is fixed to 15.

Structural connectivity matrices (SCM)

In case of structural matrices, we undertook the same study among the number of estimators. As a result, we obtained that top results correspond to a 130 (Figure 6), as can be seen in examples for 20 features. Therefore, from now on we will be using 130 estimators when dealing with structural matrices’ data.

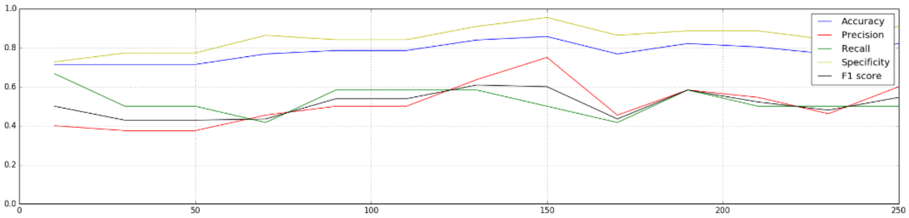


Figure 6: Performance evolution among different number of estimators. Number of selected features is fixed to 20.

3.2 Feature Maps

3.2.1 Analysis of representative features using all database

In grey matter volume, different filter methods mainly obtain similar rankings as can be inferred from lists obtained. There has to be outlined that F-test and Pearson correlation coefficient rank features in the same order, as their values differ only by a factor that does not change the ranking. As a result of that, from now on we will talk about Pearson or F-test indistinctively as they are ranking features in the same order.

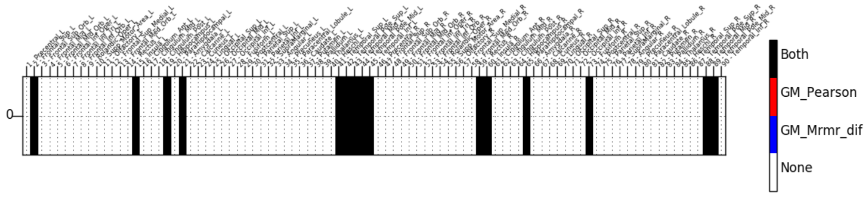


Figure 7: Similarities between Pearson method (red) and mRMR-FCD (blue) on top 15 features in grey matter volume

Another interesting comparison is between Pearson method and mRMR-FCD, to see the effect of redundancy on the order of features. However, as can be seen, the effect of redundancy is imperceptible for this small set of features. For bigger sets the difference between both methods must be more perceptible and, possibly, we aim that mRMR will outperform Pearson selection in terms of classification errors.

However, this effect can be seen in Figure 8, as there are many differences between the two versions of mRMR method: FCD and FDQ.

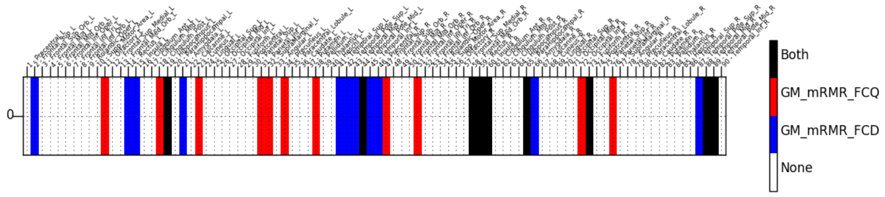


Figure 8: Similarities between mRMR-FCQ (red) and mRMR-FCD (blue) on top 15 features in grey matter volume

When using structural connectivity matrices (SCM), relationship between rankings and methods is barely the same. The only exception is that, as the number of feature increases from 90 to 405 (once the 8100 features are preprocessed), it is easier to obtain different orders and differences are greater than in grey matter volume feature maps.

As can be seen in Figure 9, the differences between the two baseline methods are 5 features out of first 20. That may seem relevant when classifying using SCM. However, if we compare both methods when accounting for grey matter volume, the differences are reduced to one feature out of 15. That points out that feature orders tend to be more different when using SCM (Figure 10). That is probably due to the nature of features, as SCM have more outliers (and false positives) in the data.

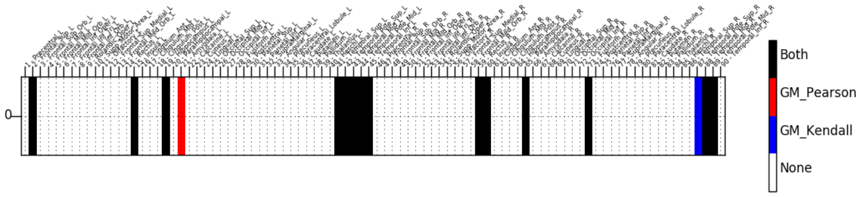


Figure 9: Similarities between Pearson (red) and Kendall (blue) on top 15 features in grey matter volume

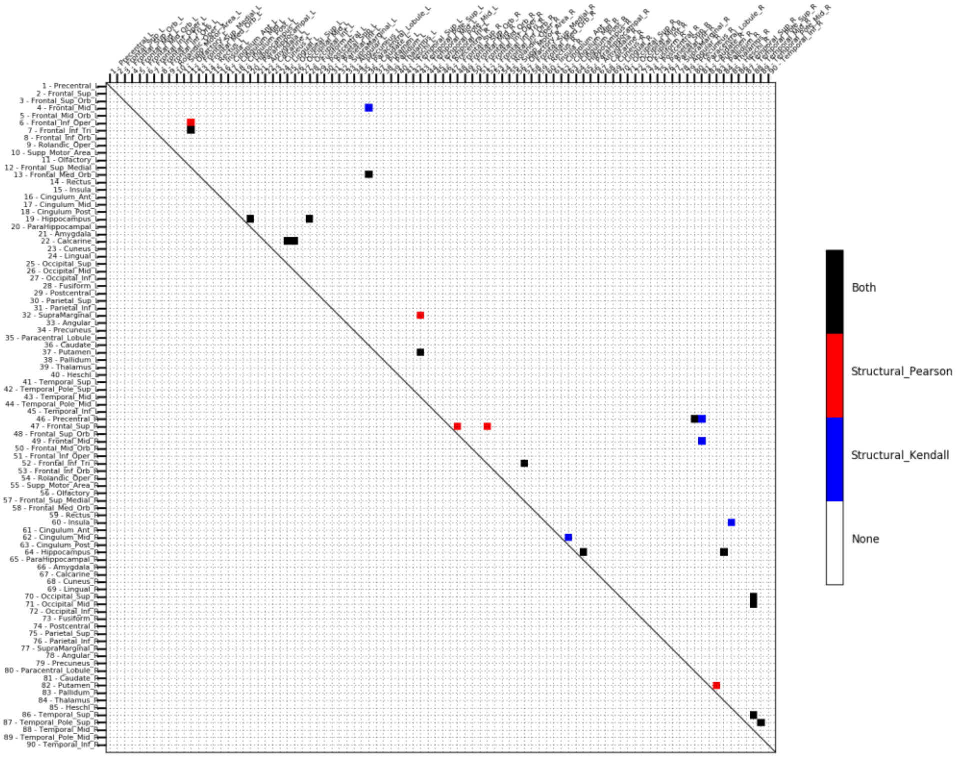


Figure 10: Similarities between mRMR-FCQ (red) and mRMR-FCD (blue) on top 15 features in SCM

In case of forward selection, comparison between SFF and SFFS methods show small differences. For the SFS model, we obtained a list of 17 features and for SFFS, a list of 15 features in the case of grey matter volume out of 90 ROI. Using structural matrices, we find out a similar behavior: the best set contained 20 features (for SFFS) and 18 features (for SFS). These are examples of the fact that the selected set does not need to be as long as requested if there is a better and shorter set.

