Universidade do Minho
Escola de Engenharia
Departamento de Informática

Master Course in Computing Engineering

Christopher Borges Costa

Development of an integrated computational platform for metabolomics data analysis and knowledge extraction

Master dissertation

Supervised by: Miguel Francisco de Almeida Pereira da Rocha

Co-supervised by: Marcelo Maraschin, Universidade Federal Santa Catarina, Brasil

Braga, December 12, 2014

ACKNOWLEDGEMENTS

Queria agradecer ao meu orientador e co-orientador, o Professor Miguel Francisco de Almeida Pereira da Rocha e o Professor Marcelo Maraschin, pela ajuda e apoio que deram para a realização desta dissertação.

Agradeço também a todas as pessoas que de alguma forma contribuíram para a realização deste projecto.

ABSTRACT

In the last few years, biological and biomedical research has been generating a large amount of quantitative data, given the surge of high-throughput techniques that are able to quantify different types of molecules in the cell. While transcriptomics and proteomics, which measure gene expression and amounts of proteins respectively, are the most mature, metabolomics, the quantification of small compounds, has been emerging in the last years as an advantageous alternative in many applications.

As it happens with other omics data, metabolomics brings important challenges regarding the capability of extracting relevant knowledge from typically large amounts of data. To respond to these challenges, an integrated computational platform for metabolomics data analysis and knowledge extraction was created to facilitate the use of several methods of visualization, data analysis and data mining.

In the first stage of the project, a state of the art analysis was conducted to assess the existing methods and computational tools in the field and what was missing or was difficult to use for a common user without computational expertise. This step helped to figure out which strategies to adopt and the main functionalities which were important to develop in the software. As a supporting framework, R was chosen given the easiness of creating and documenting data analysis scripts and the possibility of developing new packages adding new functions, while taking advantage of the numerous resources created by the vibrant R community.

So, the next step was to develop an R package with an integrated set of functions that would allow to conduct a metabolomics data analysis pipeline, with reduced effort, allowing to explore the data, apply different data analysis methods and visualize their results, in this way supporting the extraction of relevant knowledge from metabolomics data.

Regarding data analysis, the package includes functions for data loading from different formats and pre-processing, as well as different methods for univariate and multivariate data analysis, including *t*-tests, analysis of variance, correlations, principal component analysis and clustering. Also, it includes a large set of methods for machine learning with distinct models for classification and regression, as well as feature selection methods. The package supports the analysis of metabolomics data from infrared, ultra violet visible and nuclear magnetic resonance spectroscopies.

The package has been validated on real examples, considering three case studies, including the analysis of data from natural products including bees propolis and cassava, as well as metabolomics data from cancer patients. Each of these data were analyzed using the developed package with different pipelines of analysis and HTML reports that include both analysis scripts and their results, were generated using the documentation features provided by the package.

RESUMO

Nos últimos anos, a investigação biológica e biomédica tem gerado um grande número de dados quantitativos, devido ao aparecimento de técnicas de alta capacidade que permitem quantificar diferentes tipos de moléculas na célula. Enquanto a transcriptómica e a proteómica, que medem a expressão genética e quantidade de proteínas respectivamente, estão mais desenvolvidas, a metabolómica, que tem por definição a quantificação de pequenos compostos, tem emergido nestes últimos anos como uma alternativa vantajosa em muitas aplicações.

Como acontece com outros dados ómicos, a metabolómica traz importantes desafios em relação à capacidade de extracção de conhecimento relevante de uma grande quantidade de dados tipicamente. Para responder a esses desafios, uma plataforma computacional integrada para a análise de dados de metabolómica e extracção de informação foi criada para facilitar o uso de diversos métodos de visualização, análise de dados e mineração de dados.

Na primeira fase do projecto, foi efectuado um levantamento do estado da arte para avaliar os métodos e ferramentas computacionais existentes na área e o que estava em falta ou difícil de usar para um utilizador comum sem conhecimentos de informática. Esta fase ajudou a esclarecer que estratégias adoptar e as principais funcionalidades que fossem importantes para desenvolver no software. Como uma plataforma de apoio, o R foi escolhido pela sua facilidade de criação e documentar scripts de análise de dados e a possibilidade de novos pacotes adicionarem novas funcionalidades, enquanto se tira vantagem dos inúmeros recursos criados pela vibrante comunidade do R.

Assim, o próximo passo foi o desenvolvimento do pacote do R com um conjunto integrado de funções que permitem conduzir um pipeline de análise de dados, com reduzido esforço, permitindo explorar os dados, aplicar diferentes métodos de análise de dados e visualizar os seus resultados, desta maneira suportando a extracção de conhecimento relevante de dados de metabolómica.

Em relação à análise de dados, o pacote inclui funções para o carregamento dos dados de diversos formatos e para pré-processamento, assim como diferentes métodos para a análise univariada e multivariada dos dados, incluindo *t*-tests, análise de variância, correlações, análise de componentes principais e agrupamentos. Também inclui um grande conjunto de métodos para aprendizagem automática com modelos distintos para classificação ou regressão, assim como métodos de selecção de atributos. Este pacote suporta a análise de dados de metabolómica de espectroscopia de infravermelhos, ultra violeta visível e ressonância nuclear magnética.

O pacote foi validado com exemplos reais, considerando três casos de estudo, incluindo a análise dos dados de produtos naturais como a própolis e a mandioca, assim como dados de metabolómica de pacientes com cancro. Cada um desses dados foi analisado usando o pacote desenvolvido com

diferentes pipelines de análise e relatórios HTML que incluem ambos scripts de análise e os seus resultados, foram gerados usando as funcionalidades documentadas fornecidas pelo pacote.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF FORMULAS

ACRONYMS

ANN	Artificial Neural Network. 18, 19
ANOVA	Analysis of Variance. 15, 16, 33, 34, 43, 51, 52, 61
BATMAN	Bayesian AuTomated Metabolite Analyser for NMR spectra. 13
COW	Correlation Optimized Warping. 13
CSV	Comma Separated Values. 28
CT	Computed Tomography. 55
DA	Discriminant Analysis. 9
DXA	Dual energy X-ray Absortiometry. 55
FC	Fold Change. 16, 56
FDR	False Discovery Rate. 16
FTIR	Fourier-Transform Infrared. 8, 20
GA	Genetic Algorithms. 20, 65
HCA	Hierarchical Cluster Analysis. 16, 22, 36, 37
HLF	High-Level Fusion. 22
HRMAS	High Resolution Magic Angle Spinning. 7
HSD	Honest Significance Difference. 15, 16, 34, 43, 51, 52, 61
IDE	Integrated Development Environment. 26
ILF	Intermediate-Level Fusion. 22, 23
iPLS	interval Partial Least Squares. 20
IR	Infrared. 4, 11, 13, 18, 22, 23
kNN	k-Nearest Neighbors. 18, 32
LDA	Linear Discriminant Analysis. 18

Acronyms

LLF	Low-Level Fusion. 22, 23
LR	Linear Regression. 9
LV	Latent Variables. 18
MCMC	Markov Chain Monte Carlo. 14
MRI	Magnetic Resonance Imaging. 55
mRNA	Messenger Ribonucleic Acid. 3
MS	Mass Spectrometry. 22, 65
MSC	Multiplicative Scatter Correction. 13, 22, 30, 31
NMR	Nuclear Magnetic Resonance. 4, 7, 8, 11–14, 18, 21–23, 28, 41, 42, 52, 55, 65
OSC	Orthogonal Signal Correction. 9
PC	Principal Component. 53
PCA	Principal Component Analysis. 16, 18, 22, 34, 45, 46, 53, 59, 63
PLS	Partial Least Squares. 18, 20, 39
PLS-DA	Partial Least Squares - Discriminant Analysis. 18, 22
PLS-r	Partial Least Squares - regression. 18
PPD	Postharvest Physiological Deterioration. 41, 61–63
QDA	Quadratic Discriminant Analysis. 9
RFE	Recursive Feature Elimination. 20, 40
RIPPER	Repeated Incremental Pruning to Produce Error Reduction. 19
RMSE	Root Mean Square Error. 38
SIMCA	Soft Independent Modeling of Class Analogy. 18
SVD	Singular Value Decomposition. 34
SVM	Support Vector Machine. 18, 19, 22, 60
TOCSY	Total Correlation Spectroscopy. 14
TSP, 0.024 g%	Trimethylsilyl Propionate Sodium salt. 42
TSV	Tab Separated Values. 28

UV-vis	Ultraviolet-visible. 4, 10, 11, 13, 18, 23, 41, 52, 54

1

INTRODUCTION

1.1 CONTEXT

Metabolomics can be defined as the identification and quantification of all intra- cellular and extracellular metabolites with low molecular mass. It is one of the so called omics technologies that have recently revolutionized the way biological research is conducted, offering valuable tools in functional genomics and, more globally, in the characterization of biological systems (Nielsen and Jewett, 2007). Indeed, these technologies allow the global measurement of the amounts of different molecules (e.g. Messenger Ribonucleic Acid (mRNA) in transcriptomics, proteins in proteomics) providing numerous applications in biological discovery, biotechnology and biomedical research. Applications of metabolomics data include studying metabolic systems, measuring biochemical phenotypes, understanding and reconstructing genetic networks, classifying and discriminating between different samples, identifying biomarkers of disease, analyzing food and beverage, studying plant physiology, providing for novel approaches for drug discovery and development, among others (Nielsen and Jewett, 2007; Villas-Boas et al., 2007; Mozzi et al., 2012).

However, to achieve these goals, metabolomics data also bring important new challenges regarding the capability of extracting relevant knowledge from typically large amounts of data (Varmuza and Filzmoser, 2009). Indeed, omics data have promoted the development and adaptation of numerous methods for data analysis. Unlike transcriptome and proteome technologies that are based in the analysis of biopolymers with any biochemical similarity, metabolites have a large variance in chemical structures and properties, making difficult the development of high-throughput techniques and therefore reducing the number of molecules that can adequately be measured in a sample (Villas-Boas et al., 2007). This also implies a higher variety of techniques to be able to span all applications.

In order to respond to the challenges created by the data analysis of metabolomics data, a script based software was developed to address a wide variety of common tasks on metabolomics data analysis, providing a general workflow that can be adapted for specific case studies. This package includes tools for the visual exploration and preprocessing of the data, and further analysis to try to discover significant features regarding the type of the data with a wide variety of univariate and multivariate statistical methods, as well as machine learning and feature selection algorithms.

There are a few techniques to obtain metabolomics data. In this work, three techniques will be the main focus: Nuclear Magnetic Resonance (NMR), Infrared (IR) and Ultraviolet-visible (UV-vis) spectroscopies. Those will be explained later in more detail.

1.2 OBJECTIVES

Given the context described above, the main aim of this work will be the design and development of an integrated computational platform for metabolomics data analysis and knowledge extraction. The work will address the exploration and integration of data from distinct experimental techniques, focusing on NMR, UV-vis, and IR.

More specifically, the work will address the following scientific/technological goals:

- To design adequate pipelines for data analysis adapted to the distinct experimental techniques and analysis purposes.
- To implement data analysis methods for metabolomics data, taking advantage of existing open-source software tools, pursuing the development of new methods when needed.
- To design and implement specific machine learning and feature selection algorithms for the analysis of metabolomics data.
- To validate the proposed algorithms with case studies from literature and others of interest in the analysis of the potential of natural products, including for instance propolis or cassava samples.
- To write scientific publications with the results of the work.

1.3 DISSERTATION ORGANIZATION

This dissertation is divided in five chapters. This first chapter made a brief introduction to the theme of this dissertation and defines the objectives proposed with this work.

On the next chapter, the state of the art regarding metabolomics is covered, mentioning the existing techniques for metabolomics data acquisition, the workflow of a metabolomics experiment and data analysis with all its steps, the databases and free tools available for metabolomics, an overview of the main methods for data analysis (both univariate and multivariate) and a description of data integration and existing methods.

The third chapter describes the process of software development to reach the proposed package in R. All the software features are described, as well as the development strategy and the tools that were used, together with the details of the methods developed for each step of the metabolomics workflow.

In the fourth chapter, three case studies were analyzed with the software developed. The details of each case study, the respective results, as well as the interpretation of those results are included in the chapter.

Finally, the last chapter contains the conclusions of the work done and the proposals for future work.

2

STATE OF THE ART

In this chapter, the state of the art of the metabolomics field will be covered. This includes the description of the main techniques and their characteristics, and the available methods for data preprocessing and analysis. The workflow of data analysis for a metabolomics experiment will be fully covered, which includes the preprocessing, metabolite identification and quantification, univariate and multivariate data analysis, the databases and free tools existent for metabolomics and data integration.

2.1 TECHNIQUES

2.1.1 *Nuclear Magnetic Resonance*

NMR spectroscopy is one of the most frequently employed experimental techniques in the analysis of the metabolome. It allows identifying metabolites in complex matrices in a non-selective way, providing a metabolic characterization of a given cell or tissue sample. The 1D- and 2D-NMR experimental approaches are useful for structure characterization of compounds and have been applied for the analysis of metabolites in biological fluids and cells extracts. NMR spectroscopy is a robust and non-destructive sample technique, relatively inexpensive after the initial high costs of installation, also useful for quantification of metabolites in biological samples. The main drawbacks are its poor sensitivity and large sample requirement that can be an obstacle in some situations (Rochfort, 2005). Some variants (e.g. High Resolution Magic Angle Spinning (HRMAS) NMR) overcome some of these limitations.

NMR provides a spectrum with the chemical shifts in the x-axis and the intensities of the signal in the y-axis. These data are typically stored in a matrix where the chemical shifts and classification labels are the columns, and the samples are the lines (or the reverse). In this form, the data are ready to be pre-processed, analyzed and interpreted.

As said above, NMR is one of the most popular techniques employed on metabolomics experiments, so it has a lot of applications in a wide range of areas (**Table** 1). It was used, for example, to determine the metabolic fingerprint and pattern recognition of silk extracts from seven maize landraces cultivated in southern Brazil (Kuhnen et al., 2010), predict muscle wasting (Eisner et al., 2010), identify farm origin of salmon (Martinez et al., 2009) or to classify Brazilian propolis according to their geographic region (Maraschin et al., 2012).

2.1.2 *Infrared Spectroscopy*

Infrared spectroscopy is a technique based on the vibrations of the atoms in a molecule. An infrared spectrum is commonly obtained by passing infrared radiation through a sample and determining what fraction of the incident radiation is absorbed at a particular energy. The energy at which any peak in an absorption spectrum appears corresponds to the frequency of vibration of a part of a sample molecule. The infrared spectrum can be divided into three main regions: the far-infrared ($<$400 cm^{-1}), the mid-infrared (4000–400 cm^{-1}) and the near-infrared (13 000–4000 cm^{-1}).

The most significant advances in infrared spectroscopy have come about as a result of the introduction of Fourier-transform spectrometers. This type of instrument employs an interferometer and exploits the well-established mathematical process of Fourier-transformation. Fourier-Transform Infrared (FTIR) spectroscopy has dramatically improved the quality of infrared spectra and minimized the time required to obtain data. With the advantage of being simpler and less expensive, infrared spectroscopy can provide a complementary view of that provided by NMR.

There are many applications where infrared spectroscopy, usually in association with multivariate statistical methods, has been used with good results (**Table** 1). In the food and beverage area, infrared spectroscopy was used to measure milk composition (Aernouts et al., 2011) and detect and quantify milk adulteration (Santos et al., 2013), classify beef samples according to their quality (Argyri et al., 2010), compare metabolomes of transgenic and non-transgenic rice (Eymanesh et al., 2009), discriminate between different varieties of fruit vinegars (Liu et al., 2008), monitor spoilage in fresh minced pork meat (Papadopoulou et al., 2011) and predict different beer quality parameters (Polshin et al., 2011).

Regarding the biological and medical fields infrared spectroscopy has achieved successful results, for instance in differentiating mild sporadic Alzheimer's disease from normal aging of blood plasma samples (Burns et al., 2009), differentiating strains of bacteria (Kansiz et al., 1999; Preisner et al., 2007) and yeasts (Cozzolino et al., 2006), and in discriminating different plant populations and the effects of environmental changes (Khairudin and Afiqah, 2013).

Reference	Description	Techniques	Preprocessing	Analysis
Acevedo et al. (2007)	Discriminate wines denomination of origin	UV-Vis	Mean centering	SIMCA, SIMCA, KNN, ANNs, PLS-DA
Urbano et al. (2006)	Classification of wines	UV-Vis	First derivative	SIMCA
Pereira et al. (2011)	Predict wine age	UVV	Mean centering, Smoothing, 1st and 2nd derivative, MSC, SNV, OSC	PLS-r
Barbosa-García et al. (2007)	Distinguish between classes of tequila	UV-Vis	Derivative, centering of columns	PLS-DA
Anibal et al. (2009)	Detect Sudan dyes (4 types) in commercial spices	UV-Vis	Mean centering	KNN, SIMCA, PLS-DA
Kruzlicová et al. (2008)	Classification of different sorts of olive oil and pumpkin seed oil, supplemented with oil quality	UV-Vis, IR	None	LDA, Quadratic Discriminant Analysis (QDA), KNN, Linear Regression (LR), ANNs
Kumar et al. (2013)	Clustering and classification of tea varieties	UV-Vis-NIR	Normalization	PCA, K-Means, ANNs
Souto et al. (2010)	Classification of brazilian ground roast coffee	UV-Vis	None	SIMCA, LDA
Thanasoulias et al. (2003)	Discrimination of blue ball point pen inks	UV-Vis	Normalization, Log	PCA, K-Means, Discriminant Analysis (DA)
Adam et al. (2008)	Classification and individualisation of black ballpoint pen inks	UV-Vis	None	PCA
Thanasoulias et al. (2002)	Forensic soil discrimination with UV-Vis absorbance spectrum of the acid fraction of humus	UV-Vis	Normalization, Log	PCA, K-Means, DA
Aernouts et al. (2011)	Measure milk composition	IR	Regions removed, baseline correction, MSC, SNV, 1st and 2nd Savitzky-Golay derivatives, Orthogonal Signal Correction (OSC), mean centering	PLS-r
Santos et al. (2013)	Detect and quantify milk adulteration	IR	Normalization, Savitzky-Golay 2nd derivative, mean centering	SIMCA, PLS-r
Argyri et al. (2010)	Classify beef samples quality and predict the microbial load	IR	Smoothing (Savitzky-Golay), mean centering	PCA, ANNs
Eymanesh et al. (2009)	Comparison of transgenic and non transgenic rice	IR, NMR	Spectra divided in 287 areas	PCA, LDA
Liu et al. (2008)	Discriminate the varieties of fruit vinegars	IR	MSC, 1st and 2nd derivative, SNV, Smoothing (Savitzky-Golay)	PLS-DA, SVM
Papadopoulou et al. (2011)	Quantify biochemical changes occurring in fresh minced pork meat	IR	Mean centering, SNV, Outlier samples removal	PLS, PLS-DA, PLS-r
Polshin et al. (2011)	Prediction of important beer quality parameters	IR	MSC, OSC, baseline correction, SNV, 1st and 2nd Savitzky-Golay derivatives, mean centering	PCA, PLS-r

Burns et al. (2009)	Differentiating mild sporadic Alzheimer's Disease from normal aging	IR	None	LR
Kansiz et al. (1999)	Discriminate between cyanobacterial strains	IR	Normalization, 1st and 2nd Savitzky-Golay derivatives, mean centering	PCA, SIMCA, KNN
Preisner et al. (2007)	Discrimination between different types of the Enterococcus faecium bacterial strain	IR	MSC, extended MSC, baseline correction, SNV, 1st and 2nd Savitzky-Golay derivatives, mean centering, sample outlier removal	PCA, Di-PLS
Cozzolino et al. (2006)	Investigate metabolic profiles produced by S. cerevisiae deletion strains	IR	Autoscaling, centering, second derivative	PCA, LDA
Khairudin and Afiqah (2013)	Discrimination of different plant populations and study temperature effects	IR	Pareto scaling	PCA, PLS-DA
Kuhnen et al. (2010)	Pattern recognition of silk extracts from maize landraces cultivated in southern Brazil	NMR	Phasing, baseline correction	PCA, SIMCA, HCA
Eisner et al. (2010)	Predict if cancer patients are losing weight	NMR	Log transform	NaiveBayes, PLS-DA, decision trees, SVM, Pathway informed analysis
Martinez et al. (2009)	Identify farm origin of salmon	NMR	Peak alignment, normalization	PCA, SVM, Bayesian belief network, ANNs
Maraschin et al. (2012)	Classify Brazilian propolis according to their geographic region	NMR	Phasing, baseline correction, peak alignment, missing values treatment, data filtering	PLS-DA, Random Forests, decision trees, rule set
Masoum et al. (2007)	Confirmation of wild and farmed salmon and their origins	NMR	Filtering uninformative attributes, COW	SVM

Table 1: Applications of NMR, IR and UV-Vis spectroscopies

2.1.3 *Ultraviolet-visible*

Similar to infrared spectroscopy, ultraviolet-visible spectroscopy refers to the absorption of radiation as a function of wavelength, due to its interaction with the sample in the ultraviolet-visible spectral region. It uses light in the visible and adjacent (near-UV-vis and near-infrared) ranges.

