

Universidade do Minho

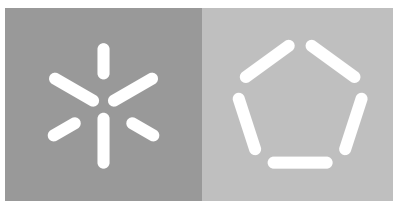
Escola de Engenharia

Departamento de Informática

Telma Adriana Pereira Afonso

Development of web-based tools for spectral data analysis and mining

September 2017



Universidade do Minho

Escola de Engenharia

Departamento de Informática

Telma Adriana Pereira Afonso

Development of web-based tools for spectral data analysis and mining

Master dissertation

Master Degree in Bioinformática

Dissertation supervised by

Miguel Francisco Almeida Pereira da Rocha

Marcelo Maraschin

September 2017

ABSTRACT

The recent advances in different analytical techniques able to produce spectral data, including Raman, [Infrared \(IR\)](#) or [Ultraviolet-Visible \(UV-vis\)](#) spectroscopies, have provided novel approaches for many research issues in the biological and chemical fields. Indeed, they have allowed to address tasks in functional genomics, sample characterization and classification, or drug discovery. To take full advantage of these data, advanced bioinformatics methods are required for data analysis and mining.

A number of methods and tools for spectral data analysis have been put forward recently, being one of the major limitations still faced the lack of integrated frameworks for extracting relevant knowledge from these data and being able to integrate these data with previous biochemical knowledge. Also, the lack of reproducibility in many data analysis or data mining processes is a strong obstacle for biological discovery, being common the lack of data and data analysis pipelines in the published work.

In recent work from the host group, *specmine*, a metabolomics and spectral data analysis/mining framework, in the form of a package for the R system, has been developed to address some of these issues. In this thesis, the main aim is to design and develop an integrated web-based platform for spectral data analysis and mining, based on the *specmine* package, providing an easier and more user friendly interface, but also addressing some of the package's current limitations. The work will also address its application in case studies related to the analysis of the characteristics and potential of natural products, addressing as well the exploration and integration of data from distinct experimental techniques, mainly focusing on [IR](#), [UV-vis](#) and Raman spectroscopies.

RESUMO

Recentes avanços nas diferentes técnicas analíticas capazes de produzir dados espectrais, incluindo as espectroscopias de Raman, Infravermelho e Ultravioleta-visível, têm contribuído com novas abordagens em vários problemas nos campos da biologia e química. De facto, tais avanços permitiram abordar tarefas em genómica funcional, caracterização e classificação de amostras, ou na descoberta de fármacos. De modo a obter o máximo de informação a partir deste tipo de dados, são necessários métodos avançados de bioinformática para a análise e extração de conhecimento dos dados.

Recentemente, vários métodos e ferramentas para análise de dados espectrais têm surgido, sendo que uma das maiores limitações enfrentadas é a falta de estruturas integradas que permitam a extração de conhecimento relevante a partir deste tipo de dados, integrando estes dados com conhecimento bioquímico prévio. A falta de reprodutibilidade em muitos processos de análise e extração de conhecimento a partir de dados é também um forte obstáculo na descoberta biológica, sendo comum a falta de *pipelines* de análise nos trabalhos atualmente publicados.

Num trabalho recente do grupo anfitrião foi desenvolvido o *specmine*, uma ferramenta para análise e extração de conhecimento de dados espectrais, sob a forma de uma biblioteca para o sistema R, de modo a abordar os problemas mencionados. Nesta tese, o principal objetivo é projetar e desenvolver uma plataforma baseada em web para análise e extração de conhecimento a partir de dados espectrais, baseada no *specmine*, fornecendo assim uma interface agradável e de fácil utilização para o utilizador, abordando algumas das atuais limitações desta ferramenta. Este trabalho irá também abordar a aplicação do *specmine* em casos de estudo relacionados com a análise de características e potencial de produtos naturais, abordando ainda a exploração e integração de dados de técnicas experimentais distintas, focando principalmente as espectroscopias de Infravermelho, Ultravioleta-visível e Raman.

CONTENTS

1	INTRODUCTION	1
1.1	Context	1
1.2	Objectives	2
1.3	Work Plan	3
2	STATE OF THE ART	5
2.1	Techniques	5
2.1.1	Infrared Spectroscopy	5
2.1.2	Ultraviolet-visible Spectroscopy	6
2.1.3	Raman Spectroscopy	7
2.2	Workflow Of A Metabolomics Experiment	10
2.2.1	Preprocessing	11
2.2.2	Univariate data analysis	14
2.2.3	Unsupervised methods	17
2.2.4	Supervised methods: machine learning	18
2.2.5	Feature selection	22
2.3	Data Fusion	23
2.4	Available Free Tools For Metabolomics and Spectral Data	25
2.5	Other General Free Tools	27
3	DEVELOPMENT	29
3.1	Development Strategy and Tools	29
3.2	Specmine, an R package for metabolomics data analysis	32
3.2.1	Data reading and dataset structure	33
3.2.2	Exploratory analysis and data pre-processing	34
3.2.3	Univariate and unsupervised multivariate analysis	37
3.2.4	Machine learning and feature selection	39
3.3	Platform Architecture	40
3.4	Authentication System	41
3.5	Private and Public Projects	42
3.6	Import Files	44
3.7	Data Visualization	44
3.8	Preprocessing	46
3.9	Data Analysis	47
3.9.1	Univariate Data Analysis	48

3.9.2	Linear Regression Analysis	50
3.9.3	Unsupervised Multivariate Analysis	50
3.9.4	Supervised Multivariate Analysis: Machine Learning	53
3.9.5	Feature Selection	53
3.10	Data Model for Project, Dataset and User Management	54
3.11	Password Encryption for Authentication System	56
4	USE CASES	59
4.1	Propolis	59
4.1.1	Context	59
4.1.2	Data Overview	60
4.1.3	Preprocessing	60
4.1.4	Univariate Analysis	61
4.1.5	Clustering	61
4.1.6	Principal Components Analysis	61
5	CASE STUDY	63
5.1	Introduction	63
5.2	Data types	65
5.2.1	Ultraviolet-visible Data	65
5.2.2	CIELAB Data	66
5.2.3	Fusion Data	66
5.3	Univariate Analysis	66

LIST OF FIGURES

Figure 1	Example of IR (A), UV-vis (B) and Raman (C) spectra with commonly used units represented in the axis.	7
Figure 2	General workflow of the various metabolomics approaches.	11
Figure 3	Graphical representation of first and second order derivatives calculation.	13
Figure 4	Example of a dendrogram resulting from a cluster analysis performed over 32 samples. The distance between samples is represented in the y axis, whereas sample names are represented in the x axis.	18
Figure 5	Graphical representation of tabular data used in a machine learning approach, including feature matrix X and a property vector y .	19
Figure 6	General workflow of a machine learning approach.	20
Figure 7	Workflow of Filter, Wrapper and Embedded approaches in feature selection.	22
Figure 8	Graphical representation of A. Low-Level Fusion (LLF), B. Intermediate-Level Fusion (ILF) and C. High-Level Fusion (HLF).	24
Figure 9	Representation of reactive programming objects in a Shiny application.	30
Figure 10	Modules in the <i>specmine</i> package. Adapted from Costa et al. (2016) .	33
Figure 11	Representation of the dataset structure in <i>specmine</i> . Adapted from Costa et al. (2016) .	33
Figure 12	Graphical representation of the application's structure.	40
Figure 13	Graphical representation of the application's file structure. The file-names in bold represent the files to which I greatly or totally contributed to, given the scope of this work.	41
Figure 14	Main page of the web application.	42
Figure 15	Authentication menu after successful login (detail).	42
Figure 16	Graphical representation of a project's structure.	43
Figure 17	Zoomed view over <i>My Projects</i> page.	43
Figure 18	Zoomed view over <i>New Project</i> window.	45
Figure 19	Zoomed view over <i>Spectra Plot</i> tab in the <i>Data Visualization</i> page.	46
Figure 20	Zoomed view over <i>Pre-Processing</i> page.	47
Figure 21	Zoomed view over the <i>Run Analysis</i> page.	48

Figure 22	Zoomed view over <i>Analysis Results</i> page for one-way Analysis of Variance (ANOVA), showing the numerical results tab and emphasizing the options used for the analysis.	49
Figure 23	Zoomed view over the <i>Analysis Results</i> page for Principal Component Analysis (PCA), showing the <i>Make Plots</i> tab for the scree plot, emphasizing the customizable options (A) and example of a pairs plot made in the web application (B).	52
Figure 24	Zoomed view over <i>Analysis Results</i> page for hierarchical clustering, showing the clustering dendrogram.	53
Figure 25	Zoomed view over the <i>Analysis Results</i> page for correlation analysis, emphasizing the correlation heatmap and respective colour scale.	54
Figure 26	MySQL model used in the platform for project, dataset and user management.	55
Figure 27	Graphical representation of the hashing process.	57
Figure 28	The dataset summary (A) and partial metadata table (B).	60
Figure 29	Propolis raw spectra (A) and spectra processed with smoothing interpolation, background, offset and baseline corrections (B), with samples colored according to season.	60
Figure 30	Hierarchical clustering analysis dendrogram of the propolis dataset. Euclidean distance and a complete agglomeration method were used.	61
Figure 31	Principal components importance (A) and variable loadings (B) tables from the PCA performed over the propolis dataset.	62
Figure 32	Scores scatter plot of the PCA performed over the propolis dataset for the first and second components.	62
Figure 33	Representation of the CIE L* a* b* color space.	64
Figure 34	ANOVA results using the <i>colors</i> metadata variable (A) and respective plot of the $-\log_{10}$ of p-values with a p-value threshold value of 0.05 (B).	67

LIST OF TABLES

Table 1	Applications of IR, UV-vis and Raman spectroscopies.	8
Table 2	Available free tools for metabolomics and spectral data.	26
Table 3	<i>Specmine</i> package functions regarding data reading and dataset structure	33
Table 4	<i>Specmine</i> package functions for data exploratory analysis and pre-processing.	35
Table 5	<i>Specmine</i> package functions for univariate and unsupervised multivariate analysis.	38
Table 6	<i>Specmine</i> package functions for machine learning and feature selection.	39

LIST OF ACRONYMS

ALS	Alternate Least Squares. 9
ANN	Artificial Neural Network. 9, 19, 21
ANOVA	Analysis of Variance. 8, 14, 15, 26, 32, 40
AUC	Area Under the ROC Curve. 34
DA	Discriminant Analysis. 9
Di-PLS	Discriminant Partial Least Squares. 8
FC	Fold Change. 15, 32, 33
FDR	False Discovery Rate. 15, 32
FIR	Far-Infrared. 5
FTIR	Fourier Transform Infrared Spectroscopy. 2, 5, 8, 23, 25
GA	Genetic Algorithms. 23
GC-MS	Gas Chromatography-Mass Spectrometry. 10, 26, 28
HCA	Hierarchical Clustering Analysis. 17, 18, 26
HLF	High-Level Fusion. vii, 24, 25
HSD	Honest Significance Difference. 15
IDE	Integrated Development Environment. 38
ILF	Intermediate-Level Fusion. vii, 24, 25
IR	Infrared. i, vii, ix, 2, 5–9, 11, 13, 14, 23, 25, 26, 28, 37
KNN	K-Nearest Neighbors. 9, 19, 21, 27
LASSO	Least Absolute Shrinkage and Selection Operator. 23
LC-MS	Liquid Chromatography-Mass Spectrometry. 10, 26, 28
LDA	Linear Discriminant Analysis. 8, 9, 19, 20, 25, 27

LLF	Low-Level Fusion. vii , 24 , 25
MIR	Mid-Infrared. 5 , 8
MLR	Multiple Linear Regression. 9
MS	Mass Spectroscopy. 26
MSC	Multiplicative Scatter Correction. 8 , 13 , 40
NIR	Near-Infrared. 5 , 7–9 , 25
NMR	Nuclear Magnetic Resonance. 10 , 26–28
OSC	Orthogonal Signal Correction. 9
PCA	Principal Component Analysis. 8 , 9 , 17 , 20 , 25 , 26 , 33 , 40 , 41
PLCA	Probabilistic Latent Component Analysis. 8
PLS	Partial Least Squares. 9 , 19 , 25
PLS-DA	Partial Least Squares Discriminant Analysis. 8 , 9 , 19 , 26
PLS-r	Partial Least Squares Regression. 8 , 9 , 19 , 27
PNN	Probabilistic Neural Network. 9
PPD	Postharvest Physiological Deterioration. 8
RDBMS	Relational Database Management System. 38 , 39 , 42
RFE	Recursive Feature Elimination. 23 , 35
RMSE	Root Mean Square Error. 34
SFA	Significant Factor Analysis. 9
SFS	Sequential Feature Selection. 23
SIMCA	Soft Independent Modelling by Class Analogy. 8 , 9 , 19 , 20
SNV	Standard Normal Variate. 8 , 9 , 12 , 13
SPA-LDA	Successive Projections Algorithm-Linear Discrimi- nant Analysis. 9
SQL	Structured Query Language. 39
SVM	Support Vector Machines. 8 , 9 , 19 , 21 , 26 , 27
TSF	Time Series Forecasting. 27

UV-vis Ultraviolet-Visible. i, vii, ix, 2, 6–9, 11, 13, 23, 25, 26,
28, 37

INTRODUCTION

1.1 CONTEXT

Metabolites represent a diverse group of small molecules including lipids, amino acids, peptides, nucleic acids, organic acids, vitamins, thiols and carbohydrates (Zhang et al., 2012). They are the end products of cell metabolic processes, and both genetic and environmental changes can trigger a response in biological systems which ultimately result in variations of metabolite levels. The set of metabolites produced by a system constitute its metabolome (Fiehn, 2002).