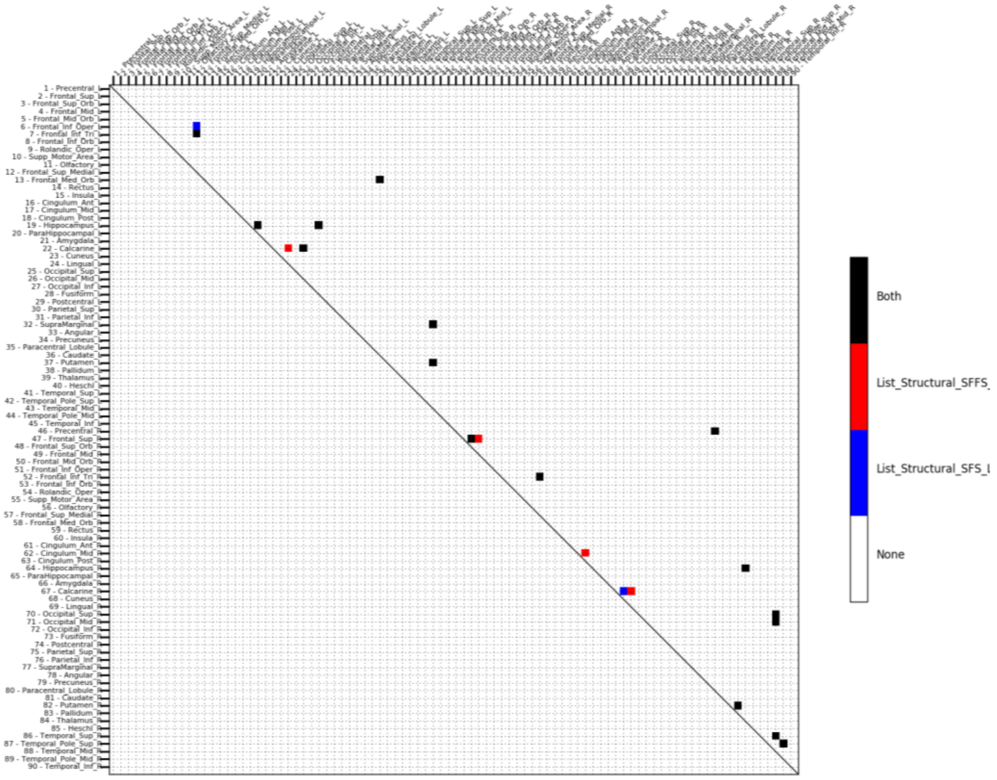


Figure 11: Similarities between SFFS (red) and SFS (blue) on top 20 features in SCM

These little differences are due to the fact that SFFS method can remove an already selected feature from the set, which is expected to optimize even more the selected set.

In L1-selection, as it is a parameter dependent method, selected features might be different depending on the parameter. As a result, this analysis is not as accurate as expected, thus there might be some variability between features along the parameter value. We manually chose a parameter that obtained the desired number of characteristics (15 in grey matter volume case and 20 for SCM).

Therefore, compared to baseline (Pearson method) the results obtained show that there are many differences between methods. That fact makes us expect poor classification results.

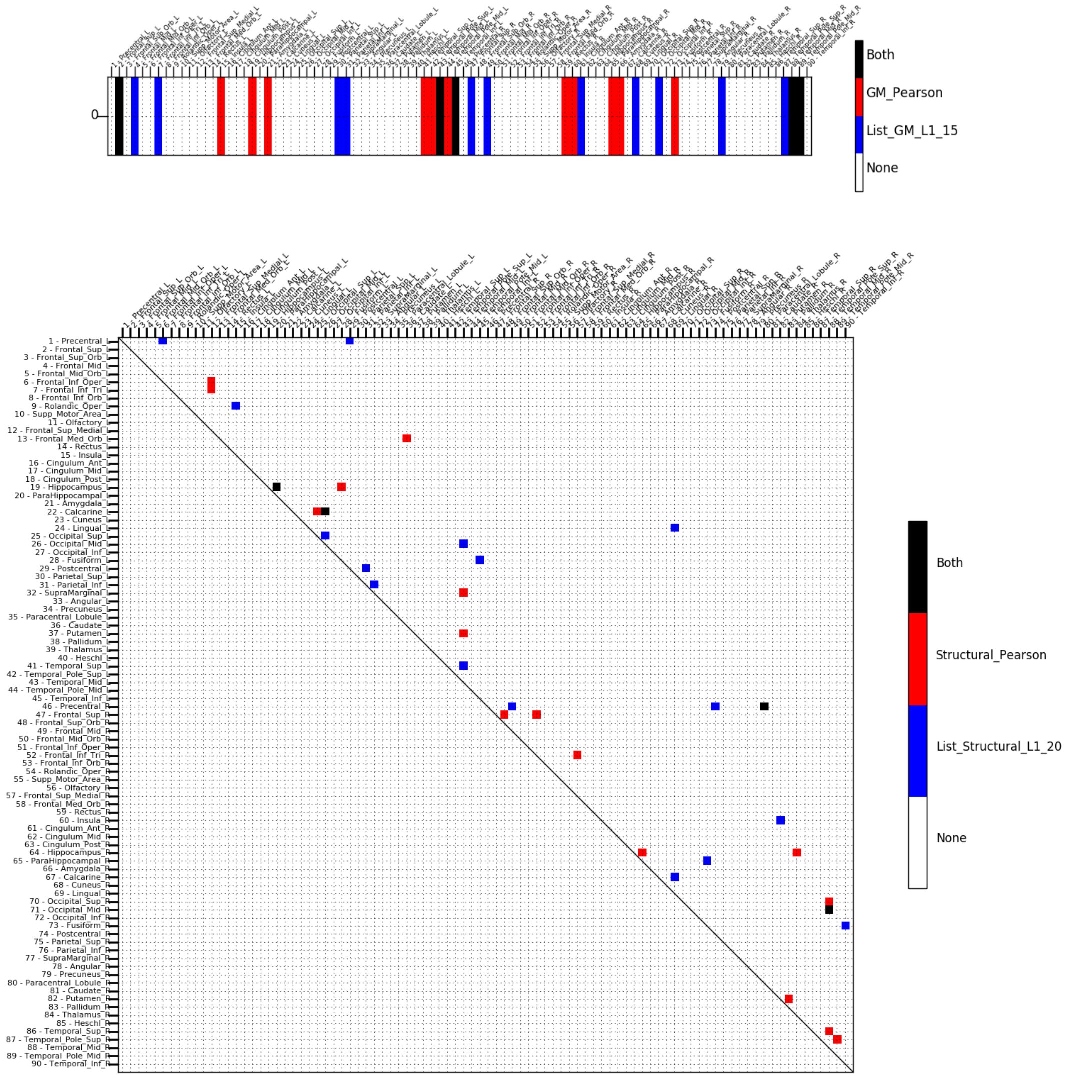


Figure 12: Similarities between Pearson (red) and L1-selection (blue) on top 15 features in grey matter and top 20 in SCM

Finally, ensemble methods show small differences in order when taking grey matter volume data into account. However, these differences increase when analyzing structural matrices.