Also with the advantage of being simpler and less expensive than more sophisticated techniques, ultraviolet-visible spectroscopy can provide a fast way of discriminating samples, thus offering a complementary view to other types of data in many situations.

Ultraviolet-visible spectroscopy has been applied successfully with chemometrics analysis in areas such as food and beverage, and forensics (**Table** 1). It was used, for instance, to classify wines according to their region, grape variety and age (Acevedo et al., 2007; Urbano et al., 2006; Pereira et al., 2011), discriminate different classes of tequila (Barbosa-García et al., 2007), determine the adulteration of spices with Sudan I-II-III-IV dyes (Anibal et al., 2009), classify different sorts of olive oil and pumpkin seed oil (Kruzlicová et al., 2008), discriminate different Indian tea varieties (Kumar et al., 2013) and classify Brazilian ground roast coffee (Souto et al., 2010).

In forensics, ultraviolet-visible spectroscopy was used in cases such as the discrimination of ball-point pen inks (Thanasoulias et al., 2003; Adam et al., 2008) and the discrimination of forensic soil (Thanasoulias et al., 2002).

2.2 WORKFLOW OF A METABOLOMICS EXPERIMENT

Two distinct approaches can be chosen for planning and executing a metabolomics experiment. The first, known as a chemometrics approach or metabolic fingerprinting, makes use of the preprocessed data, normally spectra or peaks list, and the analysis is done over that data typically for sample discrimination. That approach was for instance used in the discrimination between wild and farmed salmon and their origins (Masoum et al., 2007). The second approach, known as metabolic target analysis or profiling focuses on the identification and quantification of compounds present in the sample, being that information used to run the analysis. A metabolic target analysis was for instance used in predicting cancer-associated skeletal muscle wasting from ^{1}H-NMR profiles of urinary metabolites (Eisner et al., 2010). The second approach is used mostly for NMR data, since identifying and quantifying metabolites in IR and UV-vis data is complex.

The general workflow of a metabolomics experiment generally consists in the steps of sample preparation, data acquisition, preprocessing, data analysis and data interpretation (see Fig. 1). Once the samples are prepared and the data is acquired, it will be preprocessed to correct some issues and improve the performance of the next step, data analysis, where the information will be extracted. The last two steps will be the main focus of this project.

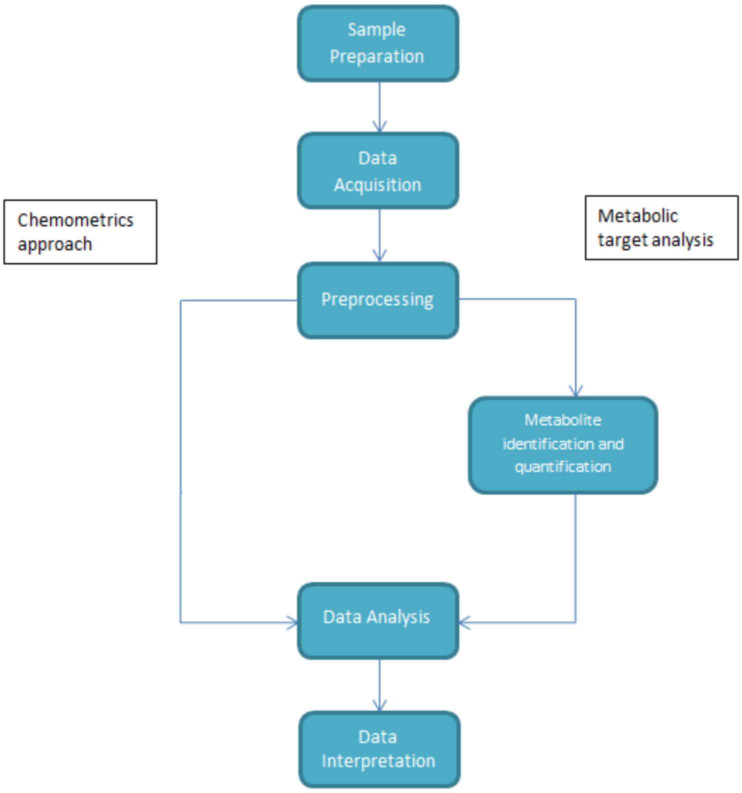


Figure 1: General workflow of a metabolomics experiment

2.2.1 *Preprocessing*

As this step has great importance in the results, almost all the literature has put some thought in preprocessing, testing sometimes a few combinations of methods and analyzing the results obtained, to know which preprocessing methods are better or worse with the data being analyzed. Preprocessing is important to make samples analyzable and comparable. Normally, the preprocessing steps consist on missing values and outlier removal, some peak spectra processing and normalization or scaling.

There are two major choices regarding missing values handling: their removal, removing the feature or the sample containing the missing value, or replacing by a value such as the row or column mean, or using more sophisticated methods (e.g. nearest neighbors). Also, there are sometimes samples or variables that are considered outliers (observation point that is distant from other observations) due to variability in measurement or experimental error. Those are typically excluded from the dataset.

Peak spectra processing refers to corrections or selections made over the spectra. For that task, there are methods such as baseline correction, which, as the names indicates, is needed to correct an unwanted linear or non-linear bias along the spectra. Another method is smoothing, which means reducing the noise of the spectra, and help for both visual interpretation and robustness of the analysis by trying to smooth enough to reduce the noise while retaining as much as possible of the peaks, especially the small ones. One popular method of smoothing is the Savitzky-Golay filter which is a process known as convolution that fits successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares.

There are some other methods related to peak spectra processing like binning, peak alignment, among others. Binning is useful when there are a large number of measurements per spectrum, it divides the spectrum into a desired number of bins, forming a new spectra with fewer variables. The reasons for using binning are that the number of variables can be too high and another less obvious reason is the implicit smoothing and the potential for correcting small peak shifts. There are a few dangers regarding the bad placing of the bins, by removing information or producing false information.

In NMR, peaks can be shifted due to instrumental variations, sample pH or interferences in the analysis, for example. Those shifts need to be corrected before the data analysis, so that each metabolite appears where it is expected. There are a few methods of doing peak alignment, some simpler than others which are more robust. The simplest form of peak alignment is to divide the spectra into a number of local windows where peaks are shifted to match across spectra. That is fast, since everything is done locally, but may lead to misalignment when peaks fall into the wrong local window or are split into two windows, just as when binning. One of the more robust peak-alignment procedures is called Correlation Optimized Warping (COW). It uses two parameters – section length and flexibility – to control how spectra can be warped towards a reference spectrum (Liland, 2011).

In many cases, to make variables comparison possible, the data needs to be standardized to be comparable. The common process of standardizing values does their subtraction by the mean and division by the standard deviation. An alternative process is the use of median and the mean absolute deviation. Centering the data can make many data analysis techniques work better and it means that the mean spectrum is subtracted from each of the spectra. It depends on the particular dataset whether centering the data does make sense or not.

The calculation of derivatives of the spectrum is often used mostly in UV-vis and IR spectroscopy. It is the result of applying a derivative transform to the data of the original spectrum, being useful for two reasons: the first and second derivatives may swing with greater amplitude then the original spectra, in many cases separating out peaks of overlapping bands. In some cases, this can be a good noise filter since changes in baseline have negligible effect on derivatives. Multiplicative Scatter Correction (MSC) is a pre-processing step needed for measurement of many elements. It is a transformation method used to compensate for additive and/or multiplicative effects in spectral data.

On ultraviolet-visible spectroscopy's metabolomics literature, it can be perceived that in comparison to infrared spectroscopy, less methods were used regarding peak spectra processing and the raw data were typically just normalized and derivatives calculated (check **Table** 1). On infrared spectroscopy, more methods for peak spectra processing were applied such as smoothing, with the Savitzky-Golay method being the most used, baseline and scatter correction, most often using MSC. In this case, derivatives were also calculated (first and second derivatives) and the data normalized.

On NMR data, peak spectra processing was employed such as peak alignment, binning and baseline correction, while usually standardization is also applied.

The methods of preprocessing used can vary much from dataset to dataset, taking into account the data's quality and the type of the data. In **Table** 1, the various preprocessing methods used in several cases from the literature are summarized in the fourth column.

2.2.2 *Metabolite identification and quantification*

Since NMR produces a large number of peaks from possibly hundreds of metabolites, the identification and quantification of metabolites is quite difficult due to shifting peak positions, peak overlap, noise and effects of the biological matrix. Figure 2 shows an example of peak overlap. So, this is a complex problem to solve with 100% accuracy, but there are some tools that achieve good results such as Bayesian AuTomated Metabolite Analyser for NMR spectra (BATMAN), MetaboMiner or Metabohunter which will be described briefly next.

BATMAN, is an R package which deconvolutes peaks from 1-dimensional NMR spectra, automatically assigning them to specific metabolites and obtaining concentration estimates. The Bayesian model incorporates information on characteristic peak patterns of metabolites and is able to account for shifts in the position of peaks commonly seen in NMR spectra of biological samples. It applies a Markov Chain Monte Carlo (MCMC) algorithm to sample from a joint posterior distribution of the model parameters and obtains concentration estimates with reduced mean estimation error compared with conventional numerical integration methods (Hao et al., 2012).

MetaboMiner is a standalone graphics software tool which can be used to automatically or semi-automatically identify metabolites in complex biofluids from 2D NMR spectra. MetaboMiner is able to handle both ^{1}1H-^{1}H Total Correlation Spectroscopy (TOCSY) and ^{1}H-^{13}C heteronuclear single quantum correlation (HSQC) data. It identifies compounds by comparing 2D spectral patterns in the NMR spectrum of the biofluid mixture with specially constructed libraries containing reference spectra of 500 pure compounds. Tests using a variety of synthetic and real spectra of compound mixtures showed that MetaboMiner is able to identify >80% of detectable metabolites from good quality NMR spectra (Xia et al., 2008).

MetaboHunter is a web server application which can be used for the automatic assignment of ^{1}H-NMR spectra of metabolites. MetaboHunter provides methods for automatic metabolite identification based on spectra or peak lists with three different search methods and with the possibility for peak drift in a user defined spectral range. The assignment is performed using as reference libraries manually curated data from two major publicly available databases of NMR metabolite standard measurements (HMDB and MMCD). Tests using a variety of synthetic and experimental spectra of single and multi-metabolite mixtures have shown that MetaboHunter is able to identify, in average, more than 80% of detectable metabolites from spectra of synthetic mixtures and more than 50% from spectra corresponding to experimental mixtures (Tulpan et al., 2011).

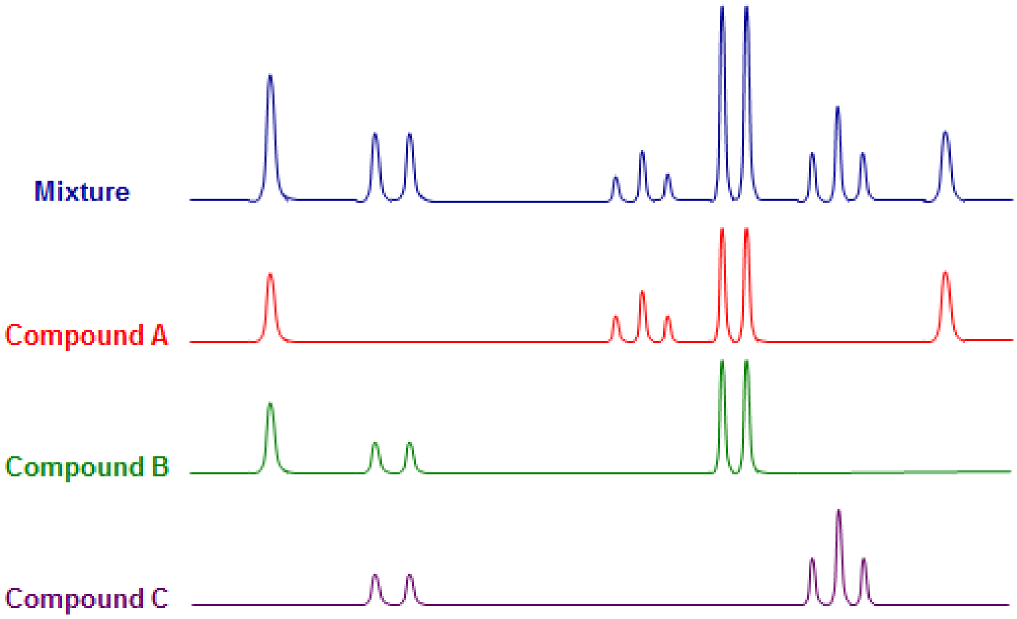


Figure 2: Example of a peak overlap

2.2.3 *Univariate data analysis*

On data analysis, the input is usually a matrix, with either a compound list or a peak list and their values for different samples. Either way, the complexity of metabolomics data requires complex analysis methods. Data analysis can be used to predict values like beer quality parameters (Polshin et al., 2011) or to predict the class label of a certain sample, as in the identification of the farm origin of salmon (Martinez et al., 2009). There are three main types of methods for data analysis: univariate analysis, unsupervised multivariate techniques and supervised multivariate techniques, these last two explained in more detail in subsection 2.2.4 and subsection 2.2.5. With the first approach, the analysis is carried out on a single variable at a time and several statistical measures can be employed to describe the data. The unsupervised and supervised multivariate approaches on the other hand use more than one statistical outcome variable at a time in the analysis. The differences between the last two is that the first do not use any metadata, e.g. information about natural groups within the data, while the latter requires samples that are divided into at least two classes (or groups) to allow the methods to conduct a learning (or training) process.

This section will focus on univariate statistical analysis. From the many techniques, a few most popular ones will be explained given their importance in metabolomics data analysis, namely *t*-tests, Analysis of Variance (ANOVA) and fold change analysis.

A *t*-test is a statistical hypothesis test in which the test statistic follows a Student's *t* distribution.It can be used only for situations where the data is splitted into two groups, being applied to a single data variable. The method can determine if the means of the variable in the two groups are significantly different from each other by giving a *p-value*. If this value is inferior to 0.05 (or any other significance level that is selected) the null hypothesis is rejected, which means that we can say that the means of the variable for the two groups are different. On the contrary, if the *p-value* is superior to 0.05, the null hypothesis cannot be rejected, which means that the estimate of the mean of the two groups can be identical. Applied to metabolomics, *t*-tests can be used on each of the variables (ppm, wavelength, etc) to see what are the most significant ones in the discrimination between two classes.

ANOVA constitutes a set of statistic models and tests associated that is used to analyze the differences between group means and associated procedures. In the simplest form of ANOVA, one-way ANOVA, it provides a statistical test to check if the means of several groups are equal, and so it generalizes the *t*-test to more than two groups. Some post hoc tests can be used with ANOVA like the Tukey's Honest Significance Difference (HSD) test used to compare all possible pairs of group means.

The F-test plays an important role in one-way ANOVA, it is used for testing the statistical significance by comparing the F statistic, which compares the variance between groups with the variance within groups. The ANOVA F-test is known to be nearly optimal in the sense of minimizing false negative errors for a fixed rate of false positive errors.

In the cases where multiple tests need to be conducted, as it is the case with metabolomics data, the p-values can be adjusted for multiple testing using various methods. One of the most common is the False Discovery Rate (FDR). It controls the expected proportion of false discoveries amongst the rejected hypotheses. The FDR is a less stringent condition than the family-wise error rate.

An ANOVA only tells you that there are differences between your groups, not where they lie. Therefore a post-hoc test like Tukey's HSD can be used to examine where the differences lie.

Also, ANOVA generalizes to the study of the effects of multiple factors (Two-way ANOVA, three-way ANOVA, etc). It examines the influence of two or more different categorical independent variables on one dependent variable. It can determine the main effect of contributions of each independent variable and also identifies if there is a significant interaction effect between them .

The Fold Change (FC) is a measure that shows how much a variable mean changes within two groups. It is calculated by getting the ratio of the mean value of the variable in one group to the same value in the other group. The higher the fold change value is for a feature, the more likely that feature is significant to discriminate the two groups.

2.2.4 *Unsupervised Methods*

Usually, Principal Component Analysis (PCA) is employed to check if natural groups (clusters) arise and/or to reduce dimensionality. PCA is the most frequently applied method for computing linear latent variables (components). PCA can be seen as a method to compute a new coordinate system formed by the latent variables (which are orthogonal) and where only the most informative dimensions are used. Latent variables from PCA optimally represent the distances between the objects in a high-dimensional variable space. It is a very popular technique and it is used frequently on all types of metabolomics data (Varmuza and Filzmoser, 2008). The results of a PCA analysis include the scores of the supplied data on the principal components, i.e. the transformed variable values that corresponds to a particular data point, the matrix of variable loadings, which are the weights of each original variable on the new coordinates and the standard deviations (or variance) explained by each of the principal components (or cumulative).

Also, to verify if natural groups arise and determine their composition, clustering techniques such as k-means or hierarchical clustering can be used. Cluster analysis tries to identify coherent groups (i.e., clusters) of objects, while no information about any group membership is available, and usually not even the optimal number of clusters is known. In other words, cluster analysis tries to find groups containing similar objects (Varmuza and Filzmoser, 2008). There are two main types of clustering methods, hierarchical and nonhierarchical clustering.

Hierarchical Cluster Analysis (HCA) organizes data in tree structures with the main clusters containing subclusters and so on. Generally, there are two types of HCA, the first is agglomerative which means that the process starts with each observation as a cluster, and then pairs of clusters are merged as the hierarchy moves up. On the other hand, the second approach is divisive, which means that all observations start in a single cluster and then the splits are done recursively as the hierarchy moves down. The result from HCA is generally a dendrogram. In order to know what clusters should be splitted or merged, a measure of dissimilarity between sets of observations is required. Commonly that measure is achieved with the use of a distance metric between two observations (e.g. Euclidean or Manhattan distances) and a linkage criterion (e.g. nearest neighbour or complete linkage) that calculates the distance between sets of observations as a function of the pairwise distances between observations.

Nonhierarchical clustering organizes data objects into a set of flat groups (typically non overlapping). It is a class of methods that aim to partition a dataset into a pre-defined number of clusters, typically using iterative algorithms that optimize a chosen criterion. One important methods is *k*-means clustering, where heuristics that start from initial random clusters, proceeds by transferring observations from one cluster to another, until no improvement in the objective function can be made. *k*-means clustering is a nonhierarchical clustering method that seeks to assign each observation to a cluster, minimizing the distance of the observation to the cluster mean.

2.2.5 *Supervised Methods: machine learning*

The general workflow of machine learning consists of creating predictive models in order to classify new samples to a specific output, i.e. the goal is to learn a general rule that maps the inputs into outputs. To construct a predictive model, the learner algorithm must generalize from its experience, i.e. training examples must be provided (labeled) and from those, the learner has to build a general model to try to predict new unlabeled samples with good accuracy results. There are two approaches, classification and regression. Classification is used when the output that we are trying to predict is a non-numeric value, i.e. a set of categories, and the algorithm will try to assign the unlabeled sample to one of the categories. Regression is used when the output is a numeric value and the algorithm will try to predict the numeric output value for the new sample.

Besides the accuracy, which is the proportion of samples correctly classified, to evaluate the performance of the models built, there are many other metrics, like the Kappa statistic, which compares the observed accuracy with an expected accuracy. The calculation of the expected accuracy is related to the actual number of instances of each class, along with the number of instances that the classifier labeled for each class. Let nc be the number of classes, t_i be the actual number of instances of a class i, c_i the number of instances that the classifier labeled as belonging to that class and n the total number of instances, the expected accuracy is calculated as it shows in Equation 1.

$$expected\_accuracy = \frac{\sum_{i=1}^{nc} \frac{t_i \times c_i}{n}}{n} \tag{1}$$

Formula 1: Expected accuracy

Now the Kappa statistic can be calculated using both the observed accuracy and the expected accuracy, as shown in Formula 2.

$$kappa\_statistic = \frac{observed\_accuracy - expected\_accuracy}{1 - expected\_accuracy} \tag{2}$$

Formula 2: Kappa statistic

Essentially, the Kappa statistic is a measure of how closely the instances classified by the classifier matched the actual data label, controlling for the accuracy of a random classifier as measured by the expected accuracy.

Popular supervised methods, used commonly in metabolomics experiments, are Partial Least Squares - Discriminant Analysis (PLS-DA) (Partial least squares – discriminant analysis), k-Nearest Neighbors (kNN), Linear Discriminant Analysis (LDA) and Soft Independent Modeling of Class Analogy (SIMCA) for classification tasks. For regression, Partial Least Squares - regression (PLS-r) (Partial least squares – regression) is usually applied.

Partial Least Squares (PLS) is a supervised multivariate calibration technique that aims to define the relationship between a set of predictor data X (independent variables) and a set of responses Y(dependent variables). The PLS method projects the initial input–output data down into a latent space, extracting a number of principal components, also known as Latent Variables (LV) with an orthogonal structure, while capturing most of the variance in the original data. The first LV conveys the largest amount of information, followed by the second LV and so forth. PLS-DA is a variant used when the Y is categorical (Varmuza and Filzmoser, 2008).