Metabolomics is a relatively new field, incorporating the so called omics technologies, alongside the mature proteomics and genomics which study proteins and genes, respectively, and can be defined as the study of chemical processes involving small molecules in biological systems, including their identification and quantification (Daviss, 2005).

The close relation of metabolites to an organism's phenotype (Fiehn, 2002) allows for metabolomics to be used in a large range of applications. Amongst these are the phenotyping of genetically modified plants, determination of gene function and monitoring responses to biotic and abiotic stress. Therefore, metabolomics provides a more comprehensive view of how cells function, as well as the identification of changes that can occur in specific metabolites (Roessner and Bowne, 2009).

This rapidly emerging field combines different strategies to achieve the experiments goals, including the identification and quantification of metabolites, through sophisticated analytical technologies combined with statistical and multivariate methods for data extraction and interpretation (Roessner and Bowne, 2009). Such experiments yield large amounts of data and if we consider that information output has been growing exponentially over the years, the need for an automated analysis process becomes a necessity (Larsen and Von Ins, 2010).

To cover such necessity, a number of different computational tools regarding spectral data analysis have become available, targeting broader purposes or more specific tasks,

as well as covering a wider or smaller range of experimental techniques. Many of such tools have been developed over the open-source R scientific computing platform <https://www.r-project.org/>, thus taking advantage of numerous previous efforts and also making all scripts available for the community. *ChemoSpec* and *hyperSpec* are two such examples, covering a wide range of platforms and data formats, including spectral data from Raman, [Fourier Transform Infrared Spectroscopy \(FTIR\)](#) and [UV-vis](#) spectroscopies, techniques that will be emphasized throughout this dissertation.

Recently, in the host group, an R package named *specmine* ([Costa et al., 2016](#)) was developed. It is a metabolomics and spectral data analysis/mining framework, addressing the development of customizable data analysis pipelines, covering different types of metabolomics and spectral data, such as [IR](#), Raman and [UV-vis](#) spectroscopies. This tool will be the supporting framework for the development of this work that will seek to improve it both in terms of interface and of available functionality.

1.2 OBJECTIVES

Given the context described above, the main aim of this work will be the design and development of web-based computational platform for spectral data analysis and knowledge extraction. The work will address the exploration and integration of data from distinct experimental techniques, focusing on [UV-vis](#), Raman and [IR](#) spectroscopies.

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

- To review the state-of-the-art in metabolomics data analysis, focusing on spectral data, including a review of the main tools implementing these methods;
- To design and implement adequate web-based interfaces for spectral data analysis and data mining pipelines, based on the functions provided by the *specmine* package;
- To design and implement novel functions for spectral data analysis/mining extending the functionality of *specmine*, covering topics as data fusion from different platforms, machine learning and feature selection;
- To validate the tools developed with several case studies of interest for the host groups in the analysis of the potential of natural products, including for instance cassava and bees' propolis;
- To write scientific publications with the results of the work.

1.3 WORK PLAN

The proposed work plan for the duration of this Masters dissertation is presented below. It includes the tasks that need to be performed with respective schedule.

- **Weeks 1-12:** Studying the state of the art; analyzing available algorithms and tools, including *specmine*; writing pre-thesis;
- **Weeks 10-18:** Analyzing requirements and designing the web-based interface;
- **Weeks 16-28:** Implementing the web-based interface for *specmine* tools;
- **Weeks 22-36:** Implementing/ improving *specmine* tools for data fusion, machine learning and feature selection;
- **Weeks 16-36:** Applying tools in selected case studies; analyzing results;
- **Weeks 13-40:** Writing the thesis and publications;

As of the writing of this document, the first point has been covered (weeks 1-12), by covering the state of the art in the metabolomics field, focusing on spectral data. The second point (weeks 10-18) was also covered, by analyzing requirements and designing the web-based interface. Currently, the third point (weeks 16-28) is being covered, with the implementation of *specmine*'s tools in the platform.

STATE OF THE ART

The present chapter will cover the state of the art of the metabolomics field, focusing on spectral data. This includes the description of the main techniques and their characteristics, discussing some of the computational tools currently available for metabolomics and spectral data analysis. The workflow of data analysis for a metabolomics experiment will be fully covered, including the preprocessing, univariate and multivariate data analysis, feature selection and data fusion steps.

2.1 TECHNIQUES

2.1.1 *Infrared Spectroscopy*

Infrared (IR) spectroscopy deals with the infrared region of the electromagnetic spectrum and is commonly used in the study and identification of chemicals. The **IR** spectrum can be divided in three main regions, namely, the **Far-Infrared (FIR)** (400 cm^{-1} - 100 cm^{-1}), the **Mid-Infrared (MIR)** (4000 cm^{-1} - 400 cm^{-1}) and **Near-Infrared (NIR)** (13000 cm^{-1} - 4000 cm^{-1}). The theory is that molecules absorb specific frequencies that are characteristic of their structure, and these frequencies match the transition energy of the bond or group that vibrates. One of the great advantages of this technique lies in the fact that virtually any sample in any state may be studied.

The introduction of Fourier-transform spectrometers have significantly contributed to the advance in **IR** spectroscopy, dramatically improving the quality of infrared spectra and minimizing the time required to obtain the data. This type of instrument employs an interferometer and exploits the well established mathematical process of Fourier-transformation. **FTIR** spectroscopy has distinct advantages over other conventional methods of biochemical analysis in that it is rapid, reliable and requires a relatively small sample size and simple sample preparation procedure (Kansiz et al., 1999).

The output from such instruments is referred to as a spectrum, and it basically consists in a graph of IR light transmittance on the vertical axis vs. wavenumber on the horizontal axis (Figure 1A), usually in cm^{-1} , decreasing from left to right (Stuart, 2004).

Many are the applications of IR spectroscopy, ranging from the food area to the biological and medical fields, and it has proven to show good results (Table 1). Regarding the food industry, this technique has been used in beer quality assessment (Polshin et al., 2011), in the discrimination of *Alicyclobacillus* strains in apple juice (Lin et al., 2005), in honey sample classification according to their adulteration levels (Subari et al., 2012) and in the detection and quantification of milk adulteration (Santos et al., 2013).

Regarding the biological and medical fields, IR spectroscopy has been widely applied in chemometrics approaches, including the differentiation of different *Saccharomyces cerevisiae* strains (Cozzolino et al., 2006), the discrimination of different plant populations (Khairudin et al., 2014; Uarrotta et al., 2014), the discrimination of clinically relevant bacteria (Preisner et al., 2007) and even fly species identification (De Lima et al., 2011).

2.1.2 Ultraviolet-visible Spectroscopy

While interaction with IR light causes molecules to undergo vibrational transitions, the higher energy radiation in the UV (200 - 400 nm) and visible (400 - 700 nm) range of the electromagnetic spectrum causes many organic molecules to undergo electronic transitions. The concept is that in molecules containing π -electrons or non-bonding electrons, photons of Ultraviolet-Visible (UV-vis) light have enough energy to cause transitions between the different electronic energy levels. The wavelength of light absorbed has the energy required to move an electron from a lower energy level to a higher energy level. Because many transitions with different energies can occur, the bands that appear in the spectrum are broadened.

The UV-vis spectrum typically has absorbance values on the vertical axis and wavelength values on the horizontal axis in nm (Figure 1B) rather than cm^{-1} (Soderberg, 2010). Compared with techniques such as IR, which produces many narrow bands, UV-vis spectroscopy provides a limited amount of qualitative information. Although the spectra produced by this technique do not enable absolute identification of an unknown compound, they are frequently used to confirm the identity of a substance through comparison of the measured spectrum with a reference spectrum (Owen, 1996).

UV-vis spectroscopy has been widely used in areas such as the food industry and forensics (Table 1). In the food industry, it was applied in problems regarding for instance the classification of ground roast coffee according to type and conservation state (Souto et al.,

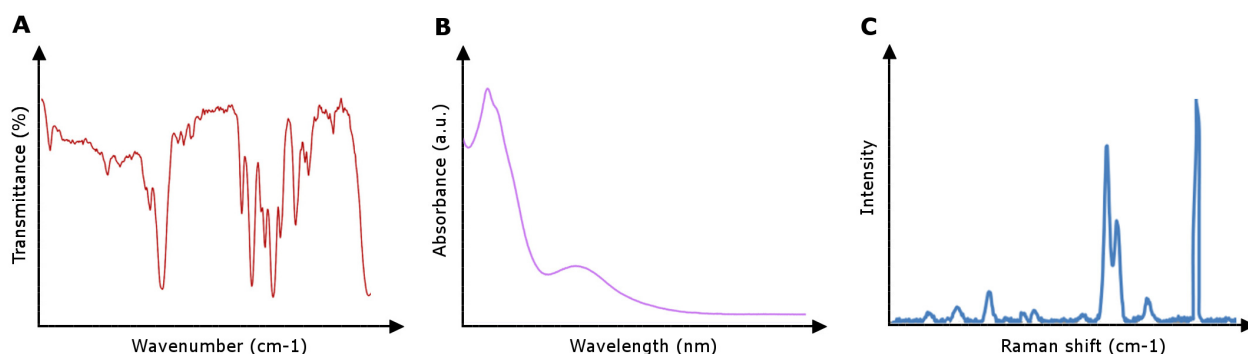


Figure 1: Example of IR (A), UV-vis (B) and Raman (C) spectra with commonly used units represented in the axis.

2010), discrimination of tea varieties (Kumar et al., 2013), prediction of wine aging (Pereira et al., 2011), wine differentiation and classification (Urbano et al., 2006) and also in tequila discrimination (Barbosa-García et al., 2007).

Regarding forensics, UV-vis spectroscopy has been used in the discrimination of blue ball-point pen inks (Thanasoulas et al., 2003) and also in soil discrimination (Thanasoulas et al., 2002).

2.1.3 Raman Spectroscopy

Spectroscopies such as Raman are employed to detect vibrational, rotational, and other low-frequency modes in a system. It is widely used to provide information on chemical structures and physical forms, in fingerprinting experiments and even to determine quantitatively or semi-quantitatively a compound in a sample. When light interacts with matter, the photons which make up the light can either be transmitted, reflected, absorbed or scattered, and it is this last tiny portion of light that Raman spectroscopy utilizes.