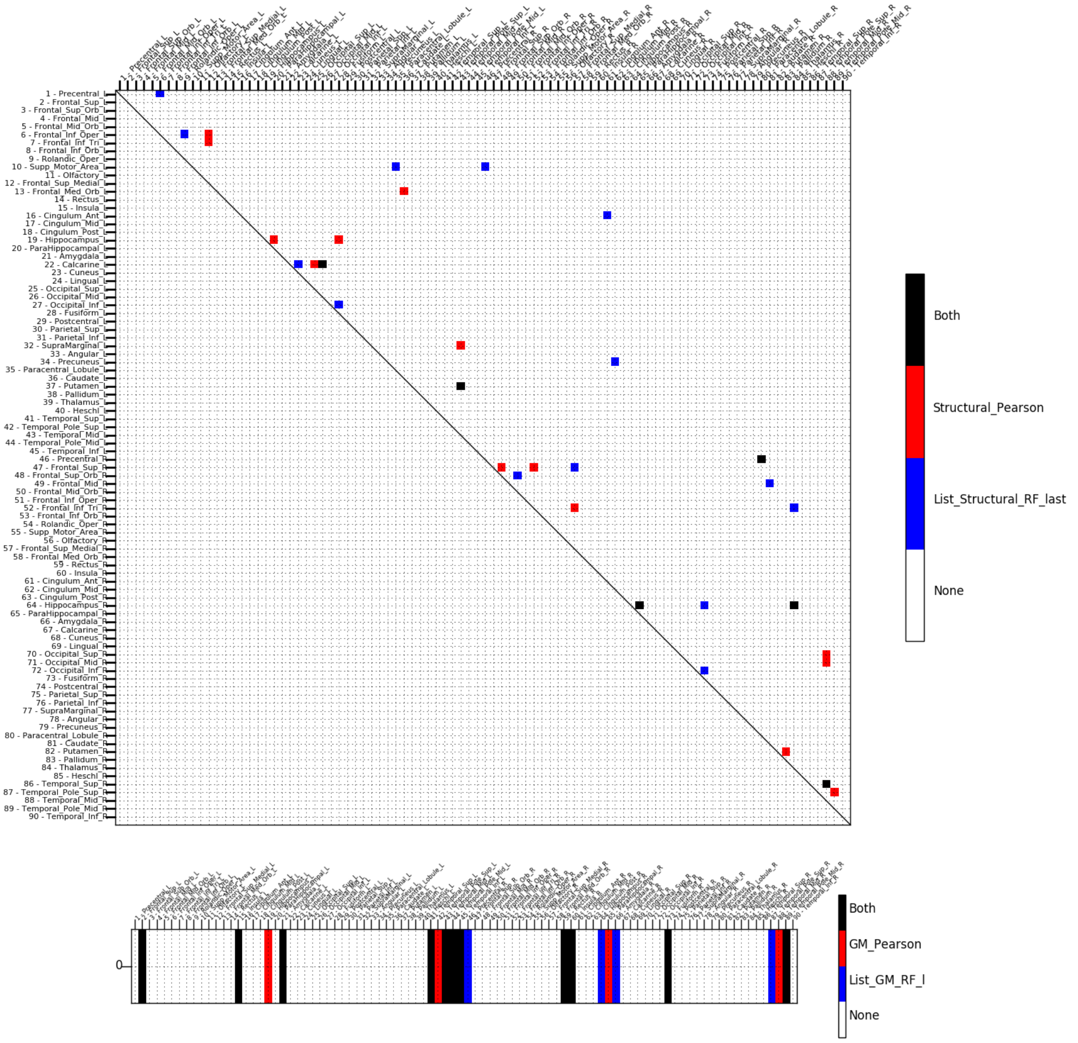


Figure 13: Similarities between Pearson (red) and selection using random forests (blue) on top 15 features in grey matter and top 20 in SCM

Contrasting results with literature:

It is interesting to compare coincidences between best methods, as they determine the most relevant features. In order to assess that, we will show the ranking of 4 of those methods to see the similarities among them. Besides, we will contrast these finding with previous work found in literature. As stated in [1] relevant areas of the brain that are affected are: medial temporal lobe, hippocampus, lateral temporal, parietal and prefrontal. Looking up at the AAL atlas used in our project (appendix B), we outlined the coincidences between similar areas in dark orange in the table. We can see that coincidences are in temporal lobes and hippocampus. However, we indicate as relevant areas such as rectus, that do not appear in [1].

Pearson	mRMR-FCD	SFFS	RF
43	43	43	43
87	87	58	65
59	59	64	42
64	64	41	40
58	58	1	72
18	18	88	59
42	42	14	63
88	88	44	88
14	14	13	86
41	41	85	45
40	1	87	14
1	40	59	20

Table 3: Top features selected by different methods

In case of structural connectivity matrices, we can see that there are many coincidences with previous findings [8] that have determined which are the relevant areas of the brain that help to discriminate between NC and PC classes. In fact, the areas found are only 23, thus a set bigger than 23 features is not expected to obtain better results.

Pearson		Kendall		mRMR-FCD		RF	
12	35	12	35	5	11	63	83
5	11	45	79	12	35	36	42
85	87	51	56	85	87	21	25
51	56	18	19	51	56	16	61
18	19	18	27	18	19	85	87
18	27	48	80	18	27	63	72
63	83	63	83	63	83	86	88
6	11	70	87	6	11	15	60
70	87	85	87	70	87	45	79
86	88	21	24	86	88	63	64
21	25	6	11	21	25	51	83
63	64	21	25	63	64	59	84
45	79	45	80	45	79	47	49
69	87	86	88	36	42	18	27
36	42	59	84	69	87	21	22
81	82	69	87	81	82	46	47
31	42	3	35	31	42	18	19
46	47	63	64	46	47	26	27
46	51	61	62	46	51	9	34
21	24	36	42	21	24	46	51

Table 4: Top features selected by different methods

Although it may seem that there are many differences, there has to be outlined that 13 out of first 20 features coincide for Pearson's coefficient ranking. This fact confirms that the features selected by Pearson's coefficient are also relevant features according to clinical literature.

3.2.2 Analysis of variability between partitions

A priori, differences between different training partitions are expected to be irrelevant as LOO strategy makes the maximum difference be of a sample (the one that is used in test). In general, we expect little variability such as in Pearson method (Figure 14), where there in grey matter volume case there is no variability will mean that method is stable among different partitions. Figures show the variability in the top 15 features for grey matter volume and in the top 20 for structural connectivity matrices. Using a higher number will show more variability. However, the differences will be caused by irrelevant features and not the top ones, so it is not an interesting case.