SIMCA is a method based on disjoint principal component models proposed by Svante Wold. The idea is to describe the multivariate data structure of each group separately in a reduced space using PCA. The special feature of SIMCA is that PCA is applied to each group separately and also the number of PCs is selected individually and not jointly for all groups. A PCA model is an envelope, in the form of a sphere, ellipsoid, cylinder, or rectangular box optimally enclosing a group of objects. This allows for an optimal dimension reduction in each group to reliably classify new objects. Due to the use of PCA, this approach works even for high-dimensional data with a small number of samples. In addition to the group assignment for new objects, SIMCA also provides information about the relevance of different variables to the classification or measures of separation (Varmuza and Filzmoser, 2008).

K-nearest neighbor (kNN) classification methods require no model to be fitted because they can be considered as memory based. These methods work in a local neighborhood around a considered test data point to be classified. The neighborhood is usually determined by the Euclidean distance, and the closest k objects are used for estimation of the group membership of a new object.

Also, more sophisticated machine learning techniques have been used to build classification models in order to give good prediction results. Techniques like Artificial Neural Network (ANN), Support Vector Machine (SVM)s (Support Vector Machines), random forests, decision trees and rule induction algorithms were applied on some metabolomics experiments with very good results on NMR, IR and UV-vis data that can be used to predict new unlabeled samples (see Table 1).

SVMs are a machine learning approach that can be used for both classification and regression. In the context of classification they produce linear boundaries between object groups in a transformed space of the variables (using a kernel), which is usually of much higher dimension than the original space. The idea of the transformed higher dimensional space is to make groups linearly separable. Furthermore, in the transformed space, the class boundaries are constructed to maximize the margin between the groups. On the other hand, the back-transformed boundaries are nonlinear.

Decision trees are represented by n-ary trees where each node specifies a test to the value of a certain input attribute and each branch that exits the node corresponds to a value of that attribute, originating a similar sub-tree. In the bottom of the tree are the leaves that represent the predicted values. The classification process is recursive, beginning in the root of the tree and going from branch to branch according to the attribute value until a leaf is reached. An example of this technique is the C4.5 algorithm and its implementation in the *J48* method included in the WEKA data mining tool (Quinlan, 1993; Witten et al., 2011).

Classification rules follow a similar structure as decision trees, but instead of an n-ary tree, they consist of a list of rules ordered by the quality of the rules. If an example's attributes follow the conditions of the rules then the classification output is the output value of the rule. An implementation of this is JRip, which implements a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER) (Cohen, 1995; Witten et al., 2011).

Random forests are an ensemble learning method for classification and regression, consisting of a set of decision trees. It uses the bagging method as a way to choose the examples and selects the set of attributes to test in each tree randomly. The CART algorithm is used as the tree induction algorithm and uses the out-of-bag error for validation (Breiman, 2001).

ANN are computational representations that mimic the human brain, trying to simulate its learning process. The term "artificial" means that neural nets are implemented in computer programs that are able to handle the large number of necessary calculations during the learning process. The most popular type of ANN has three layers, the input layer, the hidden layer and the output layer. Input data are put into the first layer, the hidden layer nodes do some calculations and the output is gathered from the last layer. ANNs can be used for regression and classification.

In **Table** 1, some of these (and other) methods used in metabolomics data analysis studies available in the literature are listed. These include tasks of visual analysis, data reduction, classification and regression tasks.

2.2.6 *Feature Selection*

Feature selection tries to find the minimum set of features (in metabolomics, can be compounds or peaks, for instance) that achieves maximum prediction performance. Feature selection is important in various ways. It makes models more robust and generally improves accuracy, also providing important information to users regarding which variables are more discriminant. There are two main approaches: filters and wrappers. Filters select features on the basis of statistical properties of the values of the feature and those of the target class, while wrappers rely on running a particular classifier and searching in the space of feature subsets, based on an optimization approach.

A number of feature selection methods has been proposed in the literature. Among them, the most popular wrappers are interval selection methods, such as interval Partial Least Squares (iPLS) used in species identification using infrared spectroscopy (Lima et al., 2011) and Genetic Algorithms (GA) used in bacteria discrimination using FTIR spectroscopy (Preisner et al., 2007).

In the iPLS method, the data are subdivided into non-overlapping partitions; each of these groups undergoes a separate PLS modeling to determine the most useful variable set. To capture relevant variation in the output, models based upon the various intervals usually need a different number of PLS components than do full-spectrum models (Balabin and Smirnov, 2011).

A GA is a metaheuristic optimization algorithm that has a population of candidate solutions. Over a number of generations, it will evolve the population of candidate solutions, generating new solutions in each generation using selection operators and reproduction operators. For each solution its quality is given by a fitness value, calculated using a defined objective function.

Also Recursive Feature Elimination (RFE), also known as backward selection, is a wrapper approach. It first uses all features to fit the model, and then each feature is ranked according to its importance to the model. Then for each subset the most important variables are kept and the model is refitted and performance calculated, with the rankings for each feature recalculated. Finally, the subset with the best performance is determined and the top features of the subset are used to fit the final model.

Regarding filter approaches, simple univariate statistical tests can be used to pre-select the features that pass some specific criterion (e,g. information gain, correlation with output variable) and then the model is built with those selected features. Also flat pattern filters that remove variables with low variability among instances can be used to filter the features prior to the construction of the model.

2.3 DATABASES OF METABOLOMICS DATA

As said before, the number of databases of metabolomics data made available to the public is growing (**Table** 2). The most notable database that can be mentioned is the Human Metabolome Database (http://www.hmdb.ca/). For experimental data, there is also MetaboLights (http://www.ebi.ac.uk/metabolights/). Other databases provide also complementary information about metabolites and metabolism, such as KEGG (http://www.genome.jp/kegg/), PubChem (http://pubchem.ncbi.nlm.nih.gov/) and others.

2.3.1 *HMDB*

The Human Metabolome Database is a freely available electronic database containing detailed information about small molecules (metabolites) found in the human body. It is intended to be used for applications in metabolomics, clinical chemistry, biomarker discovery and general education. Four additional databases, DrugBank (http://www.drugbank.ca), T3DB (http://www.t3db.org), SMPDB (http://www.smpdb.ca) and FooDB (http://www.foodb.ca) are also part of the HMDB suite of databases. DrugBank contains equivalent information on ∼1600 drugs and drug metabolites, T3DB contains information on 3100 common toxins and environmental pollutants, and SMPDB contains pathway diagrams for 440 human metabolic and disease pathways, while FoodDB contains equivalent information on ∼28,000 food components and food additives. (Wishart et al., 2013)

2.3.2 *MetaboLights*

MetaboLights is a database for metabolomics experiments and derived information. The database is cross-species, cross-technique and covers metabolite structures and their reference spectra as well as their biological roles, locations and concentrations, and experimental data from metabolic experiments. It will provide search services around spectral similarities and chemical structures (Haug et al., 2013).

Name	URL	Short description
HMDB	http://www.hmdb.ca	Detailed information on metabolites found in the human body
Metabolights	http://www.ebi.ac.uk/metabolights	Database for metabolomics experiments and derived information
MMMDB	http://mmmdb.iab.keio.ac.jp	Mouse multiple tissue metabolome database
SMDB	http://www.serummetabolome.ca	Serum metabolome database
MeKO@PRIMe	http://prime.psc.riken.jp/meko/	A web-portal for visualizing metabolomic data of Arabdiopsis
MPMR	http://metnetdb.org/mpmr_public/	A metabolome database for medicinal plants
ECMDB	http://www.ecmdb.ca/	E. coli metabolome database
YMDB	www.ymdb.ca	Yeast metabolome database
SMPDB	http://www.smpdb.ca	Small Molecule Pathway Database
FooDB	http://www.foodb.ca	Resource on food constituents, chemistry and biology
DrugBank	http://www.drugbank.ca	Combines drug data with comprehensive drug targets
BMRB	http://www.bmrb.wisc.edu/metabolomics	Central repository for experimental NMR spectral data
MMCD	http://mmcd.nmrfam.wisc.edu	Database on small molecules of biological interest
BML-NMR	http://www.bml-nmr.org/	Birmingham Metabolite Library NMR database
IIMDB	http://metabolomics.pharm.uconn.edu/iimdb	Both known and computationally generated compounds
KEGG	http://www.genome.jp/kegg/	Contains metabolic pathways from a wide variety of organisms
PubChem	http://pubchem.ncbi.nlm.nih.gov	Database of chemical structures of small organic molecules

Table 2: Databases of metabolomics data

2.4 AVAILABLE FREE TOOLS FOR METABOLOMICS

Computational tools for metabolomics have been growing in the past years being already available a good set of free tools (Table 3). One notable example is MetaboAnalyst (Xia et al., 2009, 2012), a web-based metabolomic data processing tool. It accepts a variety of input data (NMR peak lists, binned spectra, Mass Spectrometry (MS) peak lists, compound/concentration data) in a wide variety of formats. It also offers a number of options for metabolomic data processing, data normalization, multivariate statistical analysis, visualization, metabolite identification and pathway mapping. In particular, MetaboAnalyst supports methods such as: fold change analysis, t-tests, PCA, PLS-DA, hierarchical clustering SVMs. It also employs a large library of reference spectra to facilitate compound identification from most kinds of input spectra. It works based on the open-source R scientific computing platform (http://www.r-project.org).

Also, there are several packages from Bioconductor project (http://www.bioconductor.org) for R, which target a set of biological data analysis tasks, focusing on omics data, with some specific metabolomics packages mainly for MS, which is another technique for metabolomics data acquisition that will not be the focus here (http://bioconductor.org/packages/release/BiocViews.html#___Metabolomics).

Still for R, there two important packages, *hyperSpec* and *ChemoSpec*. *ChemoSpec* is a collection of functions for plotting spectra (NMR, IR, etc) and carrying out various forms of top-down exploratory data analysis, such as HCA, PCA and model-based clustering. Robust methods appropriate for this type of high-dimensional data are employed. *ChemoSpec* is designed to facilitate comparison of samples from treatment and control groups. It also has a number of data pre-processing options available, such as normalization, binning, identifying and removing problematic samples, baseline correction, identifying and removing regions of no interest like the water peak in ^{1}-NMR or the CO_2 peak in IR (Hanson, 2013).

Name	URL	Short description
MetaboAnalyst	http://www.metaboanalyst.ca	Web application to analyze metabolomic data
hyperSpec	http://hyperspec.r-forge.r-project.org	R package to handle spectral data and metadata
ChemoSpec	http://cran.r-project.org/web/packages/ChemoSpec	R package to handle spectral data
Metabolomic Package	http://cran.open-source-solution.org/web/packages/Metabonomic/	R-package GUI for the analysis of metabonomic profiles
speaq	https://code.google.com/p/speaq/	Integrated workflow for robust alignment and quantitative analysis of NMR
Automics	https://code.google.com/p/automics/	Platform for NMR-based spectral processing and data analysis
MeltDB	https://meltdb.cebitec.uni-bielefeld.de	Web-based system for data analysis and management of metabolomics
metabolomics	http://cran.r-project.org/web/packages/metabolomics	Collection of functions for statistical analysis of metabolomic data
metaP-Server	http://metabolomics.helmholtz-muenchen.de/metap2/	Web application to analyze metabolomic data
Bioconductor	http://bioconductor.org/packages/release/BiocViews.html #___Metabolomics	Bioconductor R packages for metabolomics

Table 3: Available free tools for metabolomics data

hyperSpec is a R package that allows convenient handling of *hyperSpec*tral data sets, i. e. data sets combining spectra with further data (metadata) on a per-spectrum basis. The spectra can be anything that is recorded over a common discretized axis. *hyperSpec* provides a variety of possibilities to plot spectra, spectral maps, the spectra matrix, etc. It also provides preprocessing and data analysis methods. Regarding proprocessing, *hyperSpec* offers a large variety of methods, such as cutting the spectral range, shifting spectra, removing bad data, smoothing, baseline correction, normalization or MSC. In data analysis, PCA and clustering techniques can be performed with this package (Beleites, 2012).

As wee can see from Table 3, there are many tools for the metabolomics analysis, but MetaboAnalyst, which is the only integrated framework looses flexibility because it is more oriented to graphical user interfaces (GUI).

2.5 DATA INTEGRATION

Data integration tries to aggregate data from different techniques in order to improve performance of the prediction models; this is also known as data fusion. There are mainly three types of fusion strategies, namely, information/data fusion (Low-Level Fusion (LLF)), feature fusion (Intermediate-Level Fusion (ILF)), and decision fusion (High-Level Fusion (HLF)). The first two work on variable level while the latter works on the decision level.

Variable level integration concatenates the variables into a single vector, which is called a "meta-spectrum". Data must be balanced (all variables in the same scale) prior to the fusion process. If the number of concatenated variables is quite high, a variable selection is required. LLF involves combining the outputs from two or more techniques to create a single signal (see Fig. 3). ILF first involves feature extraction onto each source of data, followed by a simple concatenation of the feature sets obtained from multiple information sources (see Fig. 4) (Subari et al., 2012). Decision level data fusion combines the classification results obtained from each individual technique using different methods like fuzzy set theory or Bayesian inference.

There are some studies using data integration from different techniques. In some, data integration does not improve classification performance significantly comparing to the individual results, but in some other cases, data integration can improve significantly the classification results.

Data fusion for determining Sudan dyes in culinary spices (Anibal et al., 2011), which combines data from ^{1}H-NMR and UV-vis spectroscopy, applies the two types of data integration and uses fuzzy set theory in the decision level approach, improving the overall performance of classification. Also pure and adulterated honey classification (Subari et al., 2012), which combines data from e-nose and IR spectroscopy, two types of variable level data integration, namely LLF and ILF, gave better results than a single technique alone.

Classifying white grape musts in variety categories (Roussel et al., 2003a,b), which combines data from UV-vis and IR spectroscopy, also applies the two main approaches. The variable level approach, LLF, does not significantly improve results while the decision level approach, which employs data integration based on the Bayesian inference, improved the classification results.

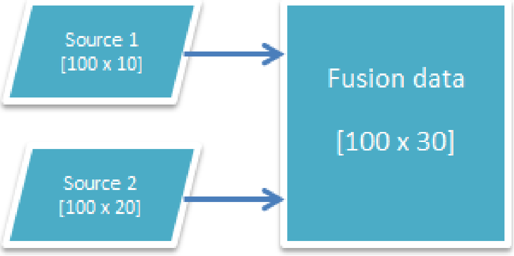


Figure 3: Low-level fusion example

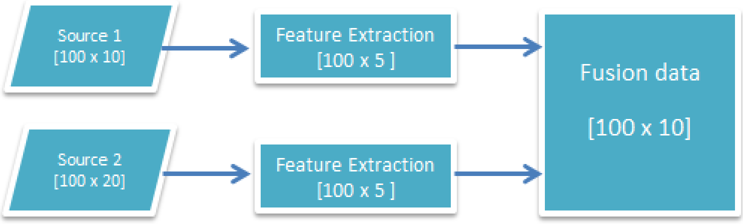


Figure 4: Intermediate-level fusion example

3

DEVELOPMENT

This chapter will cover the details about the developed package and the development process. The general structure to keep spectral or other types of metabolomics data will be explained, as well as the different functional sections of the package such as: the preprocessing, univariate analysis, unsupervised and supervised multivariate analysis, including machine learning and feature selection. Also, the technologies used (software, plugins, packages, etc) will also be referenced.

3.1 DEVELOPMENT STRATEGY AND TOOLS

To achieve the defined goals, a package with features covering the main steps of the metabolomics data analysis workflow was developed, containing functions for the data reading and dataset creating, preprocessing and data analysis.

The package was developed with functions easy to call, i.e. with few mandatory parameters, but also very flexible, since while most functions have default parameters, they also have a large number of parameters that users can use to change the default behavior. The package integrates many functions imported and sometimes adapted from other packages, integrating various packages over a unique interface. The package's functions were meant to be easy to use and to provide abundant graphical visualization options of the results. The idea is to minimize the number and complexity of the lines of code needed to make a pipeline analysis over a certain dataset, but also to easily allow creating variants for this analysis with low complexity.

The package was developed using the R environment (`http://www.r-project.org`), a free integrated software environment for data manipulation, scientific and statistical computing and graphical visualization. The most relevant characteristics of this platform are the data handling and storage facilities, a base set of operators for matrix calculations, a large set of tools for data analysis, the graphical facilities that can generate various types of graphics for data analysis. Also, it has a well developed and effective programming language, the ’S’ language, allowing to develop new functions and scripts. Since it is a free environment, it has the contribution of a large collection of *packages* developed by anyone who wants to contribute (Venables et al., 2014). With such characteristics and the goal of this thesis in mind, the choice of using R became clear.

RStudio (http://www.rstudio.com) was the Integrated Development Environment (IDE) chosen to develop the R scripts and assemble the package. Reports were made using a RStudio plugin named R Markdown (http://rmarkdown.rstudio.com), a plugin that can create easily dynamic documents, presentations and reports from R.

RStudio is a free integrated development environment for R. It comes with a console, an editor that supports direct code execution with syntax highlighting, tools for plotting, easy installation of packages, workspace management and history. RStudio is written in the C++ programming language and uses the graphical user interface from the Qt framework (Verzani, 2011). It also has very easy to use tools to assemble a package with the source code and the functions documentation.

R Markdown uses the syntax of markdown (http://daringfireball.net/projects/markdown/basics) together with embedded R code chunks that are run, and the output of that code is included in the generated report. This framework can create HTML, PDF and Microsoft Word documents, and also some presentation formats. This allows to quickly see the results from different pipelines of the data analysis and provide easy to generate and self-running reports of such analyses. It uses the *knitr* package (http://yihui.name/knitr/ as the engine for dynamic report generation with R.

In the future the developed package will be made available on CRAN, which is a network of ftp and web servers that store identical and up-to-date versions of code and documentation for R packages available for the community. Currently, the package can be accessed through the following repositorium with GIT (http://git-scm.com/):

```
git clone https://chrisbcl@bitbucket.org/chrisbcl/metabolomicspackage.git
```

It can also be installed and loaded in R automatically with the following commands:

```
library(devtools)
install_bitbucket("metabolomicspackage","chrisbcl")
library(metabolomicsUM)
```

The package also comes with the functions documentation, including all details as the description of the parameters, the return value and a short description of the function.

Different R packages were used to help developing the R scripts, which will be referred in the sections where they were used.

3.2 DATASET STRUCTURE AND CREATION

3.2.1 *Dataset structure*

The basic structure to keep spectral or other types of metabolomics data will be general, independent of the type of data and source.

A dataset will be an R list consisting of the following fields:

- **description** - a brief textual description of the dataset and possible pre-processing steps performed to reach it;

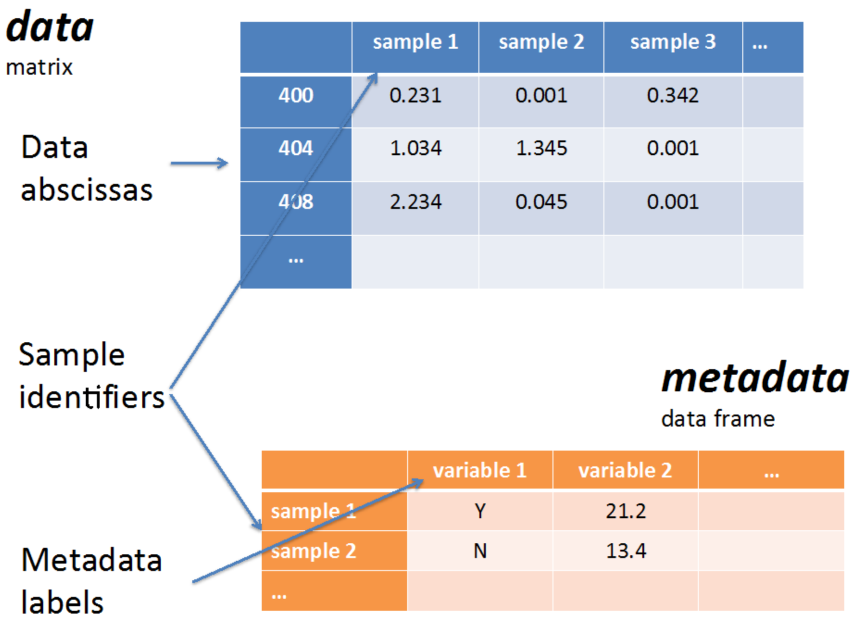
- **type** - a string indicating the type of data; possible values, at this stage, are “nmr-peaklist”, “nmr-spectra”, “concentrations”, “ir-spectra”, “uvv-spectra”, “raman-spectra”, “fluor-spectra” and “undefined”;

- **data** - the metabolomics data, kept in a numeric matrix with columns representing samples and rows representing the x axis values of the data (wavelengths, frequencies, shifts, compound names, etc); values in the matrix represent y axis values (intensity, absorbance, concentrations, etc); row names of the matrix keep the x axis values (as text strings); column names keep sample identifiers;

- **metadata** - extra variables defining information about the samples; kept in a data frame (rows are samples, columns are variables) – allows numerical and categorical (factors in R) variables;

- **labels** - list that allows to define labels for the x-axis ($x) and for the y-axis values ($val), to be used in plotting for example.