This technique uses a single frequency of radiation to irradiate the sample, and it is the radiation scattered from the molecule, one vibrational unit of energy different from the incident beam, which is detected. Most of the scattered light does not change its wavelength in the process (Rayleigh scattering) but part of it does, and such scattering is known as Raman scattering. The theory is that Raman scattering of monochromatic light (usually from a laser in the NIR or UV-vis range) is caused by excitations in the system, which result in the energy of the laser photons being shifted up or down. The intensity of the scattered light is plotted against its frequency (cm^{-1}) and the result is a Raman spectrum of the sample (Figure 1C).

Compared to IR spectroscopy, this technique is less widely used, largely due to problems with sample degradation and fluorescence. However, recent advances in instrument technology have simplified the equipment and reduced the problems substantially. These advances, together with the ability of Raman spectroscopy to examine samples in a wide range of states and minimal spectrum manipulation need, have led to a rapid growth in the application of the technique (Table 1) (Smith and Dent, 2005).

Raman spectroscopy has been widely applied in the forensics area, having proven to be a powerful tool in the identification of body fluids (Virkler and Lednev, 2010; Sikirzhytski et al., 2010, 2012). The pharmaceutical industry also makes use of this technique, addressing problems such as the detection of counterfeit products (Roggo et al., 2010; Sacré et al., 2011) and qualitative and quantitative detection of a mycotoxin in ground maize samples (Lee et al., 2013).

Table 1: Applications of IR, UV-vis and Raman spectroscopies.

Reference	Description	Techniques	Preprocessing	Analysis
Polshin et al. (2011)	Prediction of important beer quality parameters	FTIR	Multiplicative Scatter Correction (MSC), Baseline correction, Standard Normal Variate (SNV), 1st and 2nd Savitzky-Golay derivatives, Mean centering	PCA, Partial Least Squares Regression (PLS-r)
Lin et al. (2005)	Discrimination of <i>Alicyclobacillus</i> strains in apple juice	FTIR	Spectra smoothing, 2nd derivative, Normalization	PCA, Soft Independent Modelling by Class Analogy (SIMCA)
Subari et al. (2012)	Classification of honey according to the adulteration level	FTIR	Baseline correction, Normalization, Peak correction, Outlier removal	PCA, Linear Discriminant Analysis (LDA),
Santos et al. (2013)	Detection and quantification of milk adulteration	MIR	Normalization, 2nd derivative (Savitzky-Golay), Mean centering	SIMCA, PLS-r
Cozzolino et al. (2006)	Differentiation of different <i>Saccharomyces cerevisiae</i> strains	NIR	Autoscaling, Centering, 2nd derivative	PCA, LDA
Khairudin et al. (2014)	Discrimination of <i>Polygonum minus</i> populations	FTIR	Pareto scaling	PCA, Partial Least Squares Discriminant Analysis (PLS-DA)
Uarota et al. (2014)	Identification of changes and discrimination of cassava samples undergoing Postharvest Physiological Deterioration (PPD)	FTIR	Normalization, Baseline correction	PCA, PLS-DA, Hierarchical clustering, Support Vector Machines (SVM), One-way ANOVA
Preisner et al. (2007)	Discrimination between different types of the <i>Enterococcus faecium</i> bacterial strain	FTIR	1st and 2nd Savitzky-Golay derivatives, MSC, SNV, Mean centering, Outlier removal	Discriminant Partial Least Squares (Di-PLS), Probabilistic Latent Component Analysis (PLCA)

Table 1: Applications of IR, UV-vis and Raman spectroscopies. (Continued)

Reference	Description	Techniques	Preprocessing	Analysis
De Lima et al. (2011)	Identification of fly species in the genus <i>Neodexiopsis Malloch</i>	NIR	Savitzky-Golay derivative and smoothing	PCA, Partial Least Squares (PLS)
Kumar et al. (2013)	Discrimination of tea varieties	NIR, UV-vis	Normalization	PCA, K-means clustering, Probabilistic Neural Network (PNN), Artificial Neural Network (ANN)
Souto et al. (2010)	Classification of coffee extracts according to type and conservation state	UV-vis	None	PCA, SIMCA, Successive Projections Algorithm-Linear Discriminant Analysis (SPA-LDA)
Pereira et al. (2011)	Prediction of wine aging	UV-vis	Mean centering, Smoothing, 1st and 2nd derivatives, SNV, Orthogonal Signal Correction (OSC)	PLS-r
Urbano et al. (2006)	Differentiation and classification of wines	UV-vis	1st derivative	PCA, SIMCA
Barbosa-García et al. (2007)	Discrimination between classes of tequila	UV-vis	1st derivative, Centering	PCA, PLS-DA
Thanasoulas et al. (2003)	Forensic discrimination of blue ball-point pen inks	UV-vis	Normalization	K-means cluster analysis, PCA, Discriminant Analysis (DA)
Thanasoulas et al. (2002)	Forensic soil discrimination	UV-vis	Normalization	K-means cluster analysis, PCA, DA
Virkler and Lednev (2010)	Forensic body fluid identification (blood)	NIR, Raman	Normalization	Significant Factor Analysis (SFA), PCA, Alternate Least Squares (ALS)
Sikirzhytski et al. (2010)	Forensic identification of blood, semen and saliva stains	Raman	None	DA, SIMCA, LDA, PLS-DA
Sikirzhytski et al. (2012)	Identification of body fluid traces (semen and blood)	NIR, Raman	Cosmic ray interference removal, Normalization, Baseline correction	PCA, SFA, SVM
Roggo et al. (2010)	Identification of pharmaceutical tablets	Raman	Cosmic ray interference removal, SNV normalization, Scaling, Mean centering, Savitzky-Golay 1st derivative	SVM
Sacré et al. (2011)	Detection of counterfeit Viagra®	Raman	Normalization	PCA, SIMCA, K-Nearest Neighbors (KNN), LDA
Lee et al. (2013)	Qualitative and quantitative detection of a mycotoxin in ground maize samples	Raman	Background correction, Baseline correction, Normalization, Savitzky-Golay smoothing, Peak deconvolution	KNN, LDA, PLS-DA, Multiple Linear Regression (MLR), PLS-r, Cluster analysis

2.2 WORKFLOW OF A METABOLOMICS EXPERIMENT

Regarding metabolomics, there are four conceptual approaches: target analysis, metabolite profiling, metabolomics, and metabolic fingerprinting (Roessner and Bowne, 2009). The general workflow of the various metabolomics approaches is shown in Figure 2. Target analysis includes the determination and quantification of a small set of known metabolites, also called targets, making use of one particular analytical technique that shows the best performance for the compounds of interest. It has been applied for many years, including, for instance, in the analysis of head and neck cancer cells (Hu et al., 2015).

Metabolite profiling is different than the previous approach in the sense that it aims at the analysis of a larger set of compounds. Regarding their chemical structure, these compounds are both identified and unknown and their quantification either quantitative or semi-quantitative (an absolute quantification is not required). This approach is widely applied and has been used for instance in the identification of kidney cancer using urine's metabolic signatures (Kind et al., 2007).

The metabolomics approach employs complementary analytical methodologies, including Liquid Chromatography-Mass Spectrometry (LC-MS), Gas Chromatography-Mass Spectrometry (GC-MS) and/or Nuclear Magnetic Resonance (NMR) techniques, to determine and quantify as many metabolites as possible. Similarly to the previous approach, the compounds can be either identified or unknown. It is widely used, having been applied in the determination of natural stress influence on the metabolic status of sea snail (Rosenblum et al., 2005).

Lastly, in a metabolic fingerprinting approach, a metabolic "signature" or mass profile of the sample of interest is generated and then compared in a large sample population to screen for differences between the samples, it is most typically used for sample classification based on its spectrum. If the metabolites to analyze are either external and/or secreted by the cells then it is called footprinting approach. If signals that can significantly discriminate between samples are detected, then the metabolites can be identified and the biological relevance of that compound elucidated, saving valuable analysis time. It is commonly used in forensics, among other fields, having been applied for instance in the discrimination of blue ball-point pen inks (Thanasoulas et al., 2003). This approach will be emphasized throughout this dissertation.

The main steps in a metabolic fingerprinting experiment are: sample preparation, data acquisition, preprocessing, data analysis and data interpretation. Upon sample preparation and data acquisition the data is then preprocessed to allow an improved analysis, where the objective is to extract useful knowledge from the data. The preprocessing and analysis steps will be the main focus throughout this dissertation.

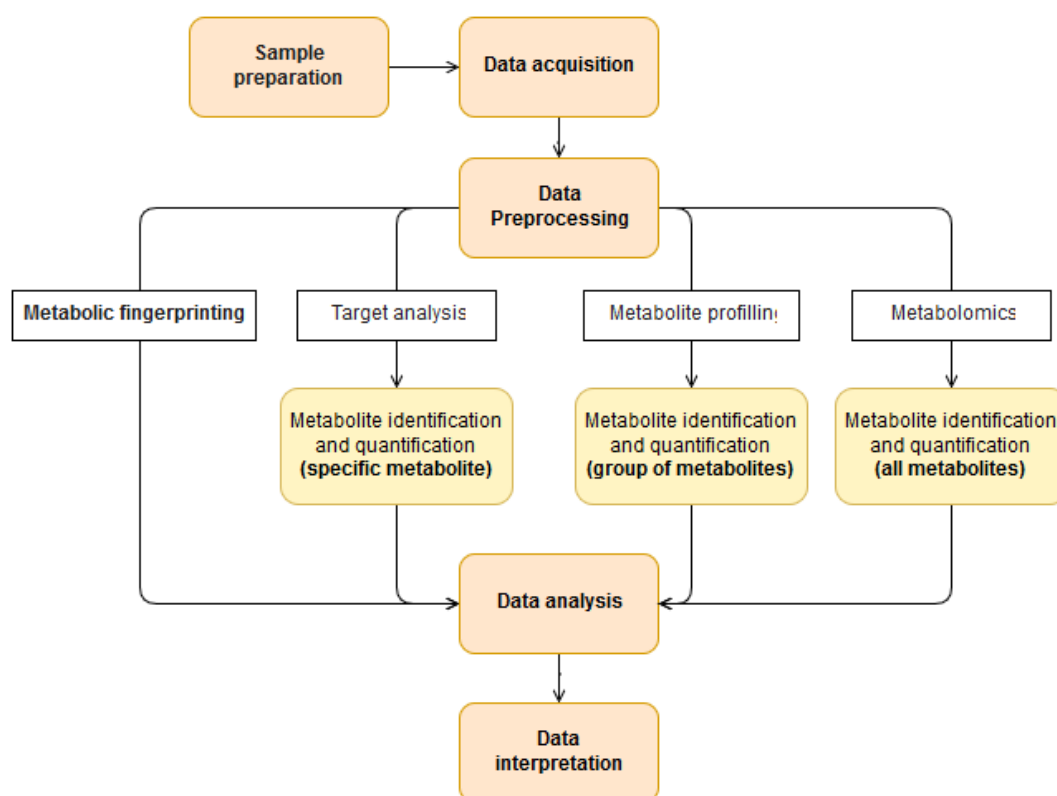


Figure 2: General workflow of the various metabolomics approaches.

2.2.1 Preprocessing

The preprocessing step covers all editing of the data up to the point of starting the analysis. This is a crucial step in any metabolomics experiment, making samples analyzable and comparable. In a metabolic fingerprinting approach, the most commonly used preprocessing methods include missing values and outlier removal, normalization or scaling, derivative calculation, mean centering and some peak spectra processing. The order in which the preprocessing steps are applied to the data is not always obvious, being sometimes governed more by practical considerations than optimal statistical analysis. The methods discussed in this section are the ones most commonly used regarding IR, UV-vis and Raman spectroscopies, the techniques focused on this dissertation.

When handling missing values, there are two main approaches: their removal or replacement. In the first approach, the value can be removed by either removing the feature or the sample containing it. In the latter, the value can be replaced using various methods, including its replacement by the row or column mean, or even using more sophisticated methods (e.g. nearest neighbors, linear approximation). Frequently, there are values within the dataset that are distant from all other observations, thus considered outliers. These

are typically excluded from the dataset, as they could interfere in the subsequent analysis results.