Nonetheless, there is a method in which that variability between partitions is relevant: mRMR FCQ. The way that redundancy is affecting the mRMR score makes little differences bigger, as the redundancy score is in the denominator.

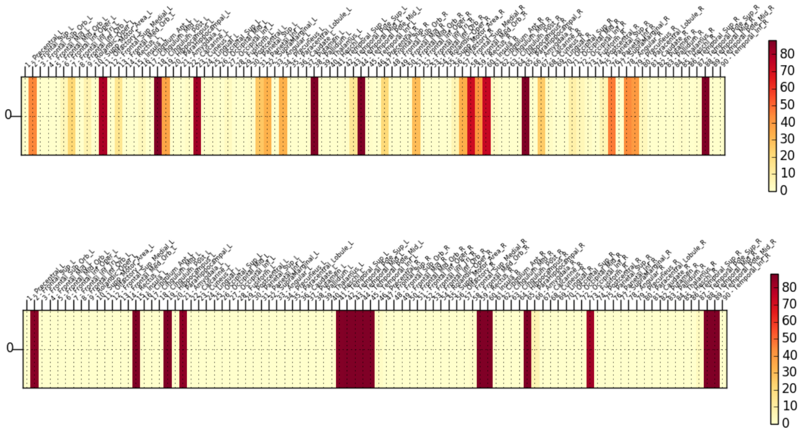


Figure 14: Variability across folds in grey matter volume. mRMR FCQ (above) and Pearson (below).

In the case of Forward selection methods, variability is bigger than in case of filter methods (Figures 15 and 16). That is due to the fact that the feature selection algorithm is very dependent on the samples, as the best subset is found by an exhaustive and iterative process among them. As a result, we can infer that models are unstable and prone to overfit, as they are adapted to training data. That aspect will be problematic when assessing the performance metrics of the classifier.

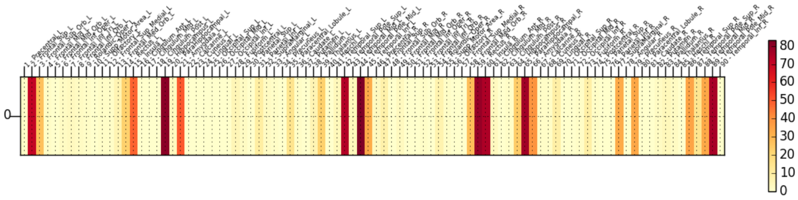


Figure 15: Variability across folds of SFS method in grey matter volume

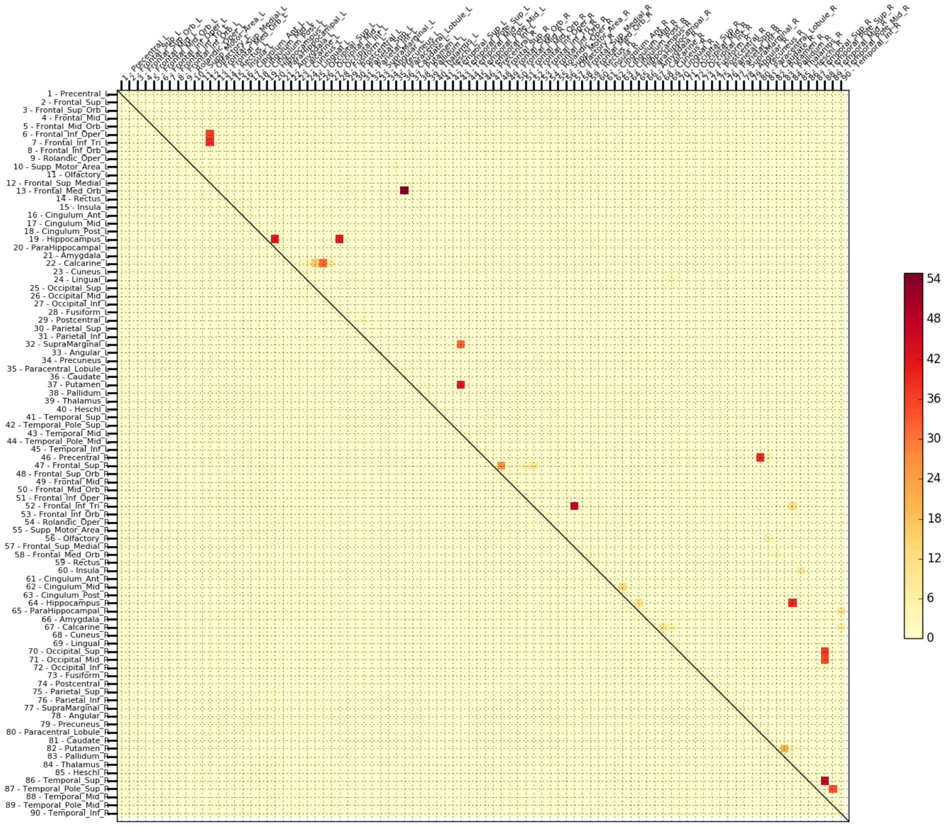


Figure 16: Variability across folds of SFFS method in SCM

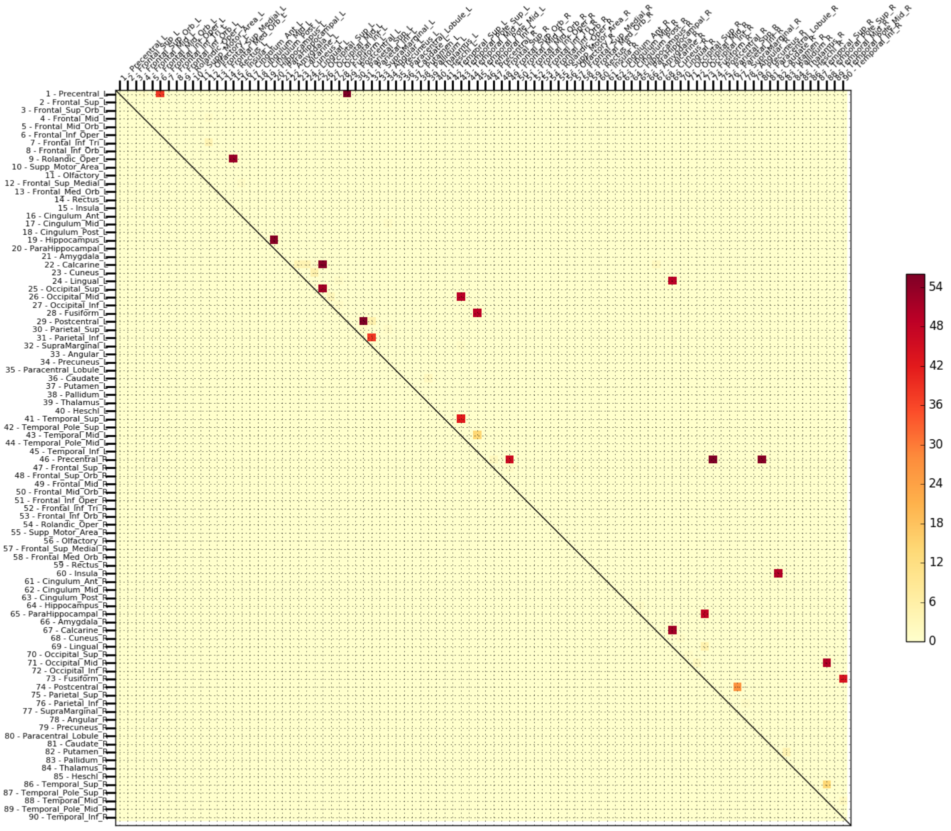


Figure 17: Variability across folds of L1 selection method in SCM

When using L1 selection method, we can see that this method is stable among different folds, as can be seen in Figure 17. However, this stability is not compensated with the fact that L1-norm selection selects different features than other methods that have better performance (as we will see in posterior sections).