In Figure 5, a graphical representation of the data structure is provided.



	sample 1	sample 2	sample 3	...
400	0.231	0.001	0.342	
404	1.034	1.345	0.001	
408	2.234	0.045	0.001	
...				

	variable 1	variable 2	...
sample 1	Y	21.2	
sample 2	N	13.4	
...			

Figure 5: Representation of the structure of the data in a dataset

In this structure, the data matrix and the type cannot be NULL. The other fields can be NULL, including the metadata. When handling spectral data, the row names of the data matrix are assumed to be numerical and can be interpreted as numerical values in many functions. If a set of non-numerical row names are defined for spectral data, the dataset will be considered invalid.

3.2.2 *Reading data and creating datasets*

The core function *create.dataset* allows to create the structure mentioned above. A number of different file formats are supported by the package, which includes Comma Separated Values (CSV) or Tab Separated Values (TSV) text files and (J)DX spectra files. The function *read.dataset.csv* allows to read data from CSV/TSV files (one for data and another, optionally, for metadata) specifying a number of options for the file format. For reading (J)DX files, the function *read.dataset.dx* can be used, where each file represents a distinct sample. Metadata can be additionally given as a CSV/TSV file. The core reading functions for (J)DX are provided by the *ChemoSpec* package, which was described in section 2.4.

Apart from the general structure given above, the package also handles data in other formats that can typically be processed to datasets in the format defined. One of these cases is the peak list format, used in the case of NMR. The structure of a peak list is simple: it is an R list, where each sample is one field of the list (the name of the field is the sample name). Each sample is represented as a data frame with two columns: one with the peak name or abscissa (e.g. frequency) and the other with the peak intensity. The functions *read.multiple.csvs* and *read.csvs.folder* allow to read a set of CSV files with peak lists, and *dataset.from.peaks* function creates a dataset from a list of peaks. Also peak alignment functions were developed and 2 different algorithms implemented, one originally developed in this work and another taken from MetaboAnalyst software which was mentioned in section 2.4. Both use a moving window and a defined step. The function that allows the grouping of peaks is the *group.peaks* function.

Both algorithms group peaks based on their ppm values with a moving window (default is 0.03 ppm). Peaks that are in the same group are aligned to their median ppm. For each group, if more than one peak is detected from the same sample, it will be replaced by their sum. If a group contains peaks that are in less than half of the samples, these will not be included in the alignment. The main difference of the algorithms is that the MetaboAnalyst one does allow overlapping of the windows, with a step size that is typically half of the window size, while the developed algorithms uses a simpler strategy by considering non-overlapping windows.

3.2.3 *Conversion to/from other packages*

The developed package interfaces well with other R packages devoted to the analysis of metabolomics data. Interfaces with the data structures of *hyperSpec* and *ChemoSpec* packages were already developed.

The interface with the *hyperSpec* package provides functions to convert from an *hyperSpec* object into our dataset and also to convert our datasets into hyperSpec objects. This allows to easily take advantage of functions implemented in this package.

Similarly, this package implements the conversion of a dataset in the format of *ChemoSpec* to our own format. The reverse conversion is not possible since *ChemoSpec* does not provide functions to create objects, as these are only created from data files.

3.3 EXPLORATORY ANALYSIS

3.3.1 *Statistics*

The package includes a number of functions that allow to calculate global statistics over the data. The functions *stats.by.variable* and *stats.by.sample* allow to calculate the main descriptive statistics over the data matrix of a dataset. The difference of the functions is that the first is for variables and the second for samples. These functions allow to get the minimum and maximum value, the first and third quantiles and the mean and median. For specific calculations, the functions *apply.by.variable* and *apply.by.sample* allow to apply any function to the variables or samples of a dataset, allowing to easily calculate means, medians, standard deviations or any other statistic. All these functions can be applied, by default, to the full dataset, or to subsets of samples or variables, specified by name or by index.

3.3.2 *Graphics*

Also, a number of functions to provide graphical visualization of the data are available. A basic visualization function allows to see the distribution of values for (a subset of) the variables in the dataset in the form of boxplots (function *boxplot.variables*). In Figure 6 we have a example of the plot with the distribution of values for a subset of the variables (in this case metabolites) in the dataset.

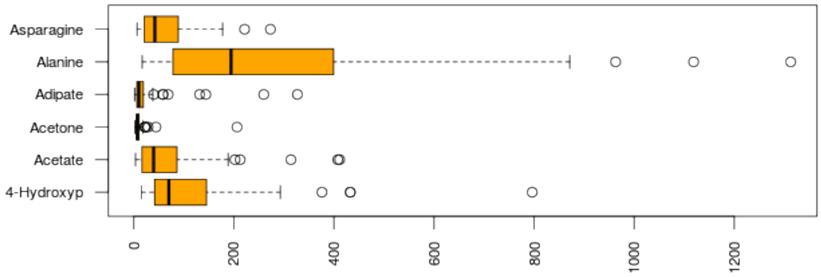
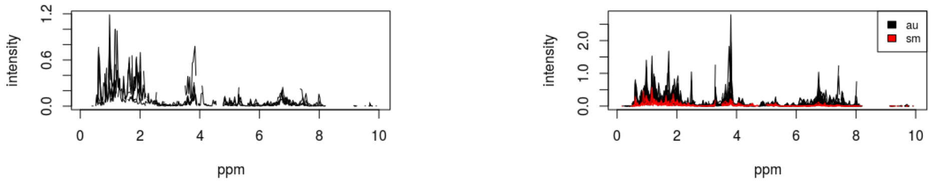


Figure 6: Boxplot for a subset of the metabolites in a dataset

The functions *plot.spectra.simple* and *plot.spectra* allow to visualize plots of the spectra in the dataset, in the last case colored by a given metadata variable (factor). Both allow to define specific subsets of samples and variable bounds. These functions are only applicable to spectral data (where variables are represented by numerical values). Figure 7 shows an example of each spectra plot.

For these graphical visualization functions, the base graphics system of R was used.



(a)

(b)

Figure 7: (a) Plot of *plot.spectra.simple* function from a subset of a dataset with 5 samples. (b) Plot of *plot.spectra* function from a subset of a dataset with 20 samples and colored by a metadata variable.

3.4 PREPROCESSING

3.4.1 *Filtering/subsetting data*

To extract relevant parts of a dataset, a number of functions were developed. The functions *subset.samples*, *subset.x.values* and *subset.metadata* allow to extract relevant subsets of samples, data variables and metadata variables, respectively. The first two options can be combined through the use of function *subset.by.samples.and.xvalues*. To use these functions the user must specify which samples or/and values are intended in the new subset.

Specific functions allow, for instance, to create a subset of random samples from the dataset (*subset.random.samples)* specifying the number of samples, or to select samples from the dataset according to the values of a metadata variable (*subset.samples.by.metadata.values*).

Similar functions are available to remove data from the dataset. These allow to define the data to be removed, rather than the data to be kept. Specific functions allow to remove samples or variables that have a number of missing values higher than a defined value (functions *remove.samples.by.nas* and *remove.variables.by.nas*).

Also, there is a function to aggregate samples (*aggregate.samples*) according to a vector specifying the groups of samples to aggregate and an aggregation function to apply to the values of the data.

3.4.2 *Spectral Corrections*

A number of functions for spectral correction are provided. These include functions for shifting and smoothing the spectra, as well as for baseline, offset and background correction, MSC and the calculation of the first derivative.

For shifting the spectra, two methods were implemented: constant and interpolation, where interpolation can be done by spline interpolation or linear interpolation. The shifts, if not given, can be automatically calculated. The implementations were based on the *hyperSpec* package and on the examples provided by the vignette in section 12.2 (Beleites, 2014).

Regarding smoothing, the function *smoothing.interpolation* has two methods implemented, also from the *hyperSpec* package, namely the *bin* method and *loess* method. The first bins the spectral axis by averaging every by data points, while the second applies the R's loess function, that fit a polynomial surface determined by one or more numerical predictors using local fitting, for spectra interpolation.

For spectra correction, the background and offset correction functions were implemented also with the help of *hyperSpec* package. The baseline correction used the *baseline* package and the MSC used the function *msc* from the *pls* package. The functions to do the spectra correction are *background.correction*, *offset.correction*, *baseline.correction* and *msc.correction*. Both background and offset correction use the function *sweep* to obtain the summary statistics, *quantile* and *min* respectively. The methods for baseline correction can be:

- **als** - Asymmetric Least Squares, baseline correction by 2nd derivative constrained weighted regression;
- **fillPeaks** - An iterative algorithm using suppression of baseline by means in local windows;
- **irls** - Iterative Restricted Least Squares, an algorithm with primary smoothing and repeated baseline suppressions and regressions with 2nd derivative constraint;
- **lowpass** - Low-pass filter, an algorithm for removing baselines based on Fast Fourier Transform filtering;
- **medianWindow** - an implementation and extension of Mark S. Friedrichs' model-free algorithm;
- **modpolyfit** - Modified polynomial fitting, an implementation of Chad A. Lieber and Anita Mahadevan-Jansen's algorithm for polynomial fitting;
- **peakDetection** - A translation from Kevin R. Coombes et al.'s MATLAB code for detecting peaks and removing baselines;
- **rfbaseline** - Robust Baseline Estimation, Wrapper for Andreas F. Ruckstuhl, Matthew P. Jacobson, Robert W. Field, James A. Dodd's algorithm based on LOWESS and weighted regression;
- **rollingBall** - Ideas from Rolling Ball algorithm for X-ray spectra by M.A.Kneen and H.J. Annegarn. Variable window width has been left out.

The *hyperSpec* package is mentioned in section 2.4.

3.4.3 *Missing Values*

To identify missing values in the dataset, the function *count.missing.values* allows to count the number of missing values in the dataset. If there are any, the function *missingvalues.imputation* allows to replace those values, according to distinct methods: a constant user-defined value, the mean or median of the variable, by the kNN or by linear interpolation. The kNN method uses the *impute* package from Bioconductor.

An alternative is to remove samples and/or variables that have a number of missing values above a given threshold, as mentioned above in subsection 3.4.1.

3.4.4 *Data normalization, transformation and scaling*

For sample normalization, data transformation and scaling there are a number of functions that allow performing these operations.

Regarding sample normalization, the *normalize* function implements a number of options including the use of a reference sample or feature, and dividing by the sum or the median of the values in the sample.

The function *transform.data* implements two transformation methods, the first is the logarithmic transformation and the second one, cubic root transformation. The formula for the logarithmic transformation, which is tolerant to zero and negative values, is presented below in Formula 3, where x are the sample's values and $min.val$ is minimum absolute value from the dataset divided by 10:

$$f(x) = \log_2\left(\frac{x + \sqrt{x^2 + min.val^2}}{2}\right) \tag{3}$$

Formula 3: Logarithmic transformation

The cubic root transformation formula is presented in Formula 4, with x being the sample's values:

$$f(x) = \sqrt[3]{x} \tag{4}$$

Formula 4: Cubic root transformation

For the purpose of data scaling, the function *scaling* comes with a few method options, which are auto scaling (subtracting by the mean and dividing by the standard deviation, Formula 5), Pareto scaling (divides by the square root of standard deviation, Formula 6), range scaling (divides by the range, Formula 7) and scaling to a specific interval (Formula 8). The scaling is applied to the variables and in the formulas presented below, x represents the variable's values of the samples and $sup.i$ and $inf.i$ the corresponding superior and inferior limits of the defined interval. Also, σ_x represents the standard deviation of x, $\bar{x}$ the mean of x, $\max x$ the maximum value of x and $\min x$ represents the minimum of x.

$$f(x) = \frac{x - \bar{x}}{\sigma_x} \tag{5}$$

Formula 5: Auto scaling

$$f(x) = \frac{x - \bar{x}}{\sqrt{\sigma_x}} \tag{6}$$

Formula 6: Pareto scaling

$$f(x) = \begin{cases} x, & \text{if } \max x = \min x \\ \frac{x-\bar{x}}{\max x - \min x}, & \text{otherwise} \end{cases} \tag{7}$$

Formula 7: Range scaling

$$f(x) = \begin{cases} x \times (sup.i - inf.i) + inf.i, & \text{if } \max x = \min x \\ \frac{x-\min x}{\max x - \min x} \times (sup.i - inf.i) + inf.i, & \text{otherwise} \end{cases} \tag{8}$$

Formula 8: Scaling to a specific interval

3.4.5 *Flat pattern filtering*

Flat pattern filtering can be used to reduce the size of the data by removing variables that show low variability. This package implements flat pattern filtering using the function *flat.pattern.filter*, where several functions can be used to measure the variability, which are: standard deviation, interquartile range, mean absolute deviation, relative standard deviation, mean and median. The number of variables to filter can be chosen as a percentage of the variables in the dataset or, alternatively, a threshold can be defined for the values calculated by the filtering function.

3.5 UNIVARIATE DATA ANALYSIS

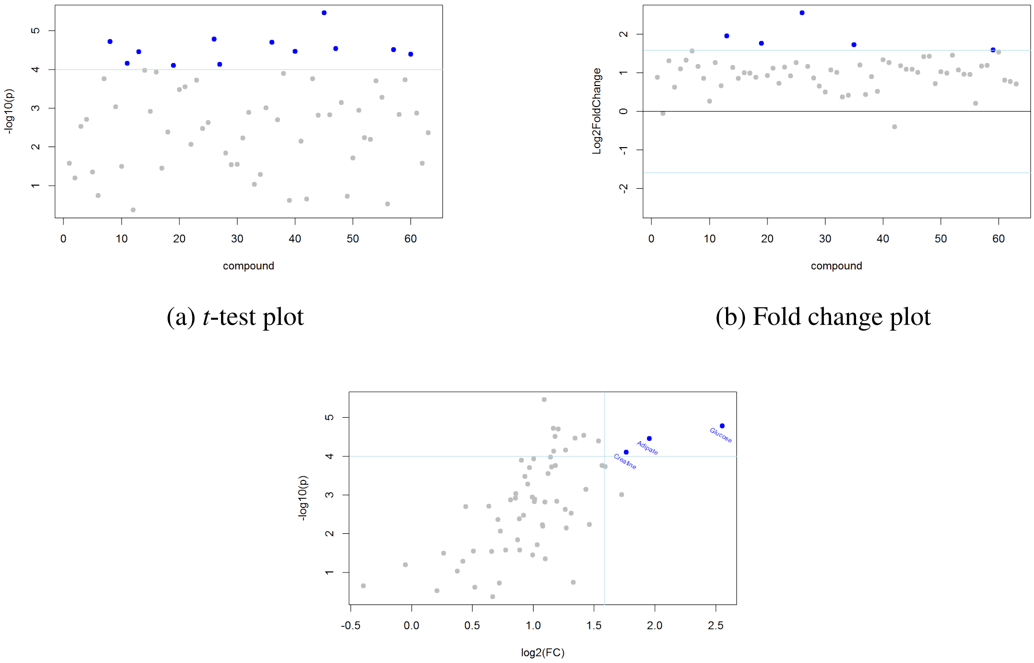
This package includes a number of distinct univariate analysis methods implemented by several functions. The main types of analysis are: correlation analysis, fold change analysis, *t*-tests and ANOVA.

Regarding correlation analysis, the function *correlations.dataset* can be used to calculate the correlations between variables or samples in the dataset and three distinct methods can be used: Pearson, Kendall or Spearman. The correlations result matrix can be visualized as a heatmap with the function *heatmap.correlations*.

The function *fold.change* allows to calculate fold changes of values considering two groups of samples as defined by a metadata variable. Also, we can visualize the result from fold change by plotting the results with the function *plot.fold.change* and give a threshold value (cyan horizontal line) for the dots in the plot appear blue in case the values are higher than the threshold and grey otherwise. An example can be seen in Figure 8b where the x-axis represents the variables and the y-axis represents the fold change value (or the base 2 logarithm of the fold change values).

To perform *t*-tests, the function *tTests.dataset* allows to measure differential values considering two groups of samples as defined by a metadata variable. Also, we can visualize the result of the *t*-test by plotting the results with the function *plot.ttests* and as the fold change plot, a threshold is given in this one too. In Figure 8a there is an example, where the x-axis represents the variables and the y-axis represents the logarithm to the base 10 of the *p-values*.

It is also possible to generate a volcano plot that intersects both results from fold change and *t*-test. This plot combines the measure of the *p-value* from the *t*-test with the fold change value that enables a quick visualization and identification of those variables that have higher changes and also are statistically significant. In figure 8c there is an example, where the x-axis represents the logarithm to the base 2 of the fold change values and the y-axis represents the negative logarithm to the base 10 of the *p-values*. The points that are above the thresholds for both fold change and *t*-test also appear in blue.



(a) *t*-test plot

(b) Fold change plot

(c) Volcano plot

Figure 8: *t*-test, fold change and volcano plot from a metabolite's concentrations dataset

Regarding ANOVA, the package implements one-way ANOVA with the *Tukey HSD post-hoc test* and also multifactorial ANOVA. The functions to do the analysis are *aov.all.vars* and *multifactor.aov.all.vars* for multifactor ANOVA. In both cases, the aproach is similar to the t-tests, where the procedure is repeated over the variables in the dataset, gathering a p-value for each and providing a rank of the variables to highlight which ones can discriminate best the groups.

All these functions use the *stats* package that contains functions for statistical calculations and random number generation from the base R system library.

3.6 UNSUPERVISED MULTIVARIATE ANALYSIS

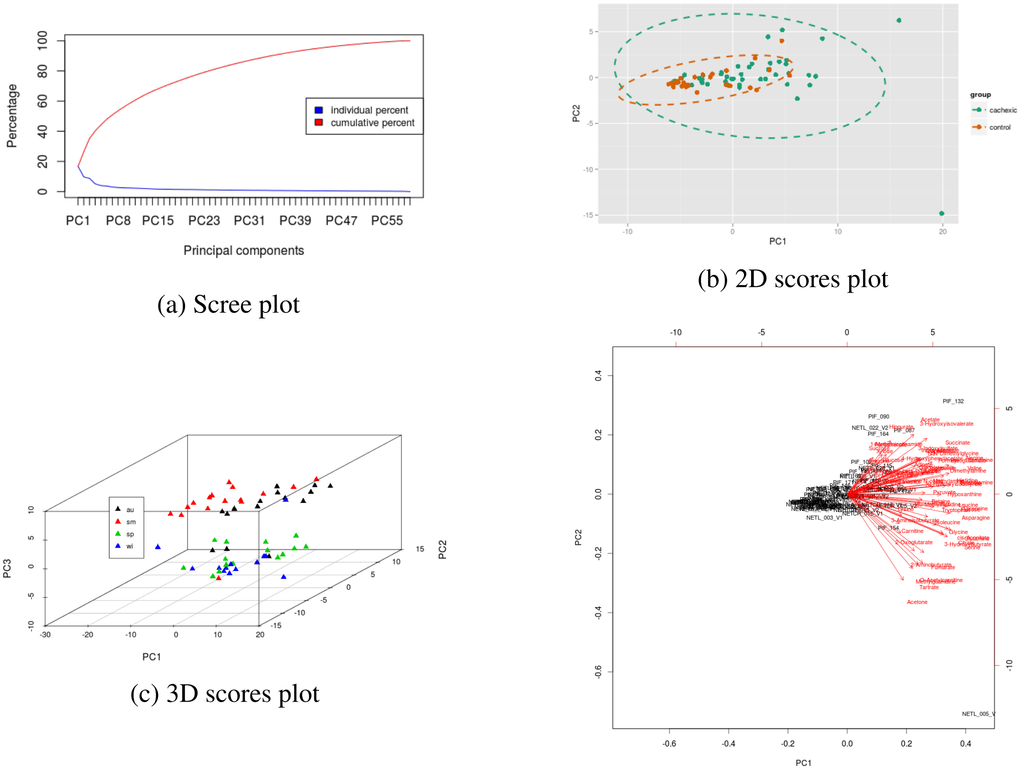
3.6.1 *Dimensionality reduction*

The package includes functions to perform PCA and a number of ways to visualize the results. To function *pca.analysis.dataset* allows to perform a PCA analysis, which uses the *prcomp* function from the already mentioned *stats* package. The function does the calculation by a Singular Value Decomposition (SVD) of the data matrix and the returned results include the standard deviations explained by the principal components (sdev), the matrix of variable loadings (rotation) and the scores matrix. Another alternative implemented is the *pca.robust* function that uses internally the *PCAgrid* function from the *pcaPP* package, which computes a desired number of robust principal components using the grid search algorithm in the plane (Filzmoser et al., 2014).