Peak spectra preprocessing aims to perform corrections over the spectra. It includes for instance baseline and background correction and smoothing methods, among others. Baseline correction method is used to correct unwanted linear or non-linear additions to the spectra. These additions are often associated with equipment used when measuring samples (e.g. non-linearities in detectors) or, for instance, by the interference of a complex matrix. Depending on the situation, this method might be essential, considering most statistical analysis techniques cannot distinguish between baselines and signal.

Background correction, as the name suggests, is applied to remove the background in the spectra. This background can be caused by various factors, including absorption associated with the sample holder and/or solvent used.

The smoothing methods are used to filter spectra noise, and might be specially helpful when signal-to-noise ratio is high or the subsequent analysis methods are very sensitive to noise. It helps in both visual interpretation and robustness of the analysis, but it is important to balance noise reduction and peak retention, specially in the small peaks (Liland, 2011). The Savitzky-Golay filter is one of the most popular smoothing methods. This method fits successive sub-sets of adjacent data points with a low-degree polynomial by the method of linear least squares, using a process known as convolution (Savitzky and Golay, 1964).

Other commonly used peak spectra preprocessing methods include peak alignment and binning. In its simplest form, peak alignment consists in dividing the spectra in a number of local windows, where peaks are shifted to match across spectra. Since everything is done locally, peak alignment is a fast method, however, it may lead to misalignment when peaks fall into the wrong local window or are split into two windows. When continuous spectra are recorded producing tens or hundreds of thousands of measurements per spectrum, the binning method can be helpful. In this method, the spectrum is divided into a desired number of bins and all measurements inside each bin summed, forming new spectra with fewer variables. The simplest reason for binning is that the number of variables can be too high for handling of the problem in ordinary computer memory. There are, however, a few dangers regarding the bad placing of the bins, by removing information or producing false information (Liland, 2011).

In many analytical methods, the variables measured for a given sample are subject to overall scaling or gain effects. Standardization methods attempt to correct for these kinds of effects by identifying some aspect of each sample which should be essentially constant between samples, giving all them an equal impact on the model. In the **Standard Normal Variate (SNV)** method, a weighted normalization is performed (not all points contribute to

the normalization equally). Therefore, the values are subtracted of their mean and then the result is divided by the standard deviation. Spectra treated in this manner have always zero mean value and a variance equal to one and are thus independent of original absorbance values.

The **Multiplicative Scatter Correction (MSC)** method is a relatively simple processing step that attempts to account for additive and/or multiplicative effects in spectral data. It does so by estimating light scattering or change in path length for each sample relatively to that of an ideal sample. Another method consists in centering the data, by calculating the average spectrum of the dataset and subtracting that average from each spectrum. In this method, the values are changed, but not the scale.

Derivative spectroscopy uses first or higher derivatives of absorbance with respect to wavelength for qualitative analysis and for quantification. Generally speaking, by differentiation of a zero order spectrum and obtaining consecutive derivative spectra the separation of overlapping peaks is achieved, increasing selectivity without separation of the analytes. First and second-order derivatives are the ones most commonly calculated. A graphical representation of first and second order derivatives calculation is shown in **Figure 3**. The first-order derivative consists in the rate of change of absorbance with respect to wavelength, while the second-order derivative has a very characteristic feature consisting in a negative band with minimum at the same wavelength as the maximum on the original spectrum. This method can be useful because spectra that are very similar in absorbance mode may reveal significant differences in the derivative mode. Another advantage resides in the fact that because the first derivative of a constant absorbance offset is zero, calculating the first derivative spectra always eliminates baseline shifts ([Kus et al., 1996](#)).

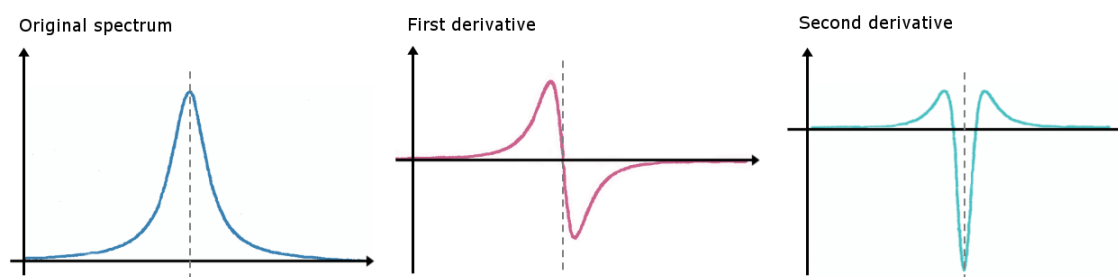


Figure 3: Graphical representation of first and second order derivatives calculation.

The preprocessing methods applied can vary greatly between datasets and type of spectroscopy used. **Table 1** summarizes the various preprocessing methods used in several experiments from the literature (in the fourth column). In IR spectroscopy, peak spectra processing methods are often applied, including smoothing and baseline correction. Normalization and scaling methods, including mean centering and **SNV** are also often used, as well as first and second-order derivatives calculation. Unlike IR spectroscopy, in UV-vis ex-

periments less preprocessing methods are applied, being normalization and first derivative calculation the most commonly applied methods. In Raman spectroscopy, the preprocessing methods usually applied are similar to those applied in IR spectroscopy.

2.2.2 Univariate data analysis

After the preprocessing step, the data is finally ready to be analyzed. This dataset is usually under the form of a matrix, with either a compound list or a peak list and their values for different samples. The main types of data analysis are: univariate analysis, unsupervised multivariate techniques and supervised multivariate techniques (machine learning).

Univariate analysis investigates each variable separately or relates a single independent variable x to a single dependent variable y . However, the data obtained from experiments regarding compounds, reactions and/or samples are multivariate in nature, which means a good characterization often requires using many variables simultaneously. Multivariate data analysis considers many variables together and thereby often gains a new and higher quality in data evaluation. The differences between supervised and unsupervised methods relies in the fact that the first ones do not need any metadata (e.g. information about natural groups within the data), while the latter requires samples to be divided into at least two classes (or groups) to allow the methods to conduct a learning (or training) process (Varmuza and Filzmoser, 2009).

This section will cover univariate analysis techniques, whereas unsupervised multivariate techniques and supervised multivariate techniques will be explored with further detail in subsection 2.2.3 and subsection 2.2.4.

Among the most popular univariate analysis techniques used in metabolic fingerprinting approaches are the t-tests, Analysis of Variance (ANOVA) and fold change analysis. Non-parametric tests often used include the Kruskal-Wallis, Kolmogorov-Smirnov and Wilcoxon signed-rank tests. Regression analysis is also often employed.

A t-test is a statistical hypothesis test in which the test statistic follows a Student's t distribution under the null hypothesis. It allows for data comparison, by determining if two sets of data are significantly different from each other. This test is most commonly used to test whether the mean of a population can have a specified value, to test if the means of two populations can be equal (two-sample test), to test whether the slope of a regression line differs significantly from 0, among other uses.

The ANOVA is a collection of statistical models used to assess the relative size of variance among group means compared to the average variance within groups. For a comparison of more than two group means, the one-way ANOVA is the appropriate method instead of

the t-test. It is similar to multiple two-sample t-tests, but since it is less conservative (results in a smaller number of type I errors) it is suited to a wide range of practical problems. The two-way ANOVA is an extension of the one-way ANOVA and it examines the influence of two different categorical independent variables on one continuous dependent variable. Besides assessing the main effect of each variable, it also assesses if there is any interaction between them.

The degree of how relatively greater the difference is between group means compared to within group variance is known to follow the F distribution. Therefore, the ANOVA makes use of the F-test to test the statistical significance by comparing the F statistic, which compares the variance between groups with the variance within groups. If any significant difference is detected by the F-test, the specific pair of group means that show differences and the pairs that do not can be examined using a post-hoc test. One such test for this task is the Tuckey's Honest Significance Difference (HSD) test (Kim, 2014).

When conducting multiple comparisons, as is the case of a metabolic fingerprinting experiment, the False Discovery Rate (FDR) is one way of conceptualizing the rate of type I errors (false positives) in null hypothesis testing. This method provides a less stringent control of Type I errors when compared to familywise error rate controlling procedures (e.g. Bonferroni correction).

The Fold Change (FC) is calculated by getting the ratio between the mean value of the selected variable in one group in comparison to the same value in another group, thus measuring how much a variable mean changes within two groups. It is an interesting measure in the sense that it allows for group discrimination according to the selected feature, given that the FC value is significant enough (usually $FC > 2$).

Many statistical tests rely heavily on distributional assumptions, such as normality. However, when these assumptions are not satisfied, commonly used statistical tests often perform poorly, resulting in a greater chance of committing an error. Non-parametric tests are designed to have desirable statistical properties when few assumptions can be made about the underlying distribution of the data. In other words, when the data are obtained from a non-normal distribution or one containing outliers, a non-parametric test is often a more powerful statistical tool than its parametric equivalent.

The Kruskal-Wallis H-test for Oneway ANOVA by Ranks is often viewed as the non-parametric equivalent of the parametric Oneway ANOVA. As a nonparametric test, it uses ranked data, and is particularly employed when: the data are ordinal and do not meet the precision of interval data, there are serious concerns about extreme deviation from normal distribution, and there is considerable difference in the number of subjects for each comparative group. This test is frequently used for when it is necessary to determine if three or more independent samples originate from the same population. A significant Kruskal-

Wallis test indicates that at least one sample stochastically dominates one other sample. The test does not identify, however, where this stochastic dominance occurs or for how many pairs of groups stochastic dominance obtains.

Kolmogorov-Smirnov Two-Sample test is another nonparametric test often used to determine if two independent samples are taken from either the same population or from two populations that have the same distribution pattern. It is sensitive to distribution differences, either in central tendency or dispersion differences, being more useful when these differences are due more to the latter case. The Kolmogorov-Smirnov test uses ordinal data and is especially useful with small samples, such as when there are fewer than 40 subjects in each of the two samples.

The Wilcoxon Signed-Rank test is a nonparametric hypothesis test often viewed as being similar to Student's t-test for matched pairs, but it is used for ordinal data or data that seriously violate any semblance of normal distribution. This method is employed when comparing two related samples (matched samples) or repeated measurements on a single sample to assess whether their population mean ranks differ (i.e. it is a paired difference test), focusing on both the magnitude and direction of the differences for matched pairs (MacFarland and Yates, 2016).

Perhaps the most widely used statistical technique is Regression analysis. Linear regression analysis allows the investigation and modeling of the relationship between a scalar dependent variable y and one or more independent variables (regressors) denoted x , assuming a linear relationship between them. This relationship is estimated through a mathematical equation (i.e. a linear model) which, in its most general form looks like:

$$y_i = \beta_0 + \beta_1 x_{i1} + \dots + \beta_p x_{ip} + \varepsilon_i, \quad i = 1, \dots, n \quad (1)$$

where y_i represents the dependent variable, $x_{i1} - x_{ip}$ the independent variables, β is a p -dimensional parameter vector and its elements called regression coefficients and, lastly, ε_i represents the error term. In almost all applications of regression, the regression equation is only an approximation to the true functional relationship between the variables of interest and are valid only over the region of the regressor variables contained in the observed data.

The case when there is only one independent variable is called simple linear regression, whereas for more than one independent variable the process is called multiple linear regression. The regression coefficients are often estimated using the least squares method, which attempts to minimize the sum of the squares of the errors made in the results of every single equation (Darlington and Hayes, 2016).

2.2.3 Unsupervised methods

Among the most commonly used unsupervised methods there are [Principal Component Analysis \(PCA\)](#) and clustering methods such as [Hierarchical Clustering Analysis \(HCA\)](#) and the k-means method.

The [PCA](#) is a statistical procedure whose principal aim is to reduce data dimension, by converting a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables (principal components). It aims to explain as much data variability as possible with as few principal components as possible. In general, principal components can be computed up to the total number of variables. It can be seen as a method to compute a new coordinate system formed by the latent variables, which is orthogonal, and where only the most informative dimensions are used. Latent variables from PCA optimally represent the distances between the objects in the high-dimensional variable space.