Finally, as expected, random forests have a some variability among folds. That is due to the fact that trees are very dependent on training data. To avoid that fact, which may lead to undesired overfitting, we use a great number of estimators. In fact, there can be seen that, as an example, in grey matter volume feature map there is remarkable variability, but we can clearly see that up to 9 features are selected in almost all folds among the top 15 we are analyzing. Behavior in structural connectivity matrices is also similar.

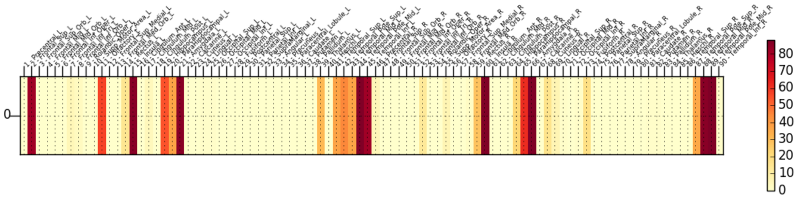


Figure 18: Variability across folds of random forests in grey matter volume

To sum up, we observe that there are some methods that are more stable across different folds (such as Pearson), which makes them reccomended for small datasets. However, those like mRMR-FCQ, tant has high variability, won't be reccomended in an scenario where we have a small database or that database is has a lot of variability among subjects.

3.3 Performance assessment

Finally, the easiest way to compare different methods is assessing their classification results. These results are presented in a figure that shows the evolution of the different metrics along the number of selected features.

The results are usually presented in two separate graphics (Figure 19): one corresponding to the evaluation using test partition and the other using again the train partition. This last figure helps us to determine if we are in an overfitting situation, where all metrics will tend to 1.

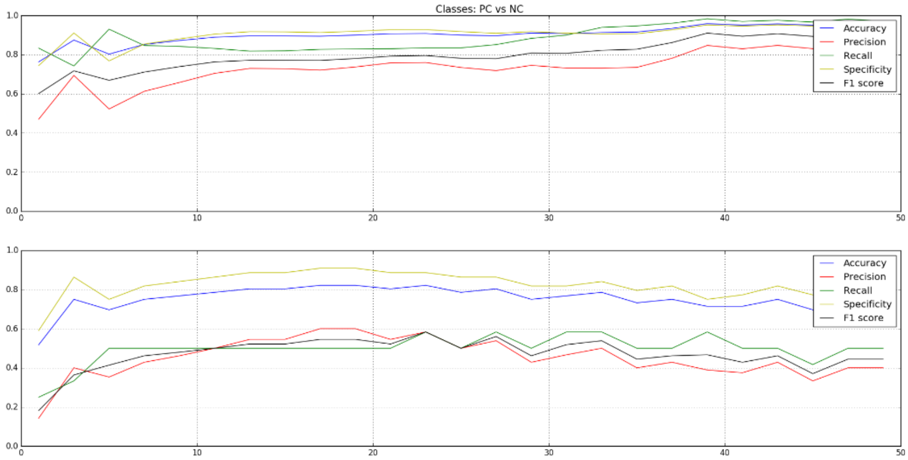


Figure 19: Performance results when selecting with T-test in SCM

Nevertheless, in this comparison we will focus only on test results and will only show train results if relevant. Before the start of the project, Pearson coefficient method and Kendall tau method were already implemented. Consequently, we will be using their results as baseline.

3.3.1 Grey matter volume

The baseline results are shown in the Figure 20. As can be seen, the best values obtained that correspond to f1-score are for Pearson method between 10 and 15 features. The value of f1-score obtained is close to 0.56.

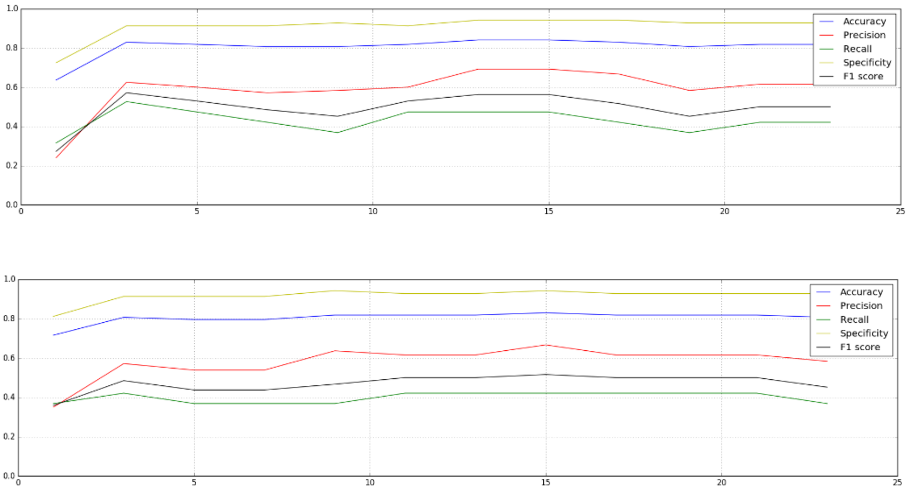


Figure 20: Performance results when selecting with Pearson (above) and Kendall (below) using grey matter volume (GMV) data

When compared to this baseline, the feature selection methods that have obtained better results are SFS and SFS. From filter methods, mRMR – FCD shows little improvement from baseline results. That means that considering the redundancy when selecting features is not as helpful as expected or, at least, when considered using the correlation as in mRMR.

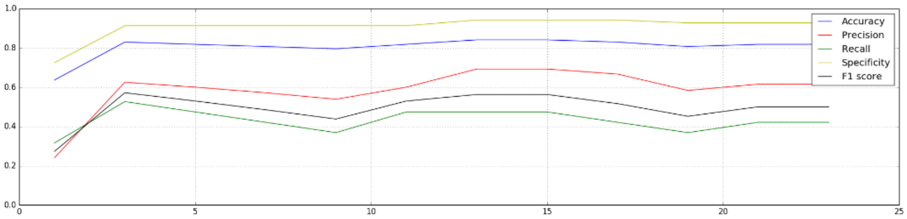


Figure 21: Performance results when selecting with mRMR-FCD in GMV

SFS and SFFS present better results. The value of f1-score may be similar, but the important fact about these results is that recall (green line in figures) is above 0.5, which is much better than baseline. That result makes forward selection methods the best ones to select among grey matter volume features.