As said before, the package has implemented a number of ways to visualize the results and the available plots are:

- **Scree plot** - function *pca.screeplot* that shows the individual percentage of the explained variance of each principal component and the cumulative percentage;
- **Scores plot** - functions *pca.scoresplot2D*, *pca.scoresplot3D* and *pca.scoresplot3D.rgl* that show the scores of two different principal components in 2D and 3D;
- **Biplots** - functions *pca.biplot* and *pca.biplot3D* produce 2D and 3D biplots that display samples as points while the variables are displayed either as vectors, linear axes or nonlinear trajectories;
- **Pairs plot** - function *pca.pairs.plot* that produce a pairs plot of the scores of the defined principal components;
- **Other plots with *k-means* results** - functions that combine some plots mentioned above with *k-means* results for coloring the points according to the cluster they belong.

Some of the plots use different packages to give the plot a nicer look. The 2D scores plot, with the function *pca.scoresplot2D*, that allows to see the scores of 2 different principal components uses the *ggplot2* package and also draws the ellipses around the data points belonging to a certain group using the *ellipse* package. For the 3D scores plot and 3D biplot, the *rgl* package was used and allows to interactively move the plot to see from all the angles. The pairs plot uses the *GGally* package to show the result of a few principal components as well the correlations score of the different groups. In Figure 9 there is example of a few of these plots.



(a) Scree plot

(b) 2D scores plot

(c) 3D scores plot

(d) Biplot

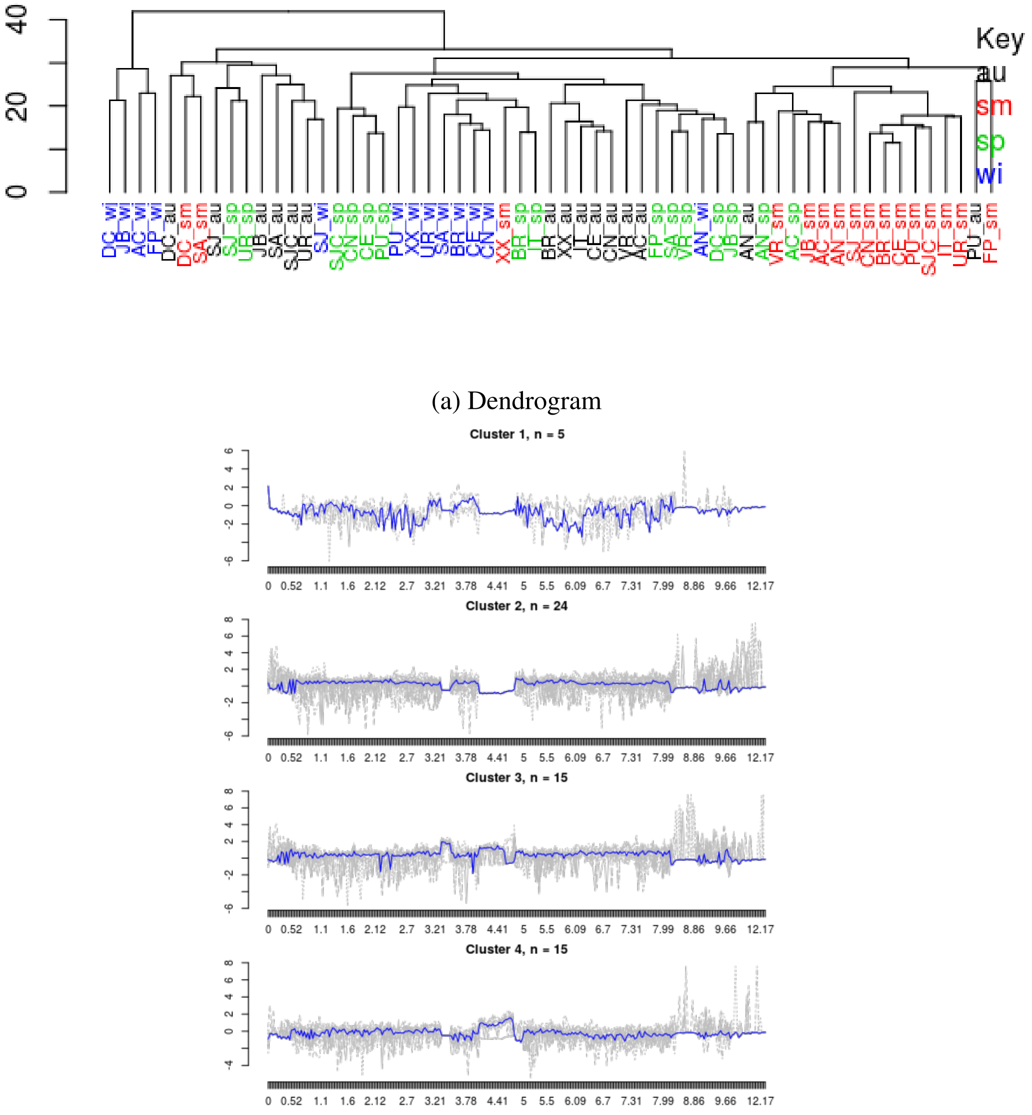
Figure 9: (a) PCA scree plot generated by *pca.screeplot* function. (b) 2D PCA scores plot (PC1 and PC2) from the function *pca.scoresplot2D*. (c) 3D PCA scores plot (PC1, PC2 and PC3) from the function *pca.scoresplot3D*. (d) PCA Biplot generated by *pca.biplot* function.

3.6.2 *Clustering*

Two clustering methods were implemented in the package: *k-means* clustering and HCA. To perform the specific methods, the functions *hierarchical.clustering* and *kmeans.clustering* are defined. We can choose the distance method according to *dist* function from *stats* package, as well the clustering method for the hierarchical clustering. For HCA, the available distance methods are: euclidean, maximum, manhattan, canberra, binary and minkowski (RCoreTeam, 2014b). For the agglomeration methods the available methods are: ward, single, complete, average, mcquitty, median and centroid (RCoreTeam, 2014a). For *k*-means the number of clusters must be given and the clustering can be done on samples or variables.

Both HCA and *k-means* functions use the *stats* package to perform the clustering analysis.

Clustering results can also be plotted. The functions *dendrogram.plot* and *dendrogram.plot.col* were developed to create a dendrogram of the HCA results (the last one coloring the leaves depending on what groups they belong defined from the selected metadata variable). The first one uses the *ggdendro* package. For *k-means*, the results can also be plotted using *kmeans.plot* function which shows in blue the median of the values of the samples in that cluster and in grey all the values of those samples. Also, the members of each cluster can be seen in a data frame with *kmeans.result.df* function. Figure 10 shows an example of a dendrogram and the result of *k*-means plot.



(a) Dendrogram

(b) *k*-means plot

Figure 10: (a) Dendrogram with colored leaves from *dendrogram.plot.col* function. (b) *k*-means plot showing four clusters.

3.7 MACHINE LEARNING AND FEATURE SELECTION

The package provides a number of functions to train, use and evaluate machine learning methods, being mostly based in the R package *caret*. A simple process of training and prediction for new examples is implemented by function *train.and.predict*, while function *train.models.performance* allows to estimate the error for a set of classifiers. Also, there are functions to evaluate the importance of each variable in the prediction models.

To train a model, the *train* function of the package *caret* is used internally, allowing to choose the parameters for the training process, namely the error estimation method, the number of folds or resampling iterations, the number of repetitions (only for repeated cross validation). The options for the estimation method can be seen in Table 4 (Kuhn et al., 2014).

Keyword	Name
boot	Bootstrap
boot632	632 Bootstrap
cv	Cross validation
repeatedcv	Repeated cross validation
LOO	Leave one out
LGOCV	Leave group out cross validation
none	Fits the model to the entire training set
oob	Out-of-bag

Table 4: Resampling methods

The *train* function executes an internal process of parameter optimization over the internal parameter(s) of the selected classifier. The values tested for each parameters can be given as a data frame with the columns named with the same name as the tuning parameters. A list of possible models and their tunable parameters can be seen in this URL: `http://topepo.github.io/caret/modelList.html`. A number indicating just the number of levels for each tuning parameters can also be given (tune length).

A set of some of the most common models are shown in Table 5 together with the internal parameters that can be optimized by the function.

Keyword	Name	Tuning parameters
pls	Partial Least Squares	Number of components (ncomp)
J48	C4.5-like Trees	Pruning confidence (C)
JRip	Rule-Based Classifier	Number of optimizations (NumOpt)
svmLinear	Support Vector Machines with Linear Kernel	Misclassifying influence (C)
rf	Random Forest	Number of variables randomly sampled as candidates at each split (mtry)

Table 5: Some models with their tuning parameters.

The two approaches for machine learning, classification and regression, are both implemented. The metrics for each one are the accuracy and Kappa statistic for classification, and for regression there is the Root Mean Square Error (RMSE) and the coefficient of determination (R^2).

A grid of parameters is created for each model and the model is trained on slightly different data for each combination of tuning parameters. With each subset, the performance of the held-out samples is calculated and the mean and standard deviation are averaged over cross-validation iterations. The combination with the optimal resampling statistic is chosen to be the final model and all training set is used to fit that model (Kuhn et al., 2014).

The returning values for training a model include the performance result for the best model, the variable's importance for each model, the full results with all the combinations of the tuning parameters tested with the accuracy averaged over cross-validation results, the confusion matrices (in case of classification) and the final models. These can be used later for predictions or visualization, as it happens in the PLS example, where a 3D plot of the first three components can be shown, as it is the case in Figure 11.

As mentioned above, the results include the variable's importance for each model. This is calculated with the *varImp* function from the *caret* package. It returns for each category of the metadata variable, the importance that the variable has in the model (in case of classification) or/and the overall importance. With the function *summary.var.importance*, we can summarize the most important variables for each model.

For classification purposes, the confusion matrix is calculated for each model. The confusion matrix allows to visualize the performance of a model, it shows how many samples were correctly classified, as well the misclassified ones (allowing to see to which class the sample was incorrectly classified). Each column of the matrix represents the instances in an actual class and the rows represent the instances in a predicted class.

Also, the package provides a number of functions to do feature selection, i.e. determine which attributes are more valuable when applying different machine learning methods. The high level function *feature.selection* allows to perform this analysis. The methods of RFE and filtering are made available also from the *caret* package.

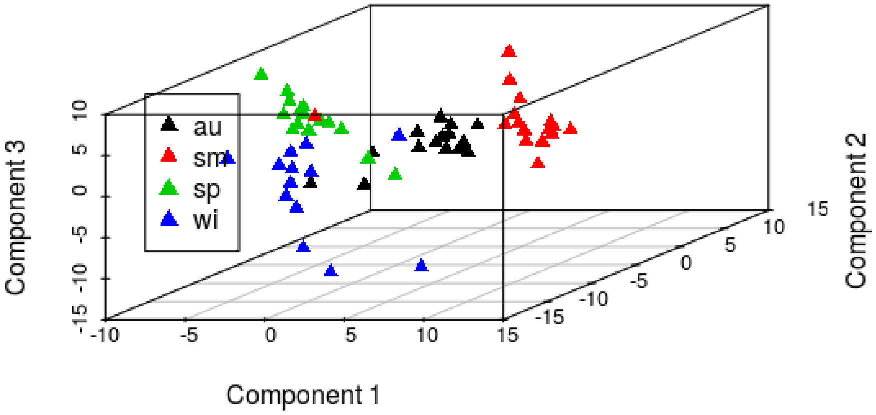


Figure 11: 3D scatter plot of the first 3 components from a PLS model.

The RFE method, that uses the *caret* function *rfe*, requires a vector with the number of the subset sizes that are going to be tested, the resampling method and the set of functions for model fitting, prediction and variable importance. The functions can be the helper functions for backwards feature selection from *caret* package, which are the linear model's functions (lmFuncs), random forest's functions (rfFuncs), linear discriminating analysis' functions (ldaFuncs), naive bayes' functions (nbFuncs) and linear regression functions (lrFuncs). The resampling method can be one of the already mentioned in Table 4 (Kuhn et al., 2014). The result from RFE includes the performance from each subset with the metrics already mentioned above, with the best model selected and the variables that were selected.

With the feature selection done with filtering, also from the *caret* package, the function *sbf* is used internally to do feature selection using univariate filters. Also the method of resampling and a set of functions for model fitting, prediction and variable filtering can be defined. The helper functions for selection by filtering can be the linear model's functions (lmSBF), random forest's functions (rfSBF), tree bag's functions (treebagSBF), linear discriminant analysis' functions (ldaSBF) and naive bayes' functions (nbSBF). (Kuhn et al., 2014)

The returning results include the resampling performance with the metrics mentioned earlier, the variables selected and the percentage of the variables selected during the resample.

4

CASE STUDIES

In this chapter, three case studies using real data will be presented to put to the test the package developed and provide meaningful data analysis pipelines. The first case study is the discrimination of propolis samples from southern Brazil (using NMR and UV-vis data), the next case study is the analysis of metabolite's concentrations of urine samples from control and cachexic cancer patients and the last one is the analysis of the effect of PPD of cassava samples.

4.1 PROPOLIS

4.1.1 *Introduction*

Propolis or bee glue is a sticky dark-colored substance produced from the collected buds or exudates of plants (resin) by bees (*Apis mellifera L.*). The resin is masticated, salivary enzymes are added, and the partially digested material is mixed with beewax and used in the hive to seal the walls, strengthen the borders of combs, and embalm dead invaders (Wollenweber et al., 1990). Humans have used propolis as a remedy since ancient times (Marcucci, 1995). In the last years, this product has been the subject of intensive studies highlighting its biological and pharmacological properties, such as the antimicrobial (Burdock, 1998; Banskota et al., 2001; Yildirim et al., 2004; G. Vardar-Ünlü and Ünlü, 2008), anti-oxidative (Kumazawa et al., 2007), anti-viral (Gekker et al., 2005), anti-tumoral (Sforcin, 2007; Tan-No et al., 2006; Awale et al., 2008), anti-inflammatory (Burdock, 1998; Banskota et al., 2001), and anti-neurodegenerative (Chen et al., 2008). Propolis was also tested as a food preserver, due to its bactericidal and bacteriostatic properties and, as its components are natural constituents of food, are recognized as safe substances (Tosi et al., 2007).

The successful medical applications of propolis led to an increased interest in its chemical composition. In general, resin comprising flavonoids and related phenolic acids represent approximately half of the propolis constituents, while beewax, volatiles, and pollen represent approximately 30%, 10%, and 5%, respectively (Bankova et al., 2000). Still, the chemical composition of the bee glue is extremely dependent on the plants found around the hive, as well as on the geographic and climatic characteristics of the site. Buds from Populus species are the main source of resins in Europe and North America propolis ("poplar type" propolis) (Marcucci, 1995). Alternatively, in regions where these plants are not native, other species from the genera Clusia in Cuba and Baccharis in Brazil are used as resin sources, increasing its diversity and complexity (Salatino et al., 2005). Less commonly, species from genera such as *Betula*, *Ulmus*, *Pinus*, *Quercus*, *Salix* and *Acacia* are also used (König, 1985).

It has long been known that propolis' chemical composition might be strongly influenced by environmental factors peculiar to the sites of collection of a given geographic region of production, as well as by seasoning. Together with chemometrics techniques, the aim of this case study is to gain insights of important features associated to chemical composition, harvest season, and geographic origin of propolis produced in the Santa Catarina state, southern Brazil.

4.1.2 *NMR data*

The propolis samples used in this study for NMR data analysis were collected in the autumn (AU), winter (WI), spring (SP), and summer (SM) of 2010 from *Apis mellifera* hives located in southern Brazil (Santa Catarina State). A total of 59 samples were collected, and the distribution of samples by seasons being: SM – 16 samples, AU and SP – 15 samples, WI – 13 samples.

Also, three agroecological regions were defined for the different apiaries, and one distributed as follows: Highlands – 12 samples, Plain – 11 samples, Plateau – 36 samples.

After a few steps to acquire the data from the samples through NMR, the data was stored for further analysis. Each sample is represented by a .csv file which contains the chemical shifts (ppm) and its corresponding value, and the name of the file contains the apiary and the season separated by underscore ('_'). For example, "SJ_wi.csv", means that the apiary is 'SJ' and the season is 'wi' (winter). Then, to assign a agroecological region from a apiary, a function was used to make that conversion (in the example, 'SJ', was converted to 'Highlands'). The files were then read to a peaks list in R.

An internal standard, Trimethylsilyl Propionate Sodium salt (TSP, 0.024 g%) at peak 0.00 ppm and resonances regions at 3.29-3.31 ppm and 4.85-5.00 ppm containing methanol-d_4 and water signals were removed from the dataset for further analysis. After the resonances removal, the peaks list is transformed into a dataset using peak alignment for further analysis.

The next step of preprocessing was to remove variables that contained more than 75% of missing values, replacing the missing values with a low value (5e-05) and then applying a logarithmic transformation and data autoscaling. In this step, the options of no logarithmic transformation and no auto scaling; only logarithmic transformation; only auto scaling; and both logarithmic transformation and posterior auto scaling were tested to evaluate different results. Both metadata variables, seasons and agroecological regions, were used in the analysis. The full results can be found in the reports generated by Markdown RStudio plugin that are provided as supplementary material in `darwin.di.uminho.pt/chris-msc`. The analysis that will be referenced here will be the one with logarithmic transformation and auto scaling.

4.1.2.1 *Univariate Analysis*

One-way ANOVA and Tukey's HSD post-hoc tests were used to detect significant statistical differences (*p-value* below 0.05) derived from the effects of propolis harvest seasons or agro-regions on the propolis spectral profiles. These were done for all peaks and the top 20 were ordered by increasing *p-value* as can be seen in Table 6.

More information about these methods is available in section 2.2.

ppm	pvalues	logs	fdr	tukey
4.66	9.585e-26	25.018	2.319e-23	sm-au; sp-sm; wi-sm
4.58	3.385e-17	16.470	4.096e-15	sm-au; sp-sm; wi-sm
4.55	6.092e-14	13.215	4.915e-12	sm-au; sp-au; wi-au; sp-sm; wi-sm
4.63	1.044e-13	12.981	6.316e-12	sm-au; sp-sm; wi-sm
4.71	2.083e-13	12.681	1.008e-11	sm-au; sp-sm; wi-sm
4.5	2.643e-13	12.578	1.066e-11	sp-au; wi-au; sp-sm; wi-sm
4.08	1.216e-12	11.915	4.204e-11	sp-au; wi-au; sp-sm; wi-sm
4.45	2.447e-12	11.611	7.026e-11	sp-au; wi-au; sp-sm; wi-sm
4.17	2.613e-12	11.583	7.026e-11	sp-au; wi-au; sp-sm; wi-sm
4.31	4.227e-12	11.374	1.023e-10	sp-au; wi-au; sp-sm; wi-sm
4.53	1.044e-11	10.981	2.297e-10	sm-au; sp-au; wi-au; sp-sm; wi-sm
4.02	6.610e-11	10.180	1.333e-09	sp-au; wi-au; sp-sm; wi-sm
4.38	8.033e-11	10.095	1.495e-09	sp-au; wi-au; sp-sm; wi-sm
4.05	1.505e-10	9.823	2.601e-09	sp-au; wi-au; sp-sm; wi-sm
4.28	2.623e-10	9.581	4.231e-09	sp-au; wi-au; sp-sm; wi-sm
4.25	3.869e-10	9.412	5.637e-09	sp-au; wi-au; sp-sm; wi-sm
4.34	3.960e-10	9.402	5.637e-09	sp-au; wi-au; sp-sm; wi-sm
4.2	1.338e-09	8.873	1.800e-08	sp-au; wi-au; sp-sm; wi-sm
4.13	4.539e-09	8.343	5.781e-08	sp-au; wi-au; sp-sm; wi-sm
4.74	2.695e-08	7.569	3.261e-07	sm-au; sp-sm; wi-sm

Table 6: ANOVA results with seasons metadata.

The features from the Table 6 indicate that compounds with anomeric structural moieties appear to have a significant effect on the discrimination of propolis samples over the seasons, because all the main resonances selected in the univariate analysis occur at the anomeric region of the spectra (3.00 ppm - 5.50 ppm).

4.1.2.2 *Clustering*

Clustering was also used, more exactly *hierarchical clustering* and *k-means*. Figure 12 shows a dendrogram resulting from *hierarchical clustering*, while Figure 13 shows a plot of *k-means* clustering, showing in blue the median of the values of the samples in that cluster and in grey all the values of that samples. Table 7 shows the members of each cluster.

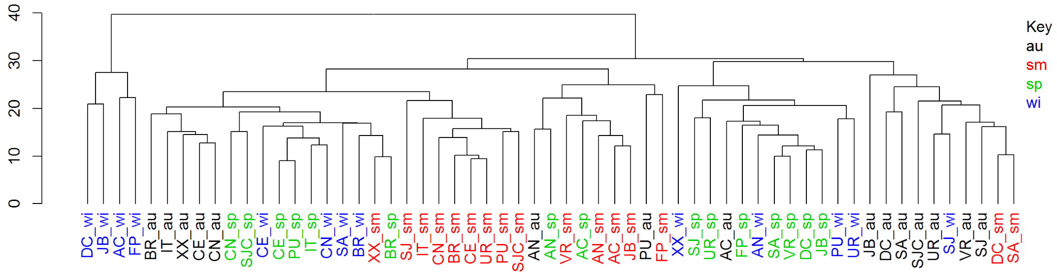


Figure 12: Dendrogram with season color labels.