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, corresponding to 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). In this method, it is generally recommended to use mean-centered data, and because it is also sensitive with respect to outliers, robust [PCA](#) can be used ([Varmuza and Filzmoser, 2009](#)).

Cluster analysis tries to identify concentrated groups (clusters) of objects, without prior information about any group membership and/or number of clusters. In other words, cluster analysis tries to find groups containing similar objects, and usually one cannot expect a unique solution for cluster analysis. There are two main types of clustering methods, namely hierarchical and non-hierarchical clustering.

In the [Hierarchical Clustering Analysis \(HCA\)](#), objects and partitions are arranged in a hierarchy and represented in a tree-like dendrogram, being a complementary, nonlinear, and widely used method for cluster analysis ([Figure 4](#)). Strategies for [Hierarchical Clustering Analysis \(HCA\)](#) generally fall into two types: agglomerative and divisive clustering.

In the first approach, each of the n objects forms a separate cluster, resulting in n clusters in the first level. In the next level the two closest clusters are merged, and so on, until finally all objects are in one single cluster. The divisive method groups all n objects in one single cluster, in the first level of the hierarchy. In the next level, this cluster is split into two smaller clusters, and so on, until finally each object forms a separate cluster. Usually one cannot expect a unique solution for cluster analysis. The result greatly depends on

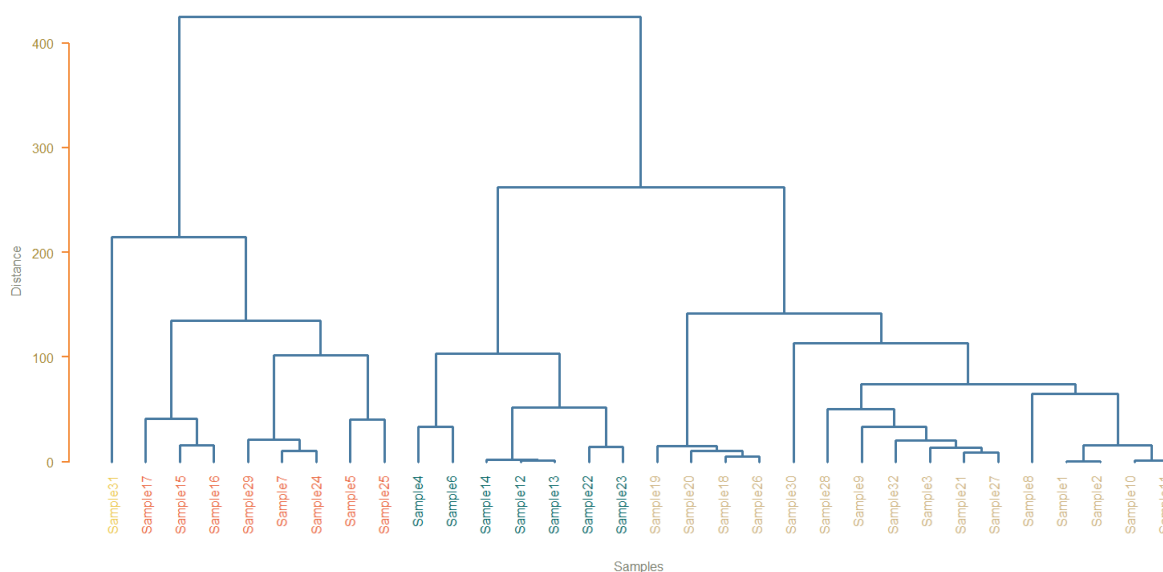


Figure 4: Example of a dendrogram resulting from a cluster analysis performed over 32 samples. The distance between samples is represented in the y axis, whereas sample names are represented in the x axis.

the used distance measure (e.g. Euclidean or Manhattan distances), the cluster algorithm (e.g. Nearest Neighbor or complete linkage), and the chosen parameters (Varmuza and Filzmoser, 2009).

Unlike HCA, non-hierarchical clustering methods aim to partition a dataset into a pre-defined number of clusters, organizing data objects into a set of typically non overlapping flat groups, by typically using iterative algorithms that optimize a chosen criterion. One of the most used non-hierarchical clustering formulations is the k-means clustering. An algorithm for this approach starts from a set of initial random clusters, then proceeding to take each point belonging to a given data set and associate it to the nearest center, minimizing the distance of the observation to the cluster mean. This last step is repeated until no improvement in the objective function can be made. This algorithm (named Lloyd algorithm) is usually fast, however, given that it is an heuristic algorithm, there is no guarantee that it will converge to the global optimum, and the result may depend on the initial set of clusters.

2.2.4 Supervised methods: machine learning

Machine learning techniques are often used when performing a metabolic fingerprinting experiment. It focuses on the construction of algorithms that can learn from and make predictions on data, through building a model from sample inputs. In order to build a

predictive model, a set of training data must be provided, and the learner algorithm must have the ability to generalize from its experience, by performing accurate predictions on new, unseen examples.

In machine learning, tabular data is the most common way of representing the data. It consists in a data table with rows representing the different samples, or X , and a single or multi-column property, or y , that is known for each example (Figure 5). The properties are usually the interesting facts of the examples. When properties are of continuous nature it is called a regression approach, whereas if they are of discrete nature it is called a classification approach (Varmuza and Filzmoser, 2009). The general workflow of a machine learning approach is represented in Figure 6.

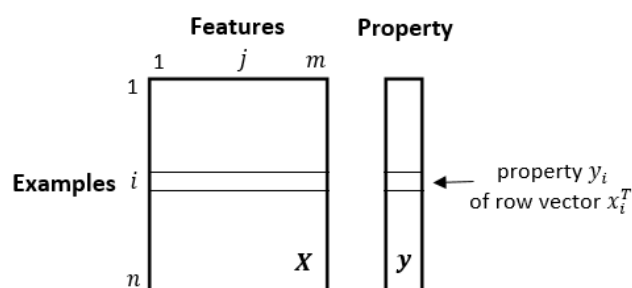


Figure 5: Graphical representation of tabular data used in a machine learning approach, including feature matrix X and a property vector y .

Among the most popular supervised methods for classification are: [Partial Least Squares Discriminant Analysis \(PLS-DA\)](#), [Linear Discriminant Analysis \(LDA\)](#), [Soft Independent Modelling by Class Analogy \(SIMCA\)](#) and Decision trees. Methods for regression tasks include [Partial Least Squares Regression \(PLS-r\)](#) and Regression trees. Methods such as [K-Nearest Neighbors \(KNN\)](#), [Support Vector Machines \(SVM\)](#), [Artificial Neural Network \(ANN\)](#) and Random Forests can be used for both classification and regression tasks.

[Partial Least Squares \(PLS\)](#) is a method to relate a matrix X of independent variables to a vector y or to a matrix Y of dependent variables. In the model structure of [PLS-r](#), the X -data is first transformed into a set of intermediate linear latent variables (components). Since it is a linear method, the final latent variable that predicts the modeled property, y , is a linear combination of the original variables. During model development, a relatively small number of [PLS](#) components are calculated, and it is the number of such components that determines the complexity of the model, which can be optimized for high prediction performance. [PLS-DA](#) is a variant used when the Y is categorical (Varmuza and Filzmoser, 2009).

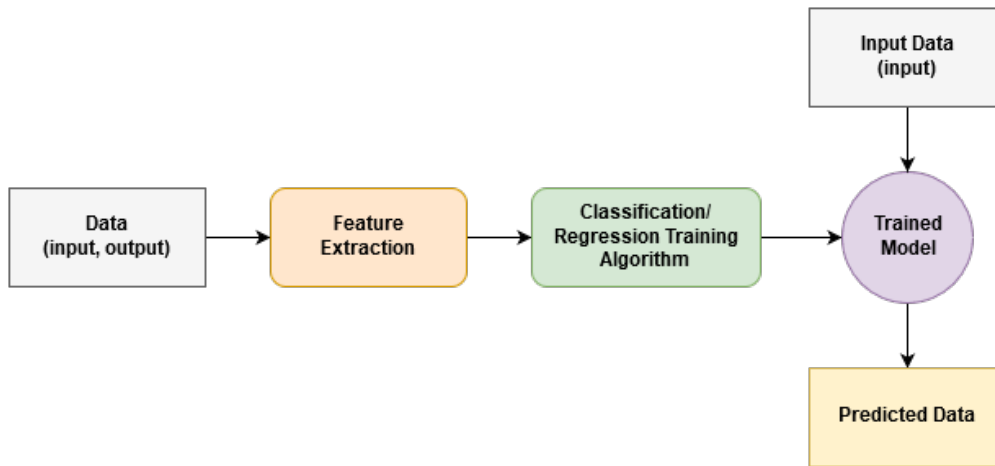


Figure 6: General workflow of a machine learning approach.

LDA attempts to find a linear combination of features (latent variable) that characterizes or separates two or more classes of objects or events. The resulting combination is commonly used as a linear classifier. The representation of a **LDA** model consists of statistical properties of the data, which are calculated for each class and then used to make predictions. For a single input variable, x , these properties are the mean and the variance of the variable for each class, whereas for multiple variables the properties consist in the means and covariance matrix. This method assumes a Gaussian distribution of the data and that each attribute has the same variance. It is closely related to **PCA**, although **PCA** does not take into consideration the underlying class structure, being sometimes used for data dimensionality reduction (Martínez and Kak, 2001).

The **SIMCA** method is based on disjoint principal component models. The idea is to describe the multivariate data structure of each group separately in a reduced space using **PCA**. The special feature of this method, is that **PCA** is applied to each group separately and also the number of principal components is selected individually and not jointly for all groups. Due to the use of **PCA**, this approach works even for high-dimensional data with rather a small number of samples, whereas methods like **LDA** can become unstable under such conditions. 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, 2009).

Decision trees consist in a flowchart-like structure in which internal nodes represent the test on an attribute, the branches represent the outcome of the test and the leaves represent a class label. The learning phase is done by splitting the source set into subsets based on an attribute value test, repeating this process in a recursive manner until the subset at a node has the same value of the target variable, or when splitting no longer adds value to the predictions. When the target variable is of discrete nature it is called a classification tree,

whereas when the target variable is of continuous nature it is called a regression tree. A Random Forest classifier builds a large collection of uncorrelated trees, and then averages them to improve the classification rate. It uses the bagging method which helps to reduce variance and corrects the decision's tree habit of overfitting to their training set (Hastie et al., 2009).

In contrast to the already mentioned machine learning methods, the KNN method does not require a model to be fit. It is a type of lazy learning method, where the function is only approximated locally and all computation is deferred until classification. For KNN classification, the task is to predict the class membership of a new object x . Using, for instance, the Euclidean distance measure, the k -nearest neighbors (of the training data) to x are determined. The neighbors are found by calculating the distances between the new object and all objects in the training set. This method has the advantage of neither requiring linearly separable groups nor compact clusters for the groups, being easily applied to multi-class problems (Varmuza and Filzmoser, 2009).

Given a set of training examples and two possible categories for each example, the SVM algorithm builds a model capable of assigning examples to one of the two categories, making it a non-probabilistic binary linear classifier. It produces linear boundaries between object groups in a transformed space of the x -variables, usually of much higher dimension than the original x -space. These class boundaries are constructed to maximize the margin between the groups. New examples are then mapped into that same space and predicted to belong to a category based on which side of the decision boundary they fall (Varmuza and Filzmoser, 2009).

In ANNs, the central idea is to extract linear combinations of the inputs as derived features, and then model the target as a nonlinear function of these features. An ANN is a two-stage regression or classification model, typically represented by a network diagram, where neural units are interconnected. This network usually has three layers, consisting in an input layer, an hidden layer and an output layer. The input data goes into the first layer, the hidden layer nodes do some calculations and then the output is gathered from the last layer. The links between different neural units can be enforcing or inhibitory in their effect on the activation state of connected neural units, usually through a limiting (threshold) function (Hastie et al., 2009).