In fact, as it has been seen in feature maps, differences between SFS and SFFS are almost imperceptible if looking at Figure 22, where it can be seen that both graphics are almost the same.

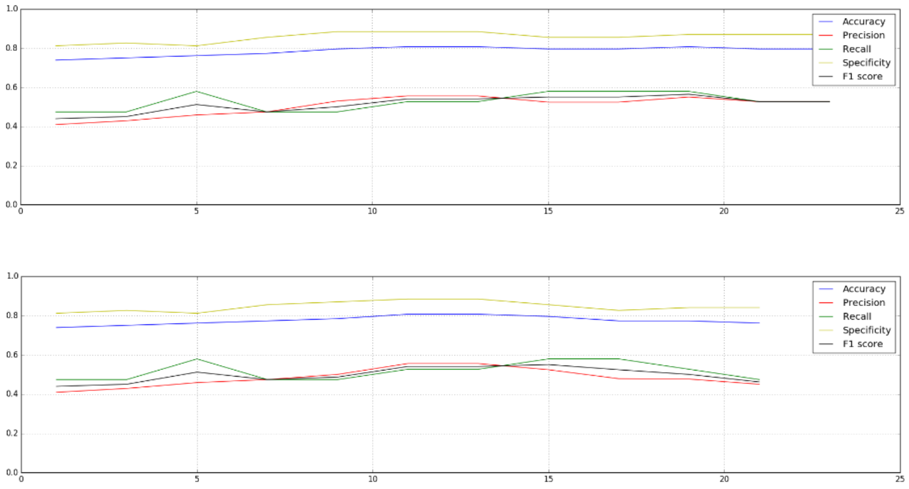


Figure 22:Performance results when selecting with SFS (above) and SFFSl (below) classifying with LR using (GMV) data

However, where difference can be seen is between forward selection methods that use LDA to classify and those that use logistic regression. If we compare figures 22 and 23, we can see that the performance using LR is much better than using LDA. That behavior is what we expected, as LR is more complex than LDA, as it has a hyperparameter that is optimized throughout the system whereas LDA does not have any.

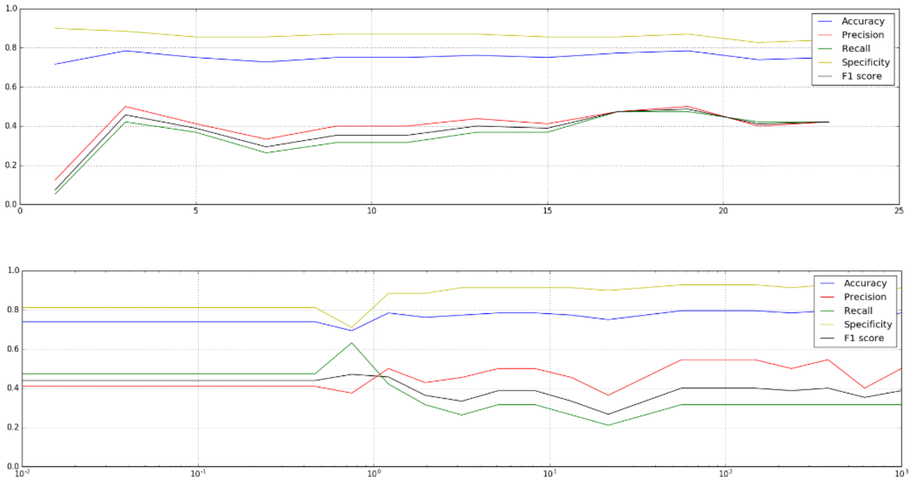


Figure 23:Performance results when selecting with SFS (above) and classifying with LDA against the number of features and L1-selection (below) against penalization parameter value

In case of L1, we have seen in the previous section that only two features were selected in the best case, so the performance assessment is also poor. In fact, when looking into graphics (Figure 23) it may seem that f1-score is quite high, but the low accuracy peak that coincides with that best value (below 0.8) shows that the results are not that reliable.

In case of using a random forest method, the results are obtained using the optimal parameters as explained before. As can be seen in the figure, the results improve the baseline (Pearson). In fact, f1-score almost reaches 0.6.

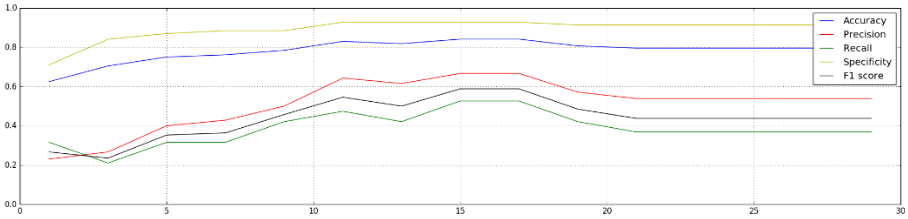


Figure 24: Performance results when selecting with random forest

To sum up, we have found that random forest is the method that best perform when using grey matter volume data from subjects in HCB database. In addition, we can also conclude that out of 90 ROIs, taking into account a number between 15 and 20 ROIs to classify is enough. That fact outlines the importance of these relevant ROIs to distinguish between NC and PC stages of Alzheimer's disease.

3.4.1 Structural connectivity matrices

The baseline results are shown in Figure 25. As can be seen, the best values obtained for Pearson method are, between 15 and 20 features, a f1-score obtained close to 0.7. For Kendall method, between 20 and 25 features with an f1-score close to 0.6. There has to be outlined that Kendall method obtains such good results in 33 and 35 features, but it seems to be an alone peak, which is unreliable.

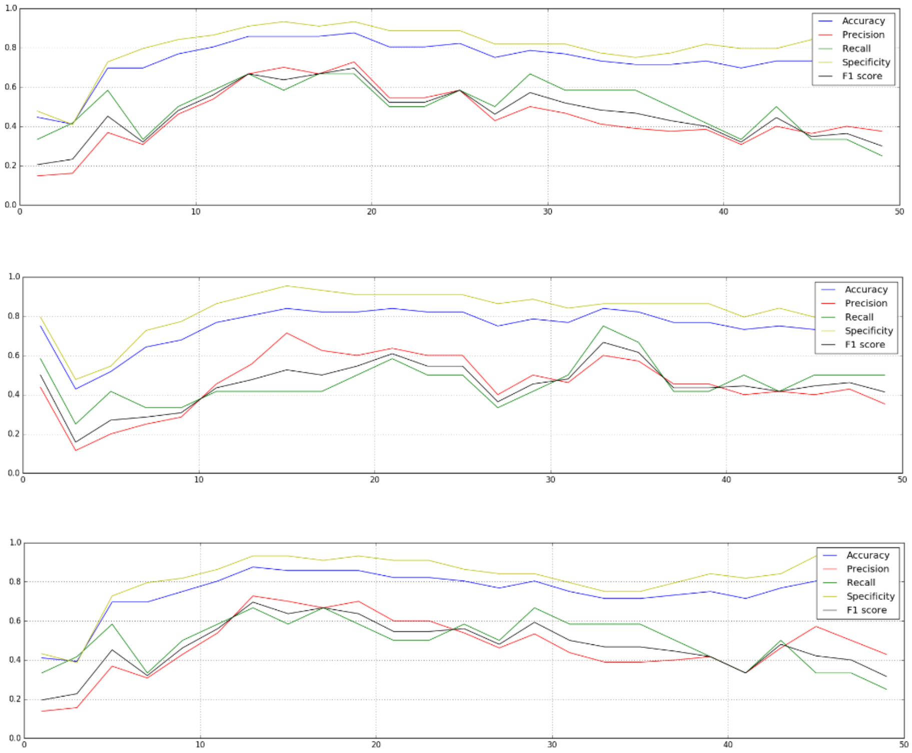


Figure 25: Performance results when selecting with Pearson (above), Kendall (middle) and mRMR-FCD (below)

As expected, mRMR-FCD method (Figure 25) slightly improves the behavior from Pearson method, but not as much as desired. Other filter methods do not improve the performance but obtain similar results.