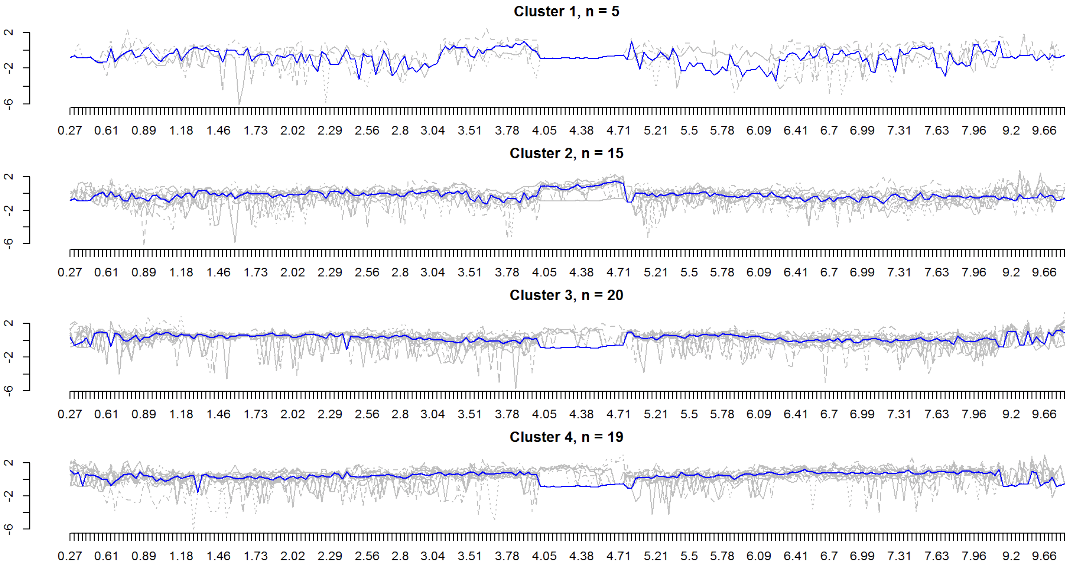


Figure 13: *k-means* plot with four clusters.

Cluster	Samples
1	AC_sp AC_wi DC_wi FP_wi JB_wi
2	AN_au PU_au AC_sm AN_sm BR_sm CE_sm CN_sm FP_sm IT_sm JB_sm SJ_sm SJC_sm UR_sm VR_sm AN_sp
3	BR_au CE_au CN_au IT_au SJC_au XX_au PU_sm XX_sm BR_sp CE_sp CN_sp IT_sp PU_sp SJC_sp BR_wi CE_wi CN_wi PU_wi SA_wi XX_wi
4	AC_au DC_au JB_au SA_au SJ_au UR_au VR_au DC_sm SA_sm DC_sp FP_sp JB_sp SA_sp SJ_sp UR_sp VR_sp AN_wi SJ_wi UR_wi

Table 7: Members of the clusters of *k-means*.

In the dendrogram, there was a reasonable separation for the seasons of propolis as we can see by the colors in the figure. In *k-means* four clusters were used and there was some separation between the seasons, mostly the summer samples and as we can see from the plot of *k-means*, the median of each cluster's values of the samples was not too different from each other.

To see more information about these methods see subsection 2.2.4.

4.1.2.3 *PCA*

In order to reduce the dimensionality of the dataset, a PCA analysis was run on the propolis data. Some plots were produced, including the 2D scores plot shown in Figure 14 and the scree plot in Figure 15.

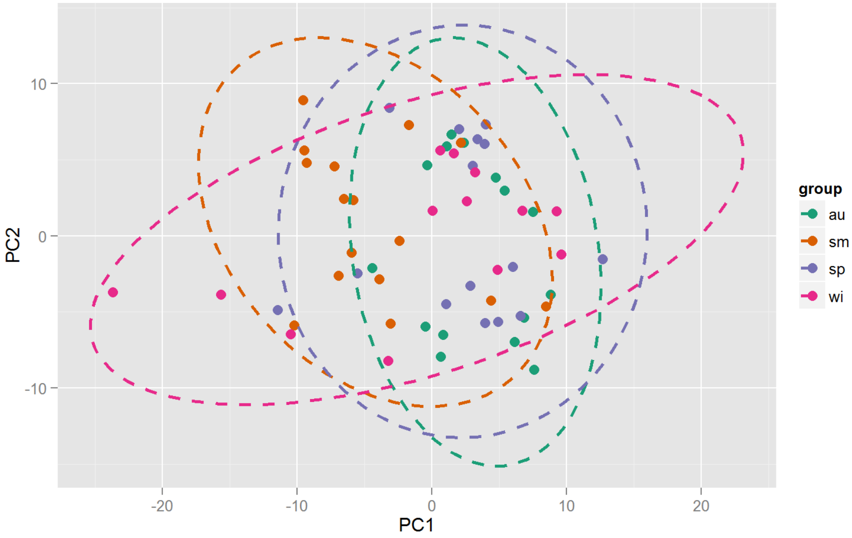


Figure 14: PCA scores plot (PC1 and PC2) grouped by seasons.

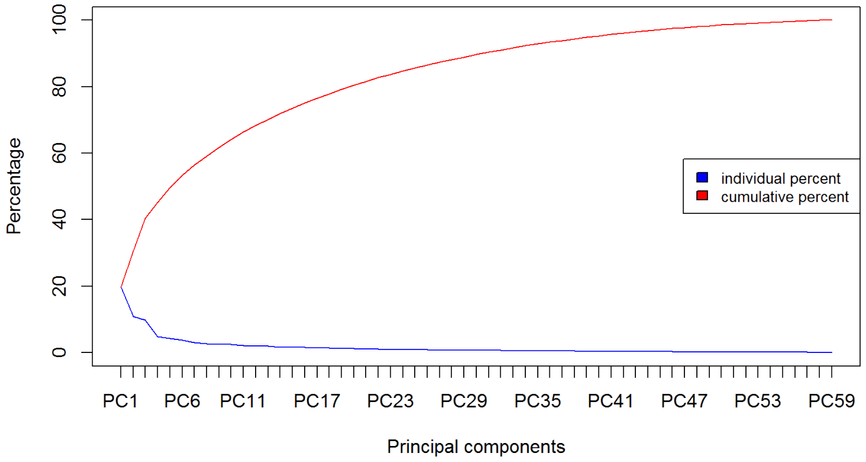


Figure 15: PCA scree plot.

The variance explained by the first two principal components was 30.66% and as we can see the separation from the PCA 2D scores plot was not so good, with some overlapping. In comparison to Figure 16, which shows the separation between the four clusters of *k-means*, we can see that the separation of the second one was much better.

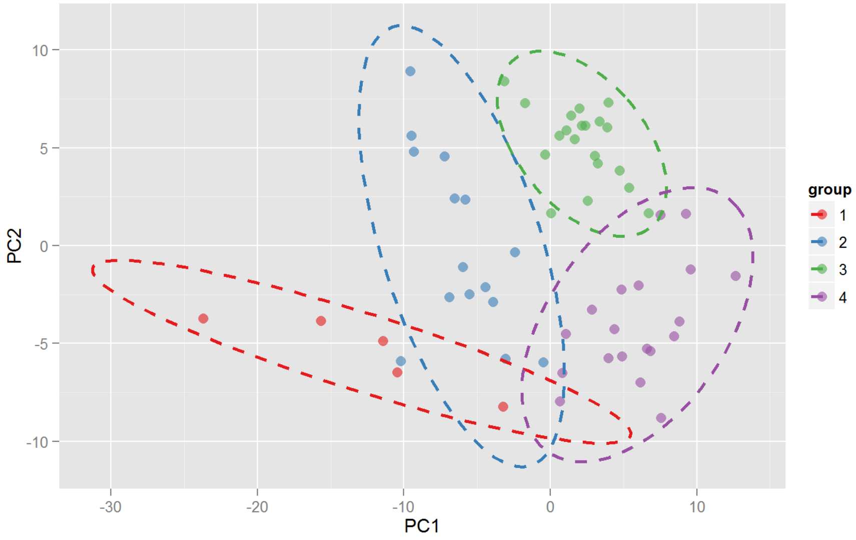


Figure 16: PCA scores plot (PC1 and PC2) grouped by the clusters from k-means.

4.1.2.4 *Machine Learning*

Classification models were built to try to discriminate samples by season and by agroecological region. Five models were tested with repeated cross-validation with 10 folds, 10 repeats, and a tune length of 20 levels, to see how well they performed in samples discrimination. The results for seasons and agroecological regions are shown in Table 8 and 9.

Model	Accuracy	Kappa	AccuracySD	KappaSD
pls	0.8678	0.8210	0.1264	0.1709
J48	0.7346	0.6433	0.1446	0.1891
JRip	0.5541	0.4013	0.2097	0.2719
svmLinear	0.8263	0.7620	0.1626	0.2247
rf	0.8456	0.7908	0.1364	0.1841

Table 8: Classification models result with seasons metadata.

Model	Accuracy	Kappa	AccuracySD	KappaSD
pls	0.7905	0.5647	0.16019	0.3383
J48	0.5931	0.2579	0.18273	0.3119
JRip	0.6115	0.0000	0.06031	0.0000
svmLinear	0.6728	0.3844	0.18719	0.3493
rf	0.7463	0.4548	0.14261	0.3173

Table 9: Classification models result with agroregions metadata.

As we can see from the Tables 8 and 9, the *pls* model gave the best results and the seasons metadata gave the best discrimination, with 86.78% accuracy with the Kappa statistic value of 0.82. In Tables 10 and 11, we can see how many components were tested with *pls* and their results, with 20 and 4 being the number of components that gave the best result with seasons and agroregions, respectively. The standard deviation on both accuracy and Kappa statistic were low, with the Kappa statistic's standard deviation from agro-regions metadata being slightly higher.

Number of components	Accuracy	Kappa	AccuracySD	KappaSD
1	0.4392	0.2455	0.1401	0.1691
2	0.6157	0.4853	0.1963	0.2619
3	0.7372	0.6445	0.1820	0.2439
4	0.7925	0.7206	0.1635	0.2162
5	0.8349	0.7775	0.1587	0.2111
6	0.8209	0.7570	0.1476	0.1990
7	0.8212	0.7588	0.1440	0.1933
8	0.8308	0.7716	0.1370	0.1837
9	0.8418	0.7858	0.1355	0.1824
10	0.8534	0.8021	0.1338	0.1786
11	0.8551	0.8040	0.1282	0.1718
12	0.8602	0.8109	0.1296	0.1737
13	0.8585	0.8082	0.1272	0.1722
14	0.8565	0.8054	0.1266	0.1713
15	0.8575	0.8071	0.1304	0.1760
16	0.8628	0.8141	0.1303	0.1762
17	0.8614	0.8122	0.1296	0.1752
18	0.8630	0.8144	0.1281	0.1733
19	0.8630	0.8145	0.1281	0.1733
20	0.8678	0.8210	0.1264	0.1709

Table 10: Full results using pls with different number of components with seasons metadata.

Number of components	Accuracy	Kappa	AccuracySD	KappaSD
1	0.6520	0.1846	0.1319	0.2962
2	0.7158	0.4151	0.1819	0.3609
3	0.7582	0.4876	0.1694	0.3629
4	0.7905	0.5647	0.1602	0.3383
5	0.7864	0.5587	0.1648	0.3456
6	0.7698	0.5383	0.1650	0.3291
7	0.7371	0.4703	0.1745	0.3549
8	0.7103	0.4313	0.1870	0.3650
9	0.6792	0.3769	0.1798	0.3520
10	0.6751	0.3707	0.1869	0.3641
11	0.6629	0.3487	0.1983	0.3741
12	0.6768	0.3747	0.1867	0.3615
13	0.6714	0.3618	0.1902	0.3701
14	0.6769	0.3725	0.1876	0.3649
15	0.6711	0.3656	0.1831	0.3597
16	0.6736	0.3722	0.1777	0.3474
17	0.6700	0.3671	0.1774	0.3446
18	0.6629	0.3553	0.1771	0.3417
19	0.6623	0.3552	0.1777	0.3429
20	0.6614	0.3544	0.1807	0.3469

Table 11: Full results using pls with different number of components with agroregions metadata.

In Tables 12 and 13 the confusion matrices of the *pls* model are shown (the entries are percentages of table totals), while Figures 17 and 18 display the 3D plots of the first 3 components of the *pls* model with seasons and agroregions metadata, respectively.

	Reference			
Prediction	au	sm	sp	wi
au	17.9	0.0	1.7	1.9
sm	0.8	25.4	0.1	0.0
sp	3.0	1.8	23.5	0.3
wi	3.5	0.0	0.0	19.9

Table 12: Confusion matrix of the pls model with seasons metadata

	Reference		
Prediction	Highlands	Plain	Plateau
Highlands	9.8	0.8	2.5
Plain	0.3	12.5	1.9
Plateau	10.1	5.3	56.7

Table 13: Confusion matrix of the pls model with agroregions metadata

A quick look over the confusion matrices shows that the summer season has a low percentage of incorrect predictions, while the Highlands agroregion has the highest number of incorrect predictions.

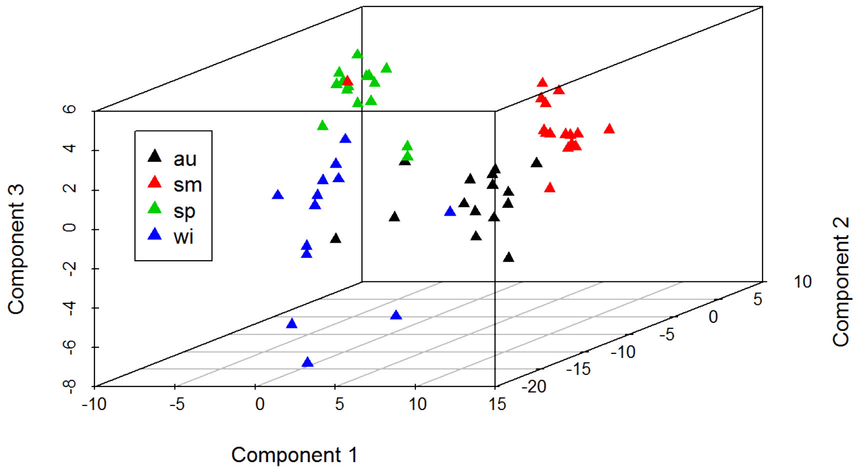


Figure 17: 3D plot of the first 3 components of the pls model with seasons metadata

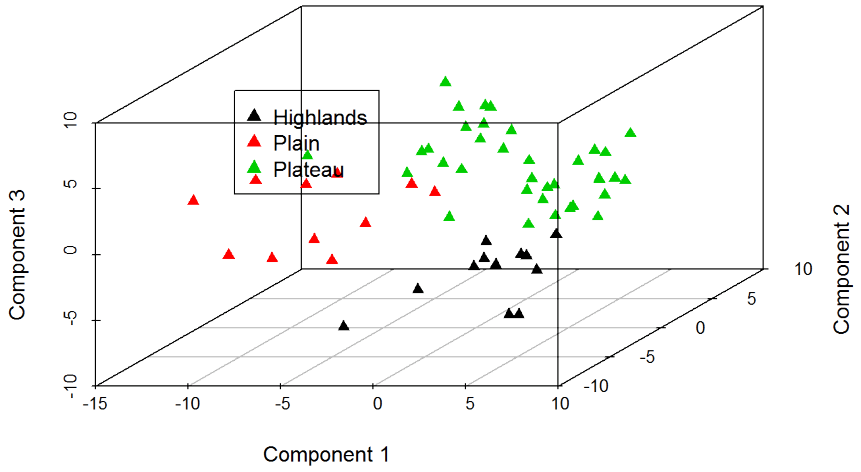


Figure 18: 3D plot of the first 3 components of the pls model with agroregions metadata

With the 3D plots of the first 3 components of the *pls* model we can see a clear separation of the classes, forming 4 different clusters with the seasons metadata, and 3 in agroregions metadata.

Also, in Table 14 we can see the top 10 variables ordered by its importance. It is easy to check that the compounds with anomeric structural moieties have a significant effect on the discrimination of propolis samples over the seasons in the *pls* model, because all the top resonances are in the anomeric region of the spectra locates between 3.00 ppm and 5.50 ppm (except for the 7^{th} resonance - 5.62). These results are quite consistent with the results given by *one-way ANOVA and Tukey's HSD post-hoc test* that were mentioned before.

ppm	au	sm	sp	wi	mean
4.66	48.10	100.00	72.37	46.51	66.75
4.58	37.27	91.23	51.67	33.38	53.39
4.71	33.47	85.05	46.91	32.92	49.59
4.84	44.30	56.08	68.76	25.77	48.73
4.63	32.72	83.80	46.17	32.10	48.70
3.93	52.65	30.79	61.17	45.38	47.50
5.62	51.43	21.89	63.70	48.43	46.36
3.9	56.94	18.46	54.90	44.24	43.64
4.28	39.66	57.48	37.40	35.47	42.50
4.17	39.97	58.44	39.33	32.06	42.45

Table 14: Variable importance of the pls model with seasons metadata

4.1.3 *UV-Vis data*

The propolis samples used in UV-vis analysis were collected on the final of every season of the years 2010 and 2011, with the collection starting in the autumn of 2010 and finishing in the summer of 2012. A total of 133 samples were collected, and the distribution of samples by seasons being: inv – 31 samples (winter), out – 35 samples (autumn), pri – 34 samples (spring), ver – 33 samples (summer).

Also, three different regions were sampled from the agroecological regions and distributed as follows: Planalto – 70 samples, Planicie – 29 samples, Serra – 29 samples.

From the year of 2010, there were 72 samples, while 61 samples were available for 2011, having three metadata variables that can be used in the analysis, the seasons, agroregions and the years.

The same principle from NMR was applied to UV data for getting the metadata from the samples names. Depending on which metadata variable was used, samples with no information on that variable were removed. Also an absorbance value which was an outlier from wavelength 529 was corrected. Two preprocessing strategies were used, using the data as it is with no preprocessing and the data corrected with background correction, offset correction and baseline correction. The results from the pipeline with no preprocessing will be shown here. The results from each one with all 3 metadata's variables used are provided as supplementary material in `darwin.di.uminho.pt/chris-msc`.

4.1.3.1 *Univariate Analysis*

The one-way ANOVA and Tukey's HSD post-hoc test results can be seen in Table 15.

wavelength	pvalues	logs	fdr	tukey
293	1.831e-07	6.737	4.707e-05	out-inv; pri-inv; ver-pri
292	1.879e-07	6.726	4.707e-05	out-inv; pri-inv; ver-pri
288	6.836e-07	6.165	1.142e-04	out-inv; pri-inv; ver-pri
295	1.130e-06	5.947	1.402e-04	out-inv; pri-inv; ver-pri
291	1.420e-06	5.848	1.402e-04	out-inv; pri-inv; ver-pri
294	1.679e-06	5.775	1.402e-04	out-inv; pri-inv; ver-pri
290	2.380e-06	5.624	1.703e-04	out-inv; pri-inv; ver-pri
296	3.312e-06	5.480	2.074e-04	out-inv; pri-inv; ver-pri
289	5.519e-06	5.258	3.072e-04	out-inv; pri-inv; ver-pri
297	1.084e-05	4.965	5.430e-04	out-inv; pri-inv; ver-pri

Table 15: ANOVA results with seasons metadata.

The results from Table 15 indicate that the wavelength region between 288 and 297 nm have significant effect in discriminating the propolis samples over the season.

4.1.3.2 *Clustering*

Figure 19 shows the dendrogram resulting from hierarchical clustering with the regions as label colors.

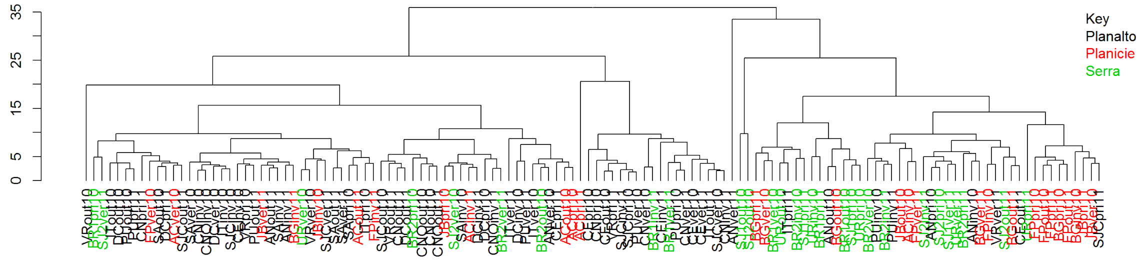


Figure 19: Dendrogram with regions as label colors.

As we can see from the dendrogram above, the samples are reasonably grouped by the 3 regions.