Some of these (and other) machine learning methods used in spectral data analysis studies available in the literature are listed on the fifth column of Table 1.

2.2.5 Feature selection

Machine learning methods have a difficulty in dealing with the large number of input features and, therefore, pre-processing of the data is essential to use these methods effectively. Feature selection is an important technique which has become indispensable in the machine learning process. It consists in the process of detecting relevant features and removing irrelevant, redundant, or noisy data. This technique greatly speeds up machine learning algorithms, also improving predictive accuracy and comprehensibility.

There are three feature selection approaches: filters, wrappers and embedded methods. The filter approach incorporates an independent measure for evaluating features subsets without involving a learning algorithm. This method can, however, miss features that are not useful alone but can be very useful in combination with others. On the other hand, the wrapper approach uses a learning algorithm for subset evaluation. This method thus selects an optimal subset that is best suited to the learning algorithm having, therefore, a better performance when compared to the filter approach in most cases. Embedded methods have been recently proposed and try to combine the advantages of both previous methods. In this method, the learning algorithm takes advantage of its own variable selection process and performs feature selection and classification simultaneously (Kumar and Minz, 2014). A graphical representation of filter, wrapper and embedded approaches is shown in Figure 7.

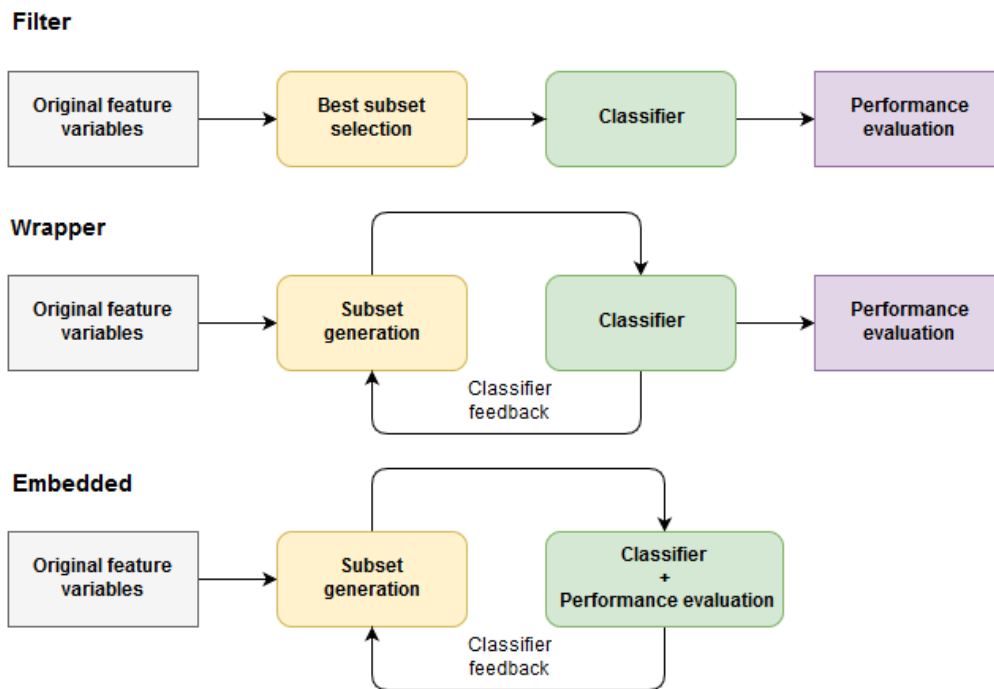


Figure 7: Workflow of Filter, Wrapper and Embedded approaches in feature selection.

Filter approaches are based on statistical tests (e.g. chi-squared test) measuring some intrinsic properties of the dataset (or features), including information gain, variance threshold and the correlation coefficient. The latter method was used for instance in the predictive biomarker discovery in biological samples (Grissa et al., 2016).

Wrapper methods include Recursive Feature Elimination (RFE), Sequential Feature Selection (SFS) algorithms and Genetic Algorithms (GA). The RFE method uses all initial features to fit the model, ranking all features according to their contribution. In each subset, most relevant variables are retained and the model is refitted. This process continues until the subset with best performance is obtained. While the RFE method uses the feature weight coefficients or feature importance, the SFS method removes (or adds) features based on a user-defined classifier/regression performance metric, until a feature subset of the desired size k is reached.

GAs are metaheuristic optimization algorithms that use an initial population of candidate solutions (individuals), which is then evolved toward better solutions. This is done by an iterative process, where in each iteration (generation) the fitness (i.e. value of the objective function) of every individual in the population is evaluated. The fittest individuals are stochastically selected from the current population and recombined to form a new generation. The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. This method was used for instance in bacteria discrimination using FTIR spectroscopy (Preisner et al., 2007).

Embedded methods use for instance decision trees and the Least Absolute Shrinkage and Selection Operator (LASSO) regression algorithm for generalized linear models. LASSO penalizes the absolute size of the regression coefficients (i.e. forces their sum to be less than a fixed value), which forces certain coefficients to be set to zero. It is a convenient method for automatic feature selection when dealing with highly correlated predictors. This method has been applied for instance in the diagnosis of insulin resistance (Milburn and Lawton, 2013).

2.3 DATA FUSION

Data fusion is a process of combining data from different sources to improve the performance of prediction models. It deals with association, detection, correlation and estimation of data to achieve a better information of the system's state. In the fusion process, data is collected by N different source types (e.g. IR and UV-vis data). The data can then be pre-processed to extract a feature vector that represents the observed data, and a ma-

chine learning approach, using the observed objects, may be performed. The output of this process must be partitioned into groups representing observations belonging to the same category and, finally, the fusion algorithms combine the multi-source data to obtain a result that has less uncertainty than it would if these sources were used individually.

There are three main categories of data fusion, depending on the abstraction level the fusion of identity declarations takes place: **Low-Level Fusion (LLF)**, **Intermediate-Level Fusion (ILF)** and **High-Level Fusion (HLF)**. A graphical representation of these approaches is shown in Figure 8.

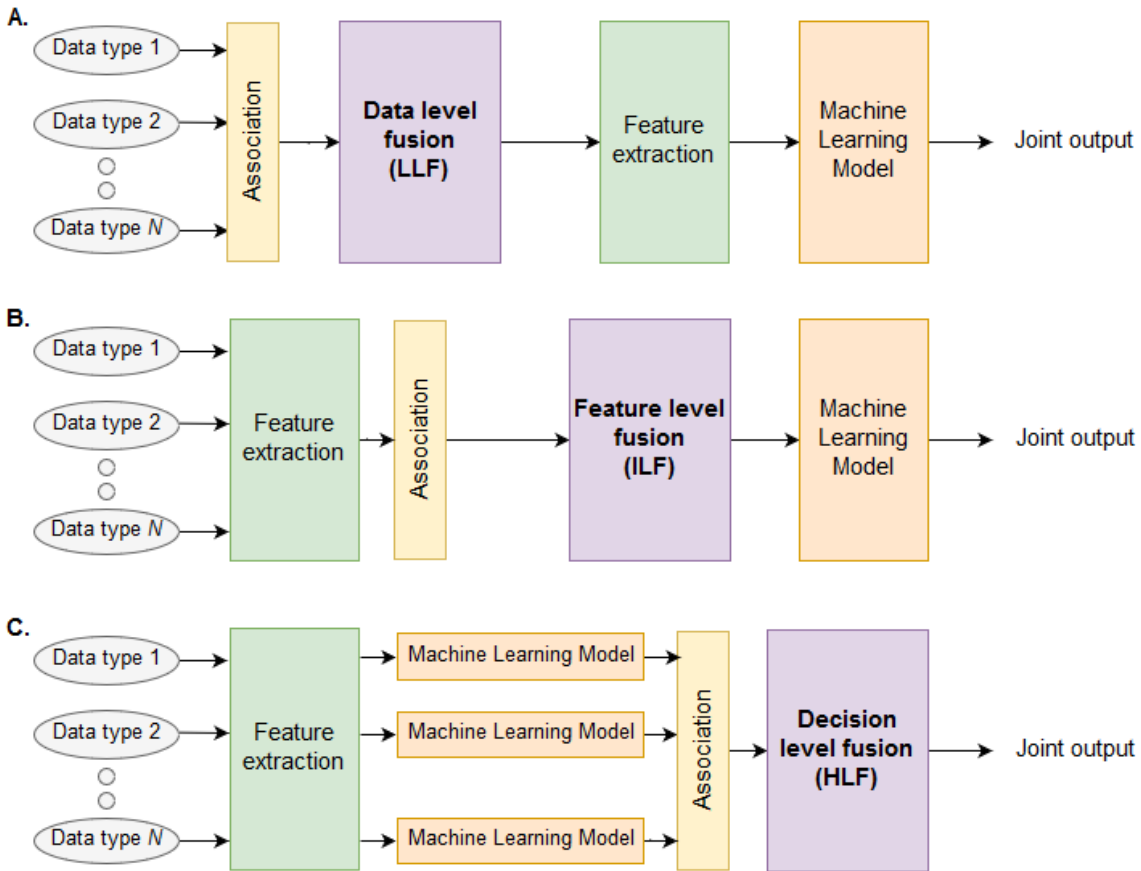


Figure 8: Graphical representation of **A. Low-Level Fusion (LLF)**, **B. Intermediate-Level Fusion (ILF)** and **C. High-Level Fusion (HLF)**.

LLF is made on a data level, by direct association and combination of raw data, representing measures of the same physical phenomena. After data combination, a feature vector is extracted and used in a machine learning process. It provides the most accurate results, assuming proper data association.

ILF on the other hand is made on a feature level. Here, a representative features vector is extracted directly from the data. After data alignment and association, the feature vectors

are concatenated into a single vector acting as an input for a machine learning process. The output is, therefore, based on the combined feature vectors from all of the data types.

Lastly, **HLF** is made on a decision level. Initially, the data from each of the data types are used in a machine learning process, which can be coupled with feature extraction (e.g. using neural networks). Data association and correlation are still required to ensure that the data to be fused refer to the same physical entity. Finally, the the results from the machine learning process using each of the data types are combined using decision level fusion techniques (e.g. Bayesian inference) (Fourati, 2016).

LLF and **ILF** approaches were applied for instance in the classification of pure and adulterated honey (Subari et al., 2012). Combining e-nose and **IR** data they found these two approaches achieved better results than single modality data. An **HLF** approach using Bayesian inference was applied in the discrimination of white grape varieties (**FTIR** and **UV-vis** data), having achieved half the misclassification error when compared to the use of single modality data (Roussel et al., 2003).

2.4 AVAILABLE FREE TOOLS FOR METABOLOMICS AND SPECTRAL DATA

In response to the increasing growth of information output through the years, a number of computational tools for metabolomics and spectral data have become available. Among these are some interesting packages on the open-source R scientific computing platform (<http://www.r-project.org>), including *hyperSpec* and *ChemoSpec*.

The *hyperSpec* package allows convenient handling of Hyperspectral data (i.e. spectra with associated space, time or other additional information). It handles data recorded over a discretized axis, obtained from **UV-vis**, **NIR**, **IR**, Raman and other spectroscopy techniques. It has several plot functions to display spectra, false-colour maps, calibration curves, among other purposes. Preprocessing methods include data normalization, intensity calibration, offset and baseline corrections, spectral interpolation, and many other methods. For the analysis, this package provides functions for clustering analysis, **PCA**, **PLS**, **LDA**, among others (Beleites and Sergo, 2016).

The *ChemoSpec* package was designed with metabolomics data sets in mind, where the samples fall into groups, such as treatment and control. It consists in a collection of functions for entirely exploratory and unsupervised data analysis of spectral data, including **IR**, **UV-vis** and Raman data, among other types of spectral data. It includes functions for plotting and inspecting spectra, as well as some preprocessing functions for data normalization and binning, identifying and removing problematic samples or regions of no interest,

baseline correction, peak alignment, among others. Unsupervised methods such as [HCA](#), [PCA](#) and model-based clustering are also covered in this package ([Hanson, 2016](#)).