However, the unexpected fact about structural connectivity matrices is that SFS and SFFS do not improve the baseline methods, as can be seen in figure 26. That can be due to the fact that there are lots of features and these exhaustive methods really have issues when dealing to high dimensional problems.

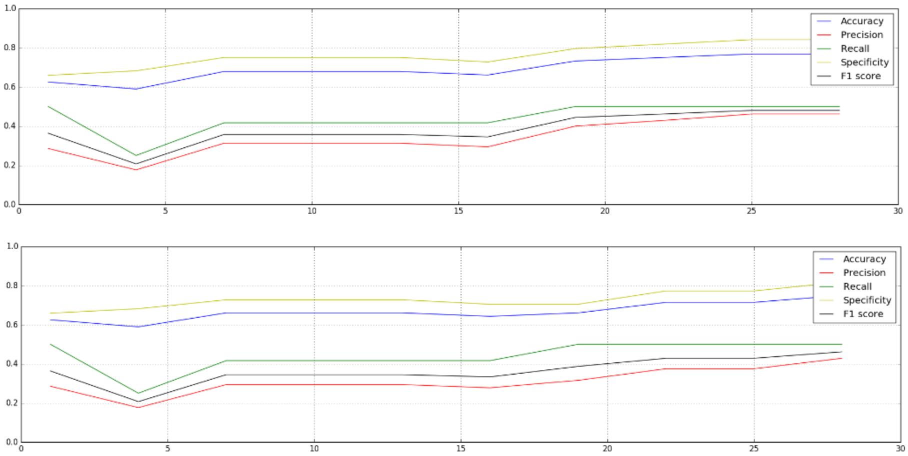


Figure 26: Performance results when selecting with SFS (above) and SFFS (below) classifying with LR

In the case of L1, we can see that across different values of the method the model tends to quickly overfit the data. The sweep of the penalization parameter can be more specific and lead to further investigation, but the results do not show any improvement compared to baseline, as f1-score does not get close to 0.6.

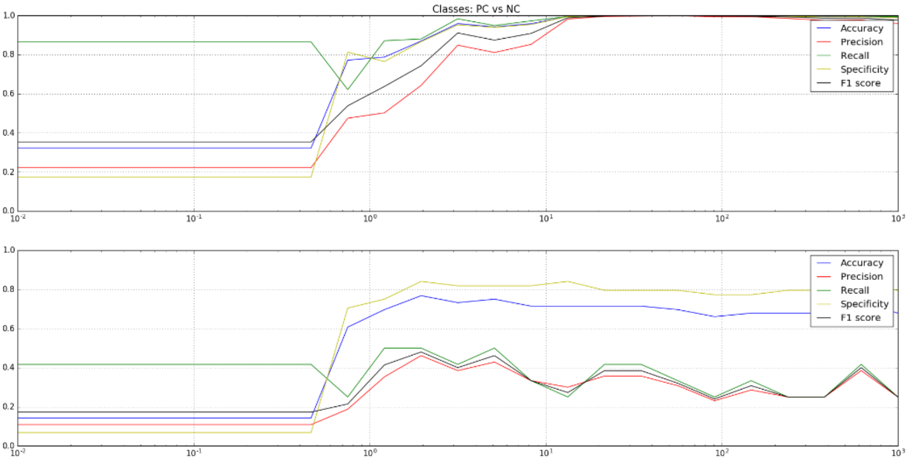


Figure 27: Performance results when selecting with L1-selection method. Train performance (above) and test performance (below) are against penalization parameter value

Finally, random forests do obtain quite good scores compared to baseline, as an f1-score above 0.6 is obtained. However, the key fact is that this value corresponds to a higher number of features whereas Pearson method obtained same results with half of them. In fact, the obtained f1-score value is also lower.

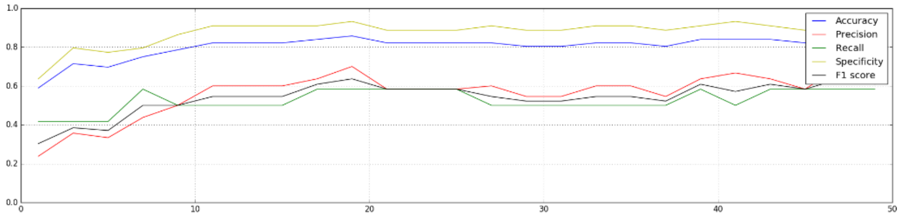


Figure 28: Performance results when selecting with random forests

To sum up, we can conclude that Pearson correlation coefficient was indeed a very smart choice to select features from structural connectivity matrices according to HCB database. In fact, mRMR – FCD method obtain similar results as the order they select features is almost the same. However, we select Pearson method as it is simpler.

Finally, as an overall assessment, we can conclude that filter methods are those who need less computation time as well as obtain better results. In addition, Pearson method is also stable thorough folds. In grey matter volumes, we have less outliers, that is the reason why other methods that had variability between folds obtain better results (e.g: SFFS).

4. Conclusions and future development

In this thesis we have presented different kinds of feature selection algorithms that have been implemented in a classification system that deals with Alzheimer's prediction problems. According to the HCB database, the result of the comparison among different methods has led to the identification of the most relevant ROIs of the brain when discriminating between normal (NC) and pre-clinical (PC) subjects. Several results are shown using our database (HCB), which have been compared and contrasted with ground truth based on previous findings in the field.

In addition to that, this deeper study among different possibilities for feature selection has allowed the implementation of methods that have achieved better results. This improvement is significant as we are working with a small dataset. In fact, this work gathers a set of feature selection techniques that can be used in biomedical applications, analyzing its characteristics and its appropriateness to our problem.

We are constrained by two factors: we cannot use many features, since our dataset is small and we cannot use small subset of features since they are noisy and low-informative. Future work can focus on how to combine those different features to create more informative features.

Moreover, as a possible and future development, these methods might be tested and compared using different kinds of classifiers such as Support Vector Machines (SVM) or K-nearest neighbors (KNN). Another aspect that can be considered is to increase the number of samples, which will help to obtain more robust results. In fact, Pasqual Maragall Foundation (FPM) is carrying a study, known as Alpha study, in which up to 2500 volunteers participate, either healthy or affected by the Alzheimer's disease. These volunteers are invited to be taken under study every three years, which will allow the dataset to increase in addition to the fact that we will be able to see differences among time as well as between stages.