4.1.3.3 *PCA*

The 2D plot scores (Principal Component (PC)1 and PC2) from PCA analysis, using the seasons metadata to color the points, is shown in Figure 20 and as we can see, there is too much overlapping between the 4 clusters, thus not making a good discrimination technique for the seasons.

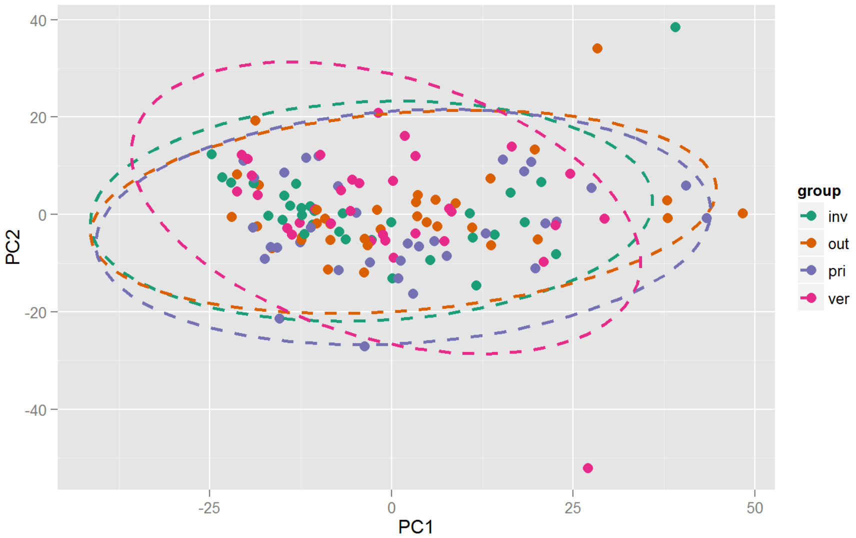


Figure 20: PCA 2D scores plot (PC1 and PC2) grouped with seasons metadata.

4.1.3.4 *Machine learning*

The results from the classification models for seasons, regions and years with repeated cross-validation with 10 folds and 10 repeats are shown in Tables 16, 17 and 18.

Model	Accuracy	Kappa	AccuracySD	KappaSD
pls	0.4208	0.2297	0.1347	0.1775
J48	0.4173	0.2225	0.1291	0.1700
JRip	0.3317	0.1122	0.1174	0.1547
svmLinear	0.4192	0.2245	0.1180	0.1584
rf	0.4586	0.2791	0.1392	0.1843

Table 16: Classification models result with seasons metadata.

Model	Accuracy	Kappa	AccuracySD	KappaSD
pls	0.7368	0.5380	0.1214	0.2150
J48	0.5900	0.3036	0.1300	0.2125
JRip	0.6361	0.3441	0.1252	0.2273
svmLinear	0.6467	0.3823	0.1221	0.2244
rf	0.6856	0.4554	0.1158	0.2022

Table 17: Classification models result with regions metadata.

Model	Accuracy	Kappa	AccuracySD	KappaSD
pls	0.6280	0.2456	0.1173	0.2367
J48	0.5662	0.1225	0.1255	0.2548
JRip	0.6473	0.2790	0.1255	0.2605
svmLinear	0.5581	0.1060	0.1191	0.2404
rf	0.6584	0.3075	0.1342	0.2702

Table 18: Classification models result with years metadata.

As we can see from the tables above, the regions had the best results as a way of discriminating the samples, with 73,68% being the maximum accuracy by the *pls* model. The years had intermediate results, being 64,75% the best accuracy result from the *JRip* model. On the other hand, the seasons had the worst results with 44,20% accuracy result from *rf* model being the best result, and so it was not possible to create a reliable predictive model to predict the seasons from the UV-vis propolis samples.

4.2 CACHEXIA

4.2.1 *Introduction*

Cachexia is a complex metabolic syndrome associated with an underlying illness (such as cancer) and characterized by loss of muscle with or without loss of fat mass (Evans et al., 2008).

Improved approaches for detecting the onset and evolution of muscle wasting would help to manage wasting syndromes and facilitate early intervention. Dual energy X-ray Absortiometry (DXA), Computed Tomography (CT) and Magnetic Resonance Imaging (MRI) are considered the most precise measures of adipose and muscle tissues currently available, but there are several limitations like the access and cost, the time consumed, and CT expose patients to radiation. So, clinicians want to find new approaches for identifying and monitoring muscle loss that are faster, cheaper, safer and more accessible (Eisner et al., 2010).

4.2.2 *Concentrations data*

To try a different approach, as metabolites produced from tissue breakdown are likely to be a sensitive indicator of muscle wasting, urine samples were collected since several end products of muscle catabolism are specifically excreted in urine (Eisner et al., 2010). A total of 77 urine samples were collected being 47 of them patients with cachexia, and 30 control patients. All one-dimensional NMR spectra of urine samples were acquired and then the metabolites were detected and quantified, i.e. for each metabolite its concentration was measured.

Two different pipelines were tested to check the results, the first with no preprocessing whatsoever on the data and the second one with log transformation and auto scaling. The one with preprocessing will be used here, except for the fold change analysis (due to preprocessing changing the absolute values). The results from the pipeline with no preprocessing, as well as the full results from the other are provided in the supplementary material in `darwin.di.uminho.pt/chris-msc`.

To see more information about the methods used in preprocessing and data analysis, please refer to see section 2.2.

4.2.2.1 *Univariate Analysis*

T-tests were performed on the dataset for all variables (compounds) and the top 10 results, ordered by increasing *p-value*, are shown in Table 19.

Compound	p.value	-log10	fdr
Quinolinate	3.452e-06	5.462	0.0002175
Glucose	1.644e-05	4.784	0.0002758
3-Hydroxyisovalerate	1.884e-05	4.725	0.0002758
Leucine	1.955e-05	4.709	0.0002758
Succinate	2.861e-05	4.544	0.0002758
Valine	3.050e-05	4.516	0.0002758
N.N-Dimethylglycine	3.373e-05	4.472	0.0002758
Adipate	3.502e-05	4.456	0.0002758
myo-Inositol	3.982e-05	4.400	0.0002787
Acetate	6.945e-05	4.158	0.0004149

Table 19: t-Tests results ordered by p-value.

In Figure 21 compounds that have a *p-value* < 0.0001 are shown in blue, and the compounds with *p-value* $>= 0.0001$ appear in grey.

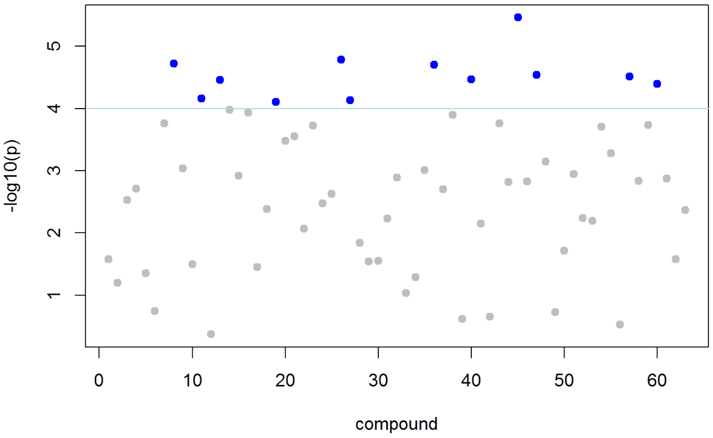


Figure 21: *t*-tests plot from the pipeline.

The results from the Table 19 show which compounds have a significant differential value in discriminating between the cachexic and control patients.

FC analysis was also done with this dataset, and the top 10 results, i.e. the ones with the highest fold changes, are shown in Table 20, by decreasing value of FC (absolute value).

Compound	Fold Change	log2(FC)
Glucose	5.869	2.553
Adipate	3.872	1.953
Creatine	3.396	1.764
Lactate	3.310	1.727
cis-Aconitate	3.009	1.589
3-Hydroxybutyrate	2.956	1.564
myo-Inositol	2.903	1.537
Trigonelline	2.752	1.460
Sucrose	2.699	1.433
Succinate	2.669	1.416

Table 20: Fold change results ordered by log_2 of fold change values from the data.

The plot in Figure 22 shows the compounds where the fold change value is above the defined threshold of 3 (points in blue) and the compounds below the threshold (points in grey).

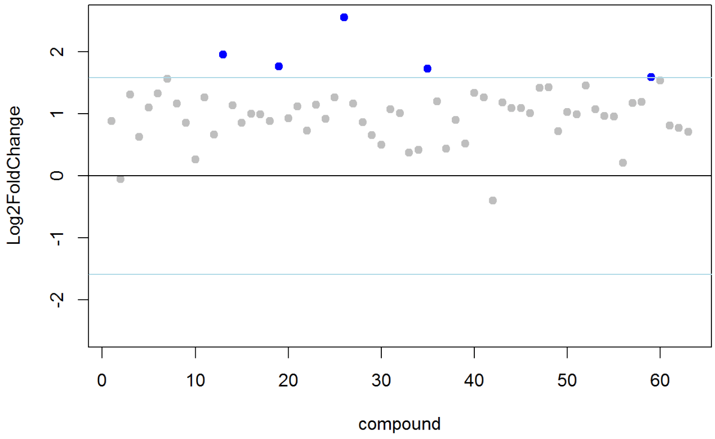


Figure 22: Fold change plot from the cachexia dataset.

Table 20 shows the compounds with highest absolute value change between the cachexic and control means.

Also, a volcano plot in Figure 23 shows the combination of fold change and t-tests with a threshold of 3 and 0.0001, respectively.

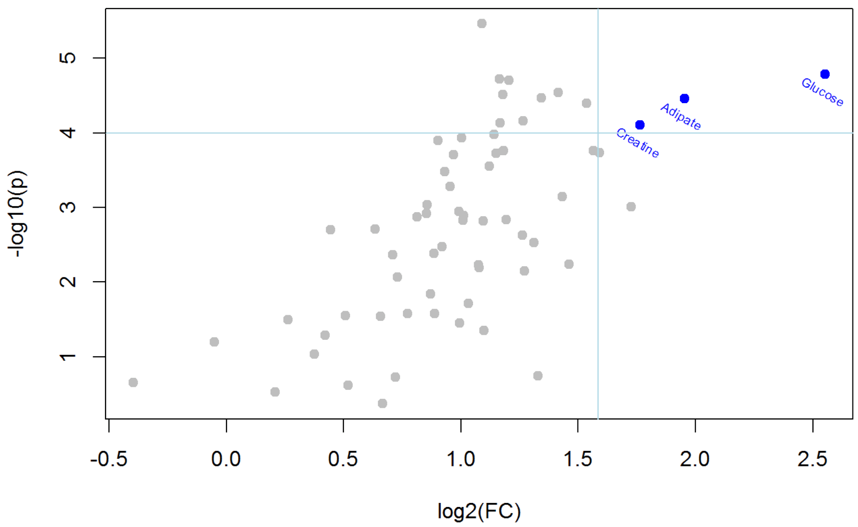


Figure 23: Volcano plot from the t-tests and fold change.

The volcano plot shows that 3 compounds (Creatine, Adipate and Glucose) obey to both thresholds defined.

4.2.2.2 *Clustering*

Although not perfect, the resulting dendrogram from hierarchical clustering is shown in Figure 24 with distinct colors for the different classes in the leaves, we can observe some sort of separation between the cachexic and control samples.

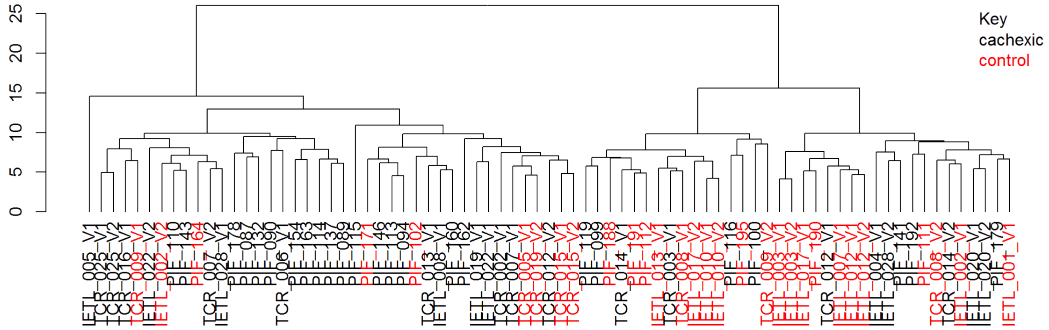


Figure 24: Resulting dendrogram from hierarchical clustering.

4.2.2.3 *PCA*

A PCA analysis was performed on the dataset and the Figure 25 shows the scree plot, showing the percentage of variance that each component explains, as well the cumulative percentage. Also in Figure 26, there is a 2D scores plot with principal components 1 and 2.

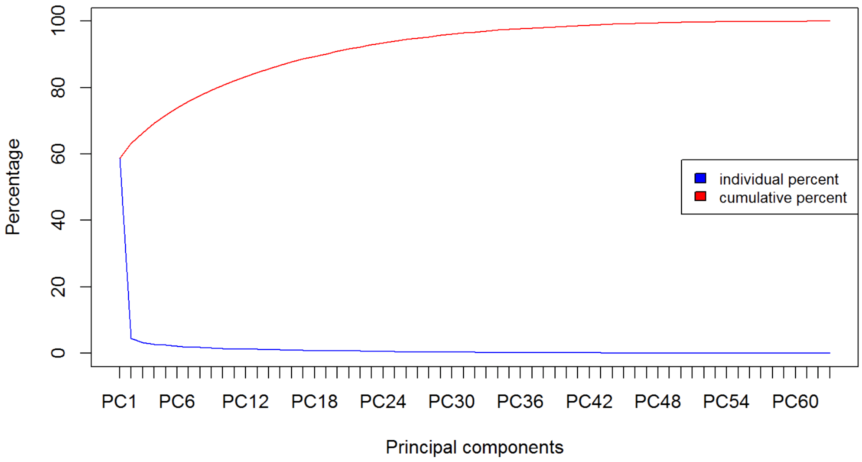


Figure 25: Scree plot.

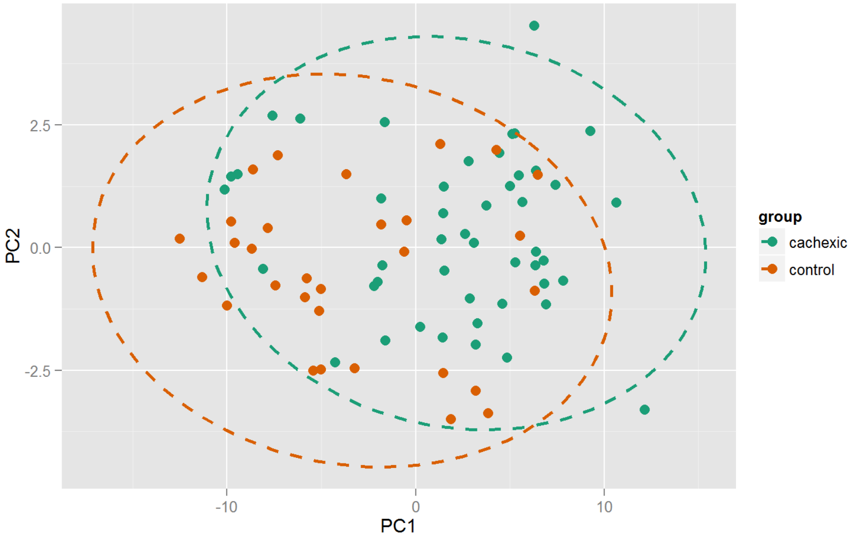


Figure 26: 2D PCA plot score (PC1 and PC2).

As can be seen on the figures, almost 60% of the variance is explained in the first principal component although there is not a clear separation in the 2D scores for the two groups.

4.2.2.4 *Machine Learning*

The results from machine learning experiments, testing 5 different models can be seen in Table 21. The resampling method was repeated cross-validation with 10 folds and 10 repetitions.

Model	Accuracy	Kappa	AccuracySD	KappaSD
pls	0.7475	0.4462	0.1352	0.3010
J48	0.6209	0.1928	0.1371	0.2972
JRip	0.6771	0.3203	0.1569	0.3237
svmLinear	0.6509	0.2784	0.1693	0.3375
rf	0.7218	0.3978	0.1464	0.3222

Table 21: Classification results.

The best model from the accuracy point of view was *pls* with 74,75%. The paper where this dataset came from (Eisner et al., 2010), uses more samples, 93 urine samples exactly, and the best model found was a SVM model using cross-validation with 10 folds that gave 82.2% of accuracy. Because of the use of more samples, the results can not be directly compared.

4.3 CASSAVA

4.3.1 *Introduction*

Cassava is a root well known and widely cultivated in tropical and subtropical regions for its starchy tuberous root, which is a great source of carbohydrates. It also has a great variety of applications, like animal feeding, culinary or alcoholic beverages. In some countries, cassava has also been tested as an ethanol biofuel feedstock.

As cassava is a tropical root, it undergoes PPD, which is characterized by streaks of blue/black in the root vascular tissue, which with time spread more and cause a more brown discoloration. PPD begins quickly within 24h postharvest and because of that, the roots need to be rapidly consumed. Some studies revealed that the deterioration is caused mostly from wound-healing responses (Uarrota et al., 2014).

This study was conducted in order to identify changes and discriminate cassava samples during post-harvest physiological deterioration and from different regions with the aid of supervised and unsupervised methods of data analysis.

4.3.2 *IR data*

Samples with different days of deterioration were collected, more specifically fresh samples (0 days), 3 days, 5 days, 8 days and 11 days of deterioration (PPD). Also the samples were collected from different varieties: SCS 253 Sangão (SAN); Branco (BRA); IAC576-70 - "Instituto Agronômico de Campinas" (IAC); and Oriental (ORI). Each combination of variety with PPD has 5 replicates, which we can aggregate. A total of 80 samples were collected and aggregating the replicates we stay with 16 samples.

Two different analysis were tested, one with the replicates and another without the replicates. For more information about the methods used below see section 2.2.

4.3.2.1 *Univariate Analysis*

An one-way ANOVA and Tukey's HSD post-hoc test was performed on the dataset with no replicates and the results are shown in Tables 22 and 23, with varieties and PPD metadata respectively.

wavelength	pvalues	logs	fdr	tukey
3388.44	0.01469	1.833	0.1183	SAN-BRA
3398.08	0.01472	1.832	0.1183	SAN-BRA
3407.73	0.01473	1.832	0.1183	SAN-BRA
3417.37	0.01486	1.828	0.1183	SAN-BRA
3378.80	0.01489	1.827	0.1183	SAN-BRA
3427.02	0.01513	1.820	0.1183	SAN-BRA
3369.15	0.01521	1.818	0.1183	SAN-BRA
3359.51	0.01566	1.805	0.1183	SAN-BRA
3436.66	0.01568	1.805	0.1183	SAN-BRA
3349.86	0.01616	1.792	0.1183	SAN-BRA

Table 22: ANOVA results using the cassava dataset with no replicates with varieties metadata.

wavelength	pvalues	logs	fdr	tukey
2356.54	0.1929	0.7147	0.8992	
2366.18	0.2031	0.6922	0.8992	
2337.25	0.2656	0.5757	0.8992	
2327.60	0.3256	0.4873	0.8992	
2346.89	0.3286	0.4833	0.8992	
2317.96	0.4691	0.3287	0.8992	
2375.82	0.5033	0.2982	0.8992	
967.81	0.5375	0.2696	0.8992	
977.45	0.5470	0.2620	0.8992	
2308.32	0.5805	0.2362	0.8992	

Table 23: ANOVA results using the cassava dataset with no replicates with PPD metadata.

From the tables above, it is easy to observe that only for the varieties metadata variable, it was possible to detect statistical differences (*p-value* < 0.05), namely considering the comparison SAN-BRA.

4.3.2.2 *Clustering*

The dendrogram from hierarchical clustering shows that the replicates stay together in the clusters but the varieties and PPDs are spread by the clusters, which is shown in Figure 27.

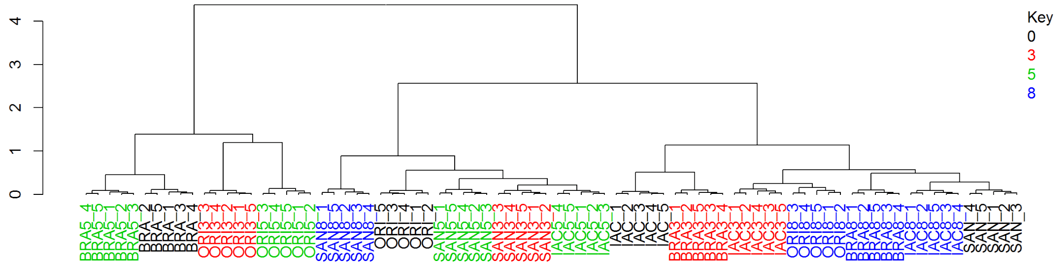


Figure 27: Resulting dendrogram from hierarchical clustering of cassava dataset with replicates.

4.3.2.3 *PCA*

The 2D PCA score plot from the PCA analysis is shown in Figure 28.