Among web-based tools available for metabolomics and spectral data analysis the most notable and comprehensive is *MetaboAnalyst*. It accepts data from either targeted profiling (concentration tables) or metabolic fingerprinting approaches (spectral bins, peak lists) produced from either [NMR](#), [LC-MS](#) or [GC-MS](#). Preprocessing of the data is available, including normalization and scaling, data transformation, outlier removal, among other methods. The statistical analysis module offers various commonly used statistical and machine learning methods, including t-tests, [ANOVA](#), [PCA](#), [PLS-DA](#), Orthogonal [PLS-DA](#), and also clustering and visualization tools to create dendrograms and heatmaps as well as to classify based on random forests and [SVM](#). *MetaboAnalyst* includes modules for other types of analysis as well, including the enrichment analysis, pathway and time-series analysis modules, among others.

Table 2: Available free tools for metabolomics and spectral data.

Name	URL	Description	Data types
chemometrics	https://CRAN.R-project.org/package=chemometrics	R package for multivariate statistical analysis in chemometrics	Chemical data
ChemoSpec	https://CRAN.R-project.org/package=ChemoSpec	R package for exploratory analysis of spectral data	NMR , IR and Raman
COLMAR	https://spin.ccic.ohio-state.edu/index.php/colmar	Webserver for NMR data analysis and compound identification	NMR
hyperSpec	https://CRAN.R-project.org/package=hyperSpec	R package for Hyperspectral data analysis	UV-vis , IR , NMR , MS , Raman , ...
MeltDB	https://meltdb.cebitec.uni-bielefeld.de/cgi-bin/login.cgi	Web-based system for metabolomics data analysis and dataset annotation	GC-MS and LC-MS
MetaboAnalyst	http://www.metaboanalyst.ca	Web-based system for metabolomics and spectral data analysis	NMR , LC-MS and GC-MS
metabolomics	https://CRAN.R-project.org/package=metabolomics	R package for metabolomics data analysis	NMR , GC-MS , LC-MS and MS
MetaboMiner	https://wishart.biology.ualberta.ca/metabominer/	Java based software for NMR data analysis and compound identification	NMR
metaP-Server	http://metap.helmholtz-muenchen.de/metap2/	Web-based system for metabolomics data analysis	NMR , GC-MS , LC-MS and MS
muma	https://CRAN.R-project.org/package=muma	R package for metabolomics univariate and multivariate analysis	NMR , GC-MS , LC-MS and MS
MVAPACK	https://bionmr.unl.edu/mvapack.php	Toolkit for NMR and MS data handling	NMR and MS
OpenMS	https://www.openms.de/	C++ library for LC-MS data handling and analysis	LC-MS
specmine	https://CRAN.R-project.org/package=specmine	R package for the integrated analysis of metabolomics and spectral data	NMR , GC-MS , LC-MS , UV-vis , IR , Raman , ...

These and other freely available tools are listed in [Table 2](#). Of all the listed tools, only some of the R packages include functions for spectral data such as [IR](#), [UV-vis](#) and [Ra-](#)

man data, while none of the web-based tools address these types of data. This is a major disadvantage among these web-based tools, since they mostly work with NMR and chromatography data, leaving the already mentioned spectroscopies data aside, which are commonly employed in metabolomic fingerprinting experiments. The *specmine* package provides a set of methods for metabolomics data analysis, including data loading in different formats, pre-processing, metabolite identification, univariate and multivariate data analysis, machine learning and feature selection. It was the base for the development of this work and, therefore, will be discussed in detail in section 3.2.

2.5 OTHER GENERAL FREE TOOLS

There are other freely available tools that, although not specific for metabolomics analysis tasks, may be of great value for such analysis. These include, for instance, packages on the open-source R scientific computing platform (<http://www.r-project.org>), namely *rminer* and *caret* packages.

The *rminer* package facilitates the use of data mining algorithms in classification and regression tasks by presenting a short and coherent set of functions. *rminer* currently has 16 classification and 18 regression methods available, including functions for PLS-r, LDA, random forests, SVM and KNN methods, among others. Time Series Forecasting (TSF) is also included, which is a special case of regression and involves the analysis of a time ordered phenomenon. This package offers a large range of evaluation metrics and graphs that can be used to evaluate the quality of the fitted models and extract knowledge learned from the data-driven models. It also adjusts the hyperparameters of the models, performing some feature selection methods (Cortez, 2016).

The *caret* package consists in set of functions for training and plotting classification and regression models. It contains some functions for data visualization in the form of boxplots, scatter and density plots, as well as data preprocessing functions for data centering and scaling, missing values handling and data transformation, among others. *Caret* also has feature selection functions, implementing both filter and wrapper approaches. Available regression and classification methods in this package include PLS-r, LDA, KNN, SVM, neural networks, random forests, among many others. Tuning model parameters is also possible, with functions that allow to choose the best set of parameters for a given model, estimating the model performance using a training set (Kuhn et al., 2016).

DEVELOPMENT

In this chapter, the software development process will be covered, including the adopted strategy and tools used in the development of the web based platform. The different modules, as well as the data model used for project, dataset and user management will also be explained, covering the encryption process for secure data storage in the database.

3.1 DEVELOPMENT STRATEGY AND TOOLS

A web based platform with features covering the main steps of the metabolomics data analysis workflow has been developed. It contains modules for data reading and dataset creation, data preprocessing and analysis, all implemented using functions from the R package *specmine*. A metabolite identification module is also included, however, considering this work focuses on spectral data (IR, UV-vis and Raman), where generally such type of analysis is not performed, this module will not be covered here. The developed web platform aims, therefore, to implement most of *specmine* features in user friendly graphical interface. It includes an authentication system, allowing the user to have his own personal workspace where projects can be stored and accessed later, with the option to share projects with other users. It is important to mention that the platform had a shared development and, therefore, only the modules I contributed to will be here discussed.

The development of this web based platform had the user in mind, being easy to use (the user does not need to know any kind of programming language) and providing at the same time abundant graphical visualization of the results, so that they can be easily interpretable. It was developed in a way that every result, either in text, table or graphical format, could be made available to the user through download.

The chosen programming language for the development of the platform was the R environment (<http://www.r-project.org>), which is a free integrated software environment for data manipulation, scientific and statistical computing and graphical visualization. It is

characterized as an effective data handling and storage facility, with a large, coherent, integrated collection of intermediate tools for data analysis and graphical display. It allows users to add additional functionality by defining new functions and also develop new packages, contributing to the already large and available collection of R packages.

More specifically, the *shiny* library was used (<https://shiny.rstudio.com/>), allowing an easy way to build interactive web applications with R. A Shiny application has two components: a user-interface (ui) script and a server script. The user-interface script controls the layout and appearance of the application, whereas the server script contains the instructions needed to build the application. Shiny works based on reactive programming, in which there are three kinds of objects: reactive sources, reactive conductors and reactive endpoints (Figure 9).

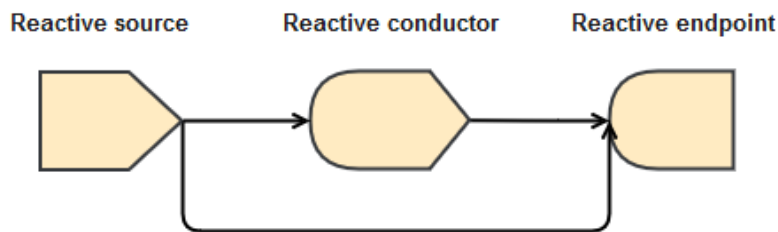


Figure 9: Representation of reactive programming objects in a Shiny application.

The reactive source typically consists in the user input through a browser interface (e.g. selecting an item, typing input). A reactive endpoint is usually something that appears in the user's browser window, such as a plot or a table of values. A reactive source can be connected to multiple endpoints, and vice versa. Most simple examples use just these two components, wiring up sources directly to endpoints. However, it is also possible to put reactive components between the sources and endpoints, namely reactive conductors. These can be useful for encapsulating slow or computationally expensive operations, making sure code does not run more times than the absolutely necessary.

Other R libraries were also used. These include:

- *shinydashboard*: allows the construction of a shiny application with a typical dashboard appearance;
- *shinyBS*: adds additional Twitter Bootstrap components to Shiny including, for instance, modal windows;
- *shinyjs*: allows to perform common useful JavaScript operations in Shiny apps like hiding, resetting or disabling elements;
- *DT*: allows data objects in R to be rendered as HTML tables;
- *RMySQL*: a database interface to MySQL;

- *bcrypt*: for string encryption (used for the authentication system);
- *GGally*: an extension of the graphics package *ggplot2*;
- *shinyWidgets*: adds better looking custom inputs widgets to shiny applications;
- *colourpicker*: has a colour picker that can be used as an input in a shiny application, useful for interactively changing the color of some plots, for instance.

More importantly, the *specmine* package was used, providing a set of methods for metabolomics data analysis, including data loading in different formats, pre-processing, metabolite identification, univariate and multivariate data analysis, machine learning and feature selection. The package functionalities will be introduced in detail in the next section, given its importance for this work.

The [Integrated Development Environment \(IDE\)](#) chosen to develop the R scripts and build the platform was RStudio (<https://www.rstudio.com/>). It is written in the C++ programming language, having an intuitive interface with many useful tools, making it easier to work with R. RStudio is a free software that has many features, including a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

Reports with the analysis results generated on the platform were made using the RStudio plug-in named R Markdown (<http://rmarkdown.rstudio.com/>). R Markdown uses markdown syntax (<https://daringfireball.net/projects/markdown/>) coupled with R code chunks that are run, displaying the code's output in the generated report. The report is generated using the *knitr* package (<https://yihui.name/knitr/>), the engine for dynamic report generation with R, and it can be in the form of a HTML, PDF or Microsoft Word document.

The database for project, dataset and user management was built using the open-source [Relational Database Management System \(RDBMS\)](#) MySQL (<https://www.mysql.com/>). It is written in C and C++, being a fast, stable and multi-user, multi-threaded [Structured Query Language \(SQL\)](#) database server. SQL consists of a data definition, manipulation and control language. It is a special-purpose domain-specific language used in programming and designed for stream processing or managing data held in a [RDBMS](#). The scope of SQL includes data insert, query, update and delete, schema creation and modification, and data access control.

3.2 SPECMINE, AN R PACKAGE FOR METABOLOMICS DATA ANALYSIS

As discussed in [section 2.4](#), most freely available tools are limited to specific types of metabolomics or spectral data and some offer a limited portfolio of data analysis tools for the construction of analysis pipelines.

To address this problem, the R package *specmine* was made available ([Costa et al., 2016](#)). It was developed under the R environment which is a free development environment for data manipulation, scientific and statistical computing and graphical visualization. *Specmine* provides a set of methods for metabolomics and spectral data analysis, including data loading in various formats, data preprocessing, metabolite identification, univariate and multivariate data analysis, machine learning and also feature selection.

The implemented methods allow for the analysis of metabolomics and spectral data, including [GC-MS](#), [LC-MS](#), [NMR](#), [IR](#) and [UV-vis](#) data, integrating many available functions provided by other metabolomics oriented R packages and also more general-purpose data analysis R functions. Some of *specmine* package dependencies include:

- *hyperSpec*: facilitates hyperspectral data sets handling (i.e. spatially or time-resolved spectra, which may consist of any data that is recorded over a discretized variable);
- *ChemoSpec*: a collection of functions for top-down exploratory data analysis of spectral data obtained via [NMR](#), [IR](#) or Raman spectroscopy;
- *rgl*: provides medium to high level functions for 3D interactive graphics;
- *ggplot2*: a system for 'declaratively' creating graphics, based on "The Grammar of Graphics";
- *caret*: miscellaneous functions for training and plotting classification and regression models.

Besides providing a tool that covers the main metabolomics and spectral data types, *specmine* also addresses a full range of tasks in data analysis, allowing for the creation of flexible and powerful analysis pipelines for specific case studies, providing abundant graphical visualization of the results. [Figure 10](#) shows the modules present in this package. Since metabolite identification is not usually the goal in a metabolic fingerprinting approach, which is emphasized throughout this dissertation, the metabolite identification module won't be discussed here.