5. Bibliography

[1] Ewers, Michael, et al. "Neuroimaging markers for the prediction and early diagnosis of Alzheimer's disease dementia." *Trends in neurosciences* 34.8 (2011): 430-442.

[2] Peng, Hanchuan, Fuhui Long, and Chris Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy." *IEEE Transactions on pattern analysis and machine intelligence*27.8 (2005): 1226-1238.

[3] Plant, Claudia, et al. "Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease." *Neuroimage* 50.1 (2010): 162-174.

[4] Saeys, Yvan, Iñaki Inza, and Pedro Larrañaga. "A review of feature selection techniques in bioinformatics." *bioinformatics* 23.19 (2007): 2507-2517.

[5] Shao, Junming, et al. "Prediction of Alzheimer's disease using individual structural connectivity networks." *Neurobiology of aging* 33.12 (2012): 2756-2765.

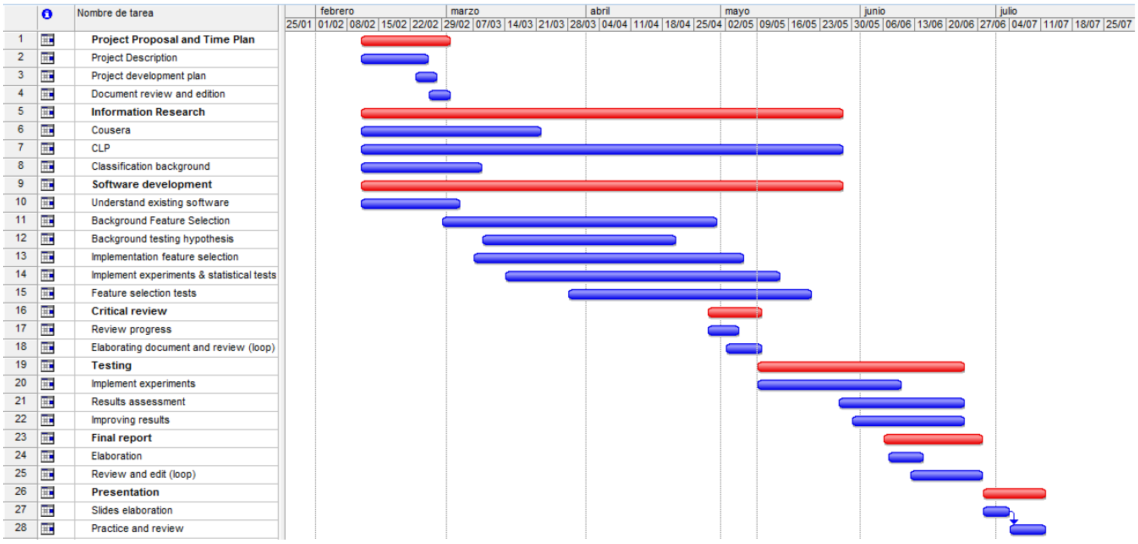
[6] Sperling, Reisa A., et al. "Toward defining the preclinical stages of Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease."*Alzheimer's & dementia* 7.3 (2011): 280-292.

[7] Panizo, Eva. Técnicas de clasificación para el diagnóstico precoz de la enfermedad de Alzheimer. (PFC)

[8] *Tucholka, Alan. Structural connectivity alterations along the AD continuum: reproducibility across two independent samples and correlation with CSF Aβ and p-tau. (to be published)*

APPENDICES

A. Gantt diagram



	Nombre de tarea
1	**Project Proposal and Time Plan**
2	Project Description
3	Project development plan
4	Document review and edition
5	**Information Research**
6	Cousera
7	CLP
8	Classification background
9	**Software development**
10	Understand existing software
11	Background Feature Selection
12	Background testing hypothesis
13	Implementation feature selection
14	Implement experiments & statistical tests
15	Feature selection tests
16	**Critical review**
17	Review progress
18	Elaborating document and review (loop)
19	**Testing**
20	Implement experiments
21	Results assessment
22	Improving results
23	**Final report**
24	Elaboration
25	Review and edit (loop)
26	**Presentation**
27	Slides elaboration
28	Practice and review

B. AAL tractography labels

Feature	Name
1	Precentral_L
2	Frontal_Sup_L
3	Frontal_Sup_Orb_L
4	Frontal_Mid_L
5	Frontal_Mid_Orb_L
6	Frontal_Inf_Oper_L
7	Frontal_Inf_Tri_L
8	Frontal_Inf_Orb_L
9	Rolandic_Oper_L
10	Supp_Motor_Area_L
11	Olfactory_L
12	Frontal_Sup_Medial_L
13	Frontal_Med_Orb_L
14	Rectus_L
15	Insula_L
16	Cingulum_Ant_L
17	Cingulum_Mid_L
18	Cingulum_Post_L
19	Hippocampus_L
20	ParaHippocampal_L
21	Amygdala_L
22	Calcarine_L
23	Cuneus_L
24	Lingual_L
25	Occipital_Sup_L
26	Occipital_Mid_L
27	Occipital_Inf_L
28	Fusiform_L
29	Postcentral_L
30	Parietal_Sup_L
31	Parietal_Inf_L
32	SupraMarginal_L
33	Angular_L
34	Precuneus_L
35	Paracentral_Lobule_L
36	Caudate_L
37	Putamen_L

38	Pallidum_L
39	Thalamus_L
40	Heschl_L
41	Temporal_Sup_L
42	Temporal_Pole_Sup_L
43	Temporal_Mid_L
44	Temporal_Pole_Mid_L
45	Temporal_Inf_L
46	Precentral_R
47	Frontal_Sup_R
48	Frontal_Sup_Orb_R
49	Frontal_Mid_R
50	Frontal_Mid_Orb_R
51	Frontal_Inf_Oper_R
52	Frontal_Inf_Tri_R
53	Frontal_Inf_Orb_R
54	Rolandic_Oper_R
55	Supp_Motor_Area_R
56	Olfactory_R
57	Frontal_Sup_Medial_R
58	Frontal_Med_Orb_R
59	Rectus_R
60	Insula_R
61	Cingulum_Ant_R
62	Cingulum_Mid_R
63	Cingulum_Post_R
64	Hippocampus_R
65	ParaHippocampal_R
66	Amygdala_R
67	Calcarine_R
68	Cuneus_R
69	Lingual_R
70	Occipital_Sup_R
71	Occipital_Mid_R
72	Occipital_Inf_R
73	Fusiform_R
74	Postcentral_R
75	Parietal_Sup_R
76	Parietal_Inf_R
77	SupraMarginal_R
78	Angular_R
79	Precuneus_R
80	Paracentral_Lobule_R
81	Caudate_R
82	Putamen_R
83	Pallidum_R
84	Thalamus_R
85	Heschl_R
86	Temporal_Sup_R
87	Temporal_Pole_Sup_R
88	Temporal_Mid_R
89	Temporal_Pole_Mid_R
90	Temporal_Inf_R