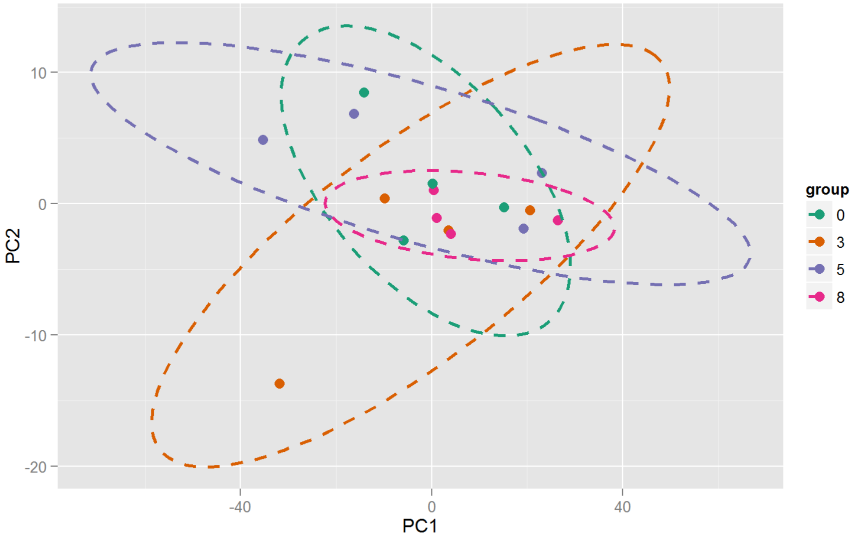


Figure 28: 2D plot score (PC1 and PC2) from cassava data without the replicates and PPDs metadata.

As can be seen in Figure 28, there is much overlap between the clusters formed by the ellipses on PPD days.

5

CONCLUSIONS AND FUTURE WORK

To face the numerous challenges that metabolomics data arises, an R package was developed during the course of this dissertation. The package includes features and methods for a variety of important aspects of metabolomics data analysis, starting with reading of the data into a defined structure for further manipulation. In this work, various methods of preprocessing and visual exploration of the data, as well methods for data analysis and data mining have been presented, implemented and demonstrated in distinct real-world case studies.

With the possibility of quickly creating and visualizing the results of one or more pipelines of analysis, the package can be used by anyone with or without an informatics background. The package is quite flexible, with functions that are easy to use having default configurations, but with the possibility of configuring the more specific details as users get acquainted with the potential of the functions.

Therefore, we believe that the package put forward during this work will be a valuable tool for researchers in a growing field of research, as it is the case with metabolomics.

For future work, there is still much room to improve the package, which includes improving the already implemented features and implementing new ones. The future work includes:

- Support to more types of metabolomics data (e.g. MS will be integrated in a very near future given its importance);
- Add more visualization methods for data exploration;
- Implement additonal preprocessing methods for existing and newly supported data types;
- Add more features to machine learning section (e.g. create interfaces with other available packages);
- Implement additional methods for feature selection (e.g. GA wrappers);
- Identification and quantification of metabolites from NMR and MS spectra;
- Implement data integration (or fusion) methods to support the integration of multiple sources of data;
- Add methods for the interpretations of the results (e.g. metabolic pathways analysis).

Also, in the near future the package will be made available through CRAN and published in a reference journal. Results from two of the case studies have already been submitted for publication and made use of this tool.

BIBLIOGRAPHY

F. J. Acevedo, J. Jiménez, S. Maldonado, E. Domínguez, and A. Narváez. Classification of wines produced in specific regions by uv-visible spectroscopy combined with support vector machines. *Journal of agricultural and food chemistry*, 55:6842–9, 2007.

C. D Adam, S. L. Sherratt, and V. L. Zholobenko. Classification and individualization of black ballpoint pen inks using principal component analysis of uv-vis absorption spectra. *Forensic science international*, 174:16–25, 2008.

B. Aernouts, E. Polshin, W. Saeys, and J. Lammertyn. Analytica chimica acta mid-infrared spectrometry of milk for dairy metabolomics : A comparison of two sampling techniques and effect of homogenization. *Analytica Chimica Acta*, 705:88–97, 2011.

C. V. Di Anibal, M. Odena, I. Ruisánchez, and M. P. Callaoa. Determining the adulteration of spices with sudan i-ii-ii-iv dyes by uv-visible spectroscopy and multivariate classification techniques. *Talanta*, 79:887–92, 2009.

C. V. Di Anibal, M. P. Callao, and I. Ruisánchez. 1h nmr and uv-visible data fusion for determining sudan dyes in culinary spices. *Talanta*, 84:829–33, 2011.

A. A. Argyri, E. Z. Panagou, P. A. Tarantilis, M. Polysiou, and G.-J. E. Nychas. Sensors and actuators b : Chemical rapid qualitative and quantitative detection of beef fillets spoilage based on fourier transform infrared spectroscopy data and artificial neural networks. *Sensors & Actuators: B. Chemical*, 145:146–154, 2010.

S. Awale, F. Li, H. Onozuka, H. Esumi, Y. Tezuka, and S. Kadota. Constituents of brazilian red propolis and their preferential cytotoxic activity against human pancreatic panc-1 cancer cell line in nutrient-deprived condition. *Bioorg. Med. Chem.*, 16:181–189, 2008.

R. M. Balabin and S. V. Smirnov. Variable selection in near-infrared spectroscopy: Benchmarking of feature selection methods on biodiesel data. *Analytica Chimica Acta*, 692:63–72, 2011.

V. S. Bankova, S. L. De Castro, and M. C. Marcucci. Propolis: recent advances in chemistry and plant origin. *Apidologie*, 31:3–15, 2000.

A. H. Banskota, Y. Tezuka, and S. H. Kadota. Recent progress in pharmacological research of propolis. *Phytother Res*, 15:561–571, 2001.

O. Barbosa-García, G. Ramos-Ortíz, J.L. Maldonadoand J.L. Pichardo-Molina, M.A. Meneses-Nava, J.E.A. Landgrave, and J. Cervantes-Martínez. Uv-vis absorption spectroscopy and multivariate analysis as a method to discriminate tequila. 66:129–34, 2007.

C. Beleites. hyperspec introduction, 2012.

C. Beleites. *hyperSpec Introduction*, May 2014. URL http://cran.r-project.org/web/packages/hyperSpec/vignettes/introduction.pdf. CENMAT and DI3, University of Trieste Spectroscopy - Imaging, IPHT Jena e.V.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

G. A. Burdock. Review of the biological properties and toxicity of bee propolis (propolis). *Food Chem Toxicol*, 36:347–363, 1998.

D. H. Burns, S. Rosendahl, D. Bandilla, O. C. Maes, H. M. Chertkow, and H. M. Schipper. Near-infrared spectroscopy of blood plasma for diagnosis of sporadic alzheimer's disease. *Journal of Alzheimer's disease : JAD*, 17:391–7, 2009.

J. Chen, Y. Long, M. Han, T. Wang, Q. Chen, and R. Wang. Water soluble derivative of propolis mitigates scopolamine-induced learning and memory impairment in mice. *Pharmacol Biochem Behav*, 90:441–446, 2008.

W. W. Cohen. Fast effective rule induction. *Twelfth International Conference on Machine Learning*, pages 115–123, 1995.

D. Cozzolino, L. Flood, J. Bellon, M. Gishen, and M. De Barros Lopes. Combining near infrared spectroscopy and multivariate analysis as a tool to differentiate different strains of saccharomyces cerevisiae : a metabolomic. pages 1089–1096, 2006.

R. Eisner, C. Stretch, T. Eastman, J. Xia, D. Hau, S. Damaraju, R. Greiner, D. S. Wishart, and V. E. Baracos. Learning to predict cancer-associated skeletal muscle wasting from 1h-nmr profiles of urinary metabolites. *Metabolomics*, 7:25–34, 2010.

W. J. Evans, J. E. Morleya, J. Argilésa, C. Balesa, V. Baracosa, D. Guttridgea, A. Jatoia, K. Kalantar-Zadeha, H. Lochsa, G. Mantovania, D. Marksa, W. E. Mitcha, M. Muscaritolia, A. Najanda, P. Ponikowskia, F. R. Fanellia, M. Schambelana, A. Scholsa, M. Schustera, D. Thomas, R. Wolfea, and S. D. Anker. Cachexia: A new definition. *Clinical Nutrition*, 27:793–799, 2008.

K. Eymanesh, M. H. Darvishi, and S. Sardari. Metabolome comparison of transgenic and non-transgenic rice by statistical analysis of ftir and nmr spectra. 16:119–123, 2009.

P. Filzmoser, H. Fritz, and K. Kalcher. *Robust PCA by Projection Pursuit*, September 2014. URL http://cran.r-project.org/web/packages/pcaPP/pcaPP.pdf.

S. Silici G. Vardar-Ünlü and M Ünlü. Composition and in vitro antimicrobial activity of populus buds and poplar-type propolis. *World J Microbiol Biotechnol*, 24:1011–1017, 2008.

G. Gekker, S. Hu, M. Spivak, J. R. Lokensgard, and P. K. Peterson. Anti-hiv-1 activity of propolis in cd4+ lymphocyte and microglial cell cultures. *J. Ethnopharmacol*, 102:158–163, 2005.

B. A. Hanson. Chemospec: An r package for chemometric analysis of spectroscopic data and chromatograms, 2013.

J. Hao, W. Astle, M. De Iorio, and T. Ebbels. Batman–an r package for the automated quantification of metabolites from nuclear magnetic resonance spectra using a bayesian model. *Bioinformatics (Oxford, England)*, 28:2088–90, 2012.

K. Haug, R. M. Salek, P. Conesa, J. Hastings, P. de Matos, M. Rijnbeek, T. Mahendrakar, M. Williams, S. Neumann, P. Rocca-Serra, E. Maguire, A. González-Beltrán, S. Sansone, J. L. Griffin, and C. Steinbeck. Metabolights– an open-access general-purpose repository for metabolomics studies and associated meta-data. *Nucleic Acids Res*, 2013.

M. Kansiz, P. Heraud, B. Wood, F. Burden, J. Beardall, and D. Mcnaughton. Fourier transform infrared microspectroscopy and chemometrics as a tool for the discrimination of cyanobacterial strains. 52, 1999.

K. Khairudin and N. Afiqah. Direct discrimination of different plant populations and study on temperature effects by fourier transform infrared spectroscopy. 2013.

B. König. Plant sources of propolis. *Bee World*, 66:136–139, 1985.

D. Kruzlicová, V. Mrázová, K. Snuderl, J. Mocák, and E. Lankmayr. Chemometric classification of edible oils. 2008.

M. Kuhn, J. Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, and RCoreTeam. *Classification and Regression Training*, August 2014. URL http://cran.r-project.org/web/packages/caret/caret.pdf. Misc functions for training and plotting classification and regression models.

S. Kuhnen, J. B. Ogliari, P. F. Dias, M. da S. Santos, A. G. F., C. C Bonham, K. V. Wood, and M. Maraschin. Metabolic fingerprint of brazilian maize landraces silk (stigma/styles) using nmr spectroscopy and chemometric methods. *Journal of agricultural and food chemistry*, 58:2194–200, 2010.

S. Kumar, P. C. Panchariya, B. Prasad, and A. L. Sharma. Discrimination of indian tea varieties using uv-vis-nir spectrophotometer and pattern recognition techniques. 1:165–174, 2013.

S. Kumazawa, R. Ueda, T. Hamasaka, S. Fukumoto, T. Fujimoto, and T. Nakayama. Antioxidant prenylated flavonoids from propolis collected in okinawa, japan. *J. Agric. Food Chem*, 55:7722–7725, 2007.

K. Liland. Multivariate methods in metabolomics – from pre-processing to dimension reduction and statistical analysis. *TrAC Trends in Analytical Chemistry*, 30:827–841, 2011.

M. Gomes De Lima, M. O. Moura, and G. G. Carbajal Arízaga. Barcoding without dna? species identification using near infrared spectroscopy. pages 1–9, 2011.

F. Liu, Y. He, and L. Wang. Determination of effective wavelengths for discrimination of fruit vinegars using near infrared spectroscopy and multivariate analysis. 5:10–17, 2008.

M. Maraschin, A. Somensi-Zeggio, S. K. Oliveira, S. Kuhnen, M. M. Tomazzoli, A. C. M. Zeri, R. Carreira, and M. Rocha. A machine learning and chemometrics assisted interpretation of spectroscopic data - a nmr-based metabolomics platform for the assessment of brazilian propolis. 2012.

M. C. Marcucci. Propolis: chemical composition, biological properties and therapeutic activity. *Apidologie*, 26:83–99, 1995.

I. Martinez, I.B. Standal, D.E. Axelson, B. Finstad, and M. Aursand. Identification of the farm origin of salmon by fatty acid and hr 13c nmr profiling. *Food Chemistry*, 116:766–773, 2009.

S. Masoum, C. Malabat, M. Jalali-Heravi, C. Guillou, S. Rezzi, and D. N. Rutledge. Application of support vector machines to 1h nmr data of fish oils: methodology for the confirmation of wild and farmed salmon and their origins. *Analytical and bioanalytical chemistry*, 387:1499–510, 2007.

F. Mozzi, M. E. Ortiz, J. Bleckwedel, L. D. Vuyst, and M. Pescuma. Metabolomics as a tool for the comprehensive understanding of fermented and functional foods with lactic acid bacteria. *Food Research International*, 2012.

J. Nielsen and M. C. Jewett. *Metabolomics: A Powerful Tool in Systems Biology*. Springer, 2007.

O. Papadopoulou, E.Z. Panagou, C.C. Tassou, and G.-J.E. Nychas. Contribution of fourier transform infrared (ftir) spectroscopy data on the quantitative determination of minced pork meat spoilage. *Food Research International*, 44:3264–3271, 2011.

A. C. Pereira, M. S. Reis, P. M. Saraiva, and J. C. Marques. Madeira wine ageing prediction based on different analytical techniques: Uv–vis, gc-ms, hplc-dad. *Chemometrics and Intelligent Laboratory Systems*, 105:43–55, 2011.

E. Polshin, B. Aernouts, W. Saeys, F. Delvaux, F. R. Delvaux, D. Saison, M. Hertog, B. M. Nicolaï, and J. Lammertyn. Beer quality screening by ft-ir spectrometry: Impact of measurement strategies, data pre-processings and variable selection algorithms. *Journal of Food Engineering*, 106:188–198, 2011.

O. Preisner, J. Almeida Lopes, R. Guiomar, J. Machado, and J. C. Menezes. Fourier transform infrared (ft-ir) spectroscopy in bacteriology: towards a reference method for bacteria discrimination. *Analytical and bioanalytical chemistry*, 387:1739–48, 2007.

J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

RCoreTeam. R: Hierarchical clustering, 2014a. URL https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html.

RCoreTeam. R: Distance matrix computation, 2014b. URL https://stat.ethz.ch/R-manual/R-devel/library/stats/html/dist.html.

S. Rochfort. *Metabolomics reviewed: A new "omics" platform technology for systems biology and implications for natural products research*. J. Nat. Prod., 2005.

S. Roussel, V. Bellon-Maurel, J. Roger, and P. Grenier. Authenticating white grape must variety with classification models based on aroma sensors, ft-ir and uv spectrometry. *Journal of Food Engineering*, 60:407–419, 2003a.

S. Roussel, V. Bellon-Maurel, J. Roger, and P. Fusion of aroma, ft-ir and uv sensor data based on the bayesian inference. application to the discrimination of white grape varieties. *Chemometrics and Intelligent Laboratory Systems*, 65:209–219, 2003b.

A. Salatino, É. W. Teixeira, G. Negri, and D. Message. Origin and chemical variation of brazilian propolis. *eCAM*, 2:33–38, 2005.

P. M. Santos, E. R. Pereira-Filho, and L. E. Rodriguez-Saona. Rapid detection and quantification of milk adulteration using infrared microspectroscopy and chemometrics analysis. *Food Chemistry*, 138:19–24, 2013.

J. M. Sforcin. Propolis and the immune system: a review. *J. Ethnopharmacol*, 113:1–14, 2007.

U. Teixeira Carvalho Polari Souto, M. J. Coelho Pontes, E. C. Silva, R. K. Harrop Galvão, M. C. Ugulino Araújo, F. A. Castriani Sanches, F. A. Silva Cunha, and M. S. Ribeiro Oliveira. Uv–vis spectrometric classification of coffees by spa–lda. *Food Chemistry*, 119:368–371, 2010.

N. Subari, J. Mohamad Saleh, A. Y. Md Shakaff, and A. Zakaria. A hybrid sensing approach for pure and adulterated honey classification. *Sensors (Basel, Switzerland)*, 12:14022–40, 2012.

K. Tan-No, K. T. Nakajima, T. Shoii, O. Nakagawasai, F. Niijima, M. Ishikawa, Y. Endo, T. Sato, S. Satoh, and K. Tadano. Anti-inflammatory effect of própolis through nitric oxide production on carrageenin-induced mouse paw edema. *Biol. Pharm. Bull*, 29:96–99, 2006.

N. C. Thanasoulias, E. T. Piliouris, M. E. Kotti, and N. P. Evmiridis. Application of multivariate chemometrics in forensic soil discrimination based on the uv-vis spectrum of the acid fraction of humus. *Forensic science international*, 130:73–82, 2002.

N. C. Thanasoulias, N. A. Parisis, and N. P. Evmiridis. Multivariate chemometrics for the forensic discrimination of blue ball-point pen inks based on their vis spectra. *Forensic Science International*, 138:75–84, 2003.

E. A. Tosi, E. Ré, M. E. Ortega, and A. F. Cazzoli. Food preservative based on propolis: bacteriostatic activity of propolis polyphenols and flavonoids upon escherichia coli. *Food Chem*, 104:1025–1029, 2007.

D. Tulpan, S. Léger, L. Belliveau, A. Culf, and M. Cuperlovic-Culf. Metabohunter: an automatic approach for identification of metabolites from 1h-nmr spectra of complex mixtures. *BMC Bioinformatics*, 2011.

V. G. Uarrota, R. Moresco, B. Coelho, E. da Costa Nunes, L. A. Martins Peruch, E. de Oliveira Neubert, M. Rocha, and M. Maraschin. Metabolomics combined with chemometric tools (pca, hca, pls-da and svm) for screening cassava (manihot esculenta crantz) roots during postharvest physiological deterioration. *Food Chemistry*, 161:67–78, 2014.

M. Urbano, M. D. Luque de Castro, P. M. Pérez, J. García-Olmo, and M. A. Gómez-Nieto. Ultraviolet–visible spectroscopy and pattern recognition methods for differentiation and classification of wines. *Food Chemistry*, 97:166–175, 2006.

K. Varmuza and P. Filzmoser. *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, 2008.

K. Varmuza and P. Filzmoser. *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, 2009.

W. N. Venables, D. M. Smith, and R Core Team. *An Introduction to R*, July 2014. Notes on R: A Programming Environment for Data Analysis and Graphics Version 3.1.1 (2014-07-10).

J. Verzani. *Getting Started with RStudio*. O'Reilly Media, Inc, 2011.

S. Villas-Boas, U. Roessner, M. A. E. Hansen, J. Smedsgaard, and J. Nielsen. *Metabolome Analysis: An Introduction*. Wiley, 2007.

D. S. Wishart, T. Jewison, A. Chi Guo, M. Wilson, C. Knox, Y. Liu, Y. Djoumbou, R. Mandal, F. Aziat, E. Dong, S. Bouatra, I. Sinelnikov, D. Arndt, J. Xia, P. Liu, F. Yallou, T. Bjorndahl, R. Perez-Pineiro, R. Eisner, F. Allen, V. Neveu, R. Greiner, and A. Scalbert. Hmdb 3.0—the human metabolome database in 2013. *Nucleic Acids Res*, 2013.

I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Pratical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, third edition, 2011.

E. Wollenweber, B. M. Hausen, and W. Greenaway. Phenolic constituents and sensitizing properties of propolis, poplar balsam and balsam of peru. *Bulletin de Groupe Polyphenol*, 15:112–120, 1990.

J. Xia, T. C. Bjorndahl, P. Tang, and D. S. Wishart. Metabominer–semi-automated identification of metabolites from 2d nmr spectra of complex biofluids. *BMC bioinformatics*, 9:507, 2008.

J. Xia, N. Psychogios, N. Young, and D. S. Wishart. Metaboanalyst: a web server for metabolomic data analysis and interpretation. *Nucleic Acids Res*, 37:652–60, 2009.

J. Xia, R. Mandal, I. V. Sinelnikov, D. Broadhurst, and D. S. Wishart. Metaboanalyst 2.0–a comprehensive server for metabolomic data analysis. *Nucleic Acids Res*, 40:127–33, 2012.

Z. Yildirim, S. Hacievliyagil, N. O. Kutlu, M. Kurkcuoglu, M. Iraz, and R. Durma. Effect of water extract of turkish propolis on tuberculosis infection in guinea-pigs. *Pharmacol Res*, 49:287–292, 2004.