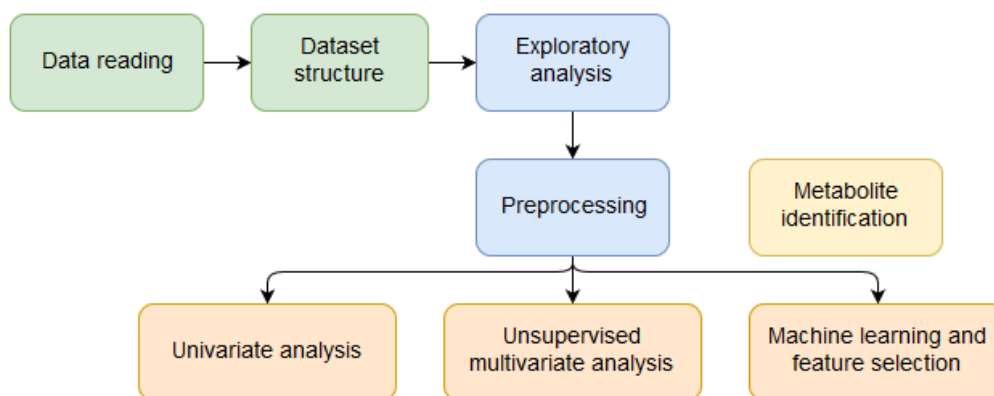


Figure 10: Modules in the *specmine* package. Adapted from Costa et al. (2016).

3.2.1 Data reading and dataset structure

Specmine supports a number of different file formats, including comma (or tab) separated values (CSV or TSV) files, (J)DX spectra files, NetCDF, mzDATA and mzXMLMS data. The metadata file can be given in the CSV/TSV format. Data can also be loaded as a peaks list, which are converted into a dataset using peak alignment functions.

The structure of the dataset used in this package is independent of the data type and source and consists in an R list with the following fields: description of the dataset, the type of data, the data matrix, the metadata data frame and the labels for the x and y-axis. A graphical representation of the dataset structure is represented in Figure 11.

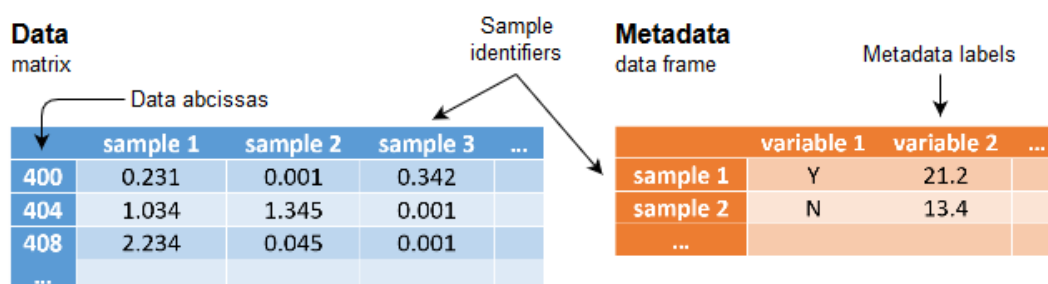


Figure 11: Representation of the dataset structure in *specmine*. Adapted from Costa et al. (2016).

A list of all *specmine* functions regarding data reading and dataset structure is shown in Table 3.

Table 3: *Specmine* package functions regarding data reading and dataset structure

Function name	Description
check_dataset	Check if the dataset is valid and if not give the proper error message.
convert_from_chemospec	Convert the dataset in the ChemoSpec format to a dataset of this package.
convert_from_hyperspec	Convert the dataset in the hyperspec format to a dataset of this package.

Table 3: *Specmine* package functions regarding data reading and dataset structure. (Continued)

Function name	Description
<code>convert.to.hyperspec</code>	Convert a dataset to an hyperspec object.
<code>create_dataset</code>	Create a dataset from existing objects
<code>dataset.from.peaks</code>	Converts a peak list to a dataset.
<code>is_spectra</code>	Check if the dataset is from spectral data where x.values are numeric.
<code>read.csvs.folder</code>	Reads multiple CSV files in a given folder.
<code>read_dataset.csv</code>	Reads the data from a CSV file and creates the dataset.
<code>read_dataset.dx</code>	Reads the data from the (J)DX files and creates the dataset.
<code>read_dataset.spc</code>	Reads the data from the SPC files and creates the dataset.
<code>read_data.csv</code>	Reads the data from the CSV file.
<code>read_data.dx</code>	Reads the data from the (J)DX files.
<code>read_data.spc</code>	Reads the data from the SPC files.
<code>read_metadata</code>	Read the metadata from a file.
<code>read.ms.spectra</code>	Read the data from the MS files and creates the dataset.
<code>read_multiple.csvs</code>	Reads multiple CSVs, each one with a sample.

3.2.2 Exploratory analysis and data pre-processing

Specmine includes functions that allow to calculate global statistics and visualize the data in a graphical way. The package can calculate the main descriptive statistics over the data matrix of a dataset for both variables and samples, having functions that can be applied over the entire dataset or to a subset of samples or variables. Graphical visualization of the data is done for instance in the form of boxplots, which allows to see the distribution of values for a set of variables. The package also provides functions for spectra plotting, where the variables are represented by numerical values. Visualization functions rely on both `ggplot2` and the base graphics system of R.

Preprocessing methods are also provided for the different types of data. They include methods for extracting relevant parts of a dataset, namely subsets of samples, data and metadata variables. Spectral preprocessing methods include functions for shifting correction, multiplicative scatter correction, first derivative, baseline, offset and background corrections. Some methods for smoothing interpolation are also available, including bin or loess smoothing, as well as Savitzky-Golay filters. Missing values can be treated by either removing samples and/or variables that have a number of missing values above a given threshold or replaced using a variety of different methods.

Specmine also includes functions for data normalization, which can be done by sum, median, a reference sample or feature, for data transformation using cubic root and logarithmic methods, and scaling using auto, range and pareto methods. The package also provides flat pattern filters with distinct metrics and parameters that allow to remove vari-

ables with low variance. The various *specmine* functions for data exploratory analysis and preprocessing are listed in Table 4.

Table 4: *Specmine* package functions for data exploratory analysis and preprocessing.

Function name	Description
absorbance_to_transmittance	Converts absorbance values to transmittance values.
aggregate_samples	Aggregate samples according to an aggregate function like mean, median, etc.
apply_by_group	Apply a function to samples from a given metadata's group.
apply_by_groups	Apply a function to samples from a metadata's variable.
apply_by_sample	Applies a function to the values of each sample
apply_by_variable	Applies a function to the values of each variable
background_correction	Perform background correction on the spectra.
baseline_correction	Performs baseline correction on the dataset.
boxplot_variables	Boxplot of each variable of the dataset.
boxplot_vars_factor	Boxplot of variables with metadata's variable factors from the dataset.
compare_regions_by_sample	Compare two regions of a dataset by samples.
convert_to_factor	Convert a metadata's variable to factor.
count_missing_values	Counts the missing values on the dataset.
count_missing_values_per_sample	Counts the missing values on each sample of the dataset.
count_missing_values_per_variable	Counts the missing values on each variable of the dataset.
cubic_root_transform	Performs cubic root transformation on the data matrix.
data_correction	Perform spectra corrections with 3 different methods.
find_equal_samples	Finds samples that have the same peak values - x and y (equal data frames)
first_derivative	Calculates the first derivative of the data.
get_data	Get the data matrix from dataset
get_data_as_df	Get the data matrix from the dataset as a data frame.
get_data_value	Get a data value given the x-axis labels and the sample
get_data_values	Gets the values of all samples in the dataset given a set of x axis names or indexes.
get_metadata	Get the metadata from the dataset
get_metadata_value	Get the metadata value
get_metadata_var	Get the values of a metadata variable from the dataset.
get_peak_values	Gets the peak values from a data frame of samples' peaks.
get_samples_names_dx	Function to get the names of the DX files from a folder.
get_samples_names_spc	Function to get the names of the SPC files from a folder.
get_sample_names	Get the sample names from the dataset.
get_type	Get the type of the data from the dataset
get_value_label	Get the value label from the dataset
get_x_label	Get the x-axis label from the dataset.
get_x_values_as_num	Get the x-axis values from the dataset as numbers.
get_x_values_as_text	Get the x-axis values from the dataset as text.
group_peaks	Group peaks with peak alignment.
impute_nas_knn	Impute missing values with KNN
impute_nas_linapprox	Impute missing values with linear approximation.
impute_nas_mean	Impute missing values with mean
impute_nas_median	Impute missing values with median
impute_nas_value	Impute missing values with value replacement.
indexes_to_xvalue_interval	Returns x-values corresponding to a vector of indexes (only to numerical values - spectra)
log_transform	Performs logarithmic transformation on the data matrix.
low_level_fusion	Low level fusion method for integrate different datasets (only samples with the same name on all datasets will be merged)

Table 4: *Specmine* package functions for data exploratory analysis and preprocessing. (Continued)

Function name	Description
mean_centering	Performs mean centering on the dataset.
merge_datasets	Merges two datasets with the same variables and metadata's variables.
merge_data_metadata	Merges the data and metadata from the dataset into a single data.frame.
metadata_as_variables	Use one or more metadata variables as variables.
missingvalues_imputation	Treats the missing values of a dataset according to a specific method.
msc_correction	Perform multiplicative scatter correction on the spectra.
multiplot	Multiplot from <i>ggplot2</i> package
normalize	Normalize the data from the dataset with a specific method.
normalize_samples	Normalize the data from a datamatrix with a specific method.
num_samples	Get the number of samples from a dataset.
num_x_values	Get the number of x-axis values.
offset_correction	Perform offset correction on the data.
peaks_per_sample	Counts number of peaks in a sample (given its index).
peaks_per_samples	Calculates the number of peaks on each sample.
plotvar_twofactor	Plot variable distribution on two factors from the dataset.
plot_spectra	Plot spectra from dataset.
plot_spectra_simple	Plot spectra from dataset (simple version).
remove_data	Remove data from the dataset.
remove_data_variables	Remove data variables from the dataset.
remove_metadata_variables	Remove metadata's variables from the dataset
remove_peaks_interval	Removes peaks from a given interval.
remove_peaks_interval_sample_list	Removes peaks on a sample list given a peak interval.
remove_samples	Remove samples from the dataset.
remove_samples_by_nas	Remove samples from the dataset by the number of NAs
remove_samples_by_na_metadata	Remove samples from the dataset with the metadata's variable value with NAs.
remove_variables_by_nas	Remove variables from the dataset by the number of NAs
remove_x_values_by_interval	Remove an interval of x-values from the dataset.
replace_data_value	Replace a data value for a new value on the dataset.
replace_metadata_value	Replace a metadata's variable value of a sample.
savitzky_golay	Smoothing and derivative of the data using Savitzky-Golay.
scaling	Performs scaling according to a method.
scaling_samples	Performs scaling according to a method.
set_metadata	Updates the dataset's metadata with a new one.
set_sample_names	Set new samples names to the dataset.
set_value_label	Set a new value label for the dataset.
set_x_label	Set a new x-label to the dataset.
set_x_values	Set new x-values to the dataset
shift_correction	Shifts the spectra according to a specific method.
smoothing_interpolation	Performs smoothing interpolation according to a specific method.
snv_dataset	Performs Standard Normal Variate on the dataset.
stats_by_sample	Get a summary of statistics of the samples.
stats_by_variable	Get a summary of statistics of the variables.
subset_by_samples_and_xvalues	Gets a subset of specific samples and x-values.
subset_metadata	Subsets the metadata according to the specified metadata's variables.
subset_random_samples	Gets a subset of random samples from the dataset.
subset_samples	Gets a subset of specific samples from the dataset.
subset_samples_by_metadata_values	Gets a subset of specific samples according to metadata's values from the dataset.
subset_x_values	Gets a subset of specific x-values from the dataset.
subset_x_values_by_interval	Gets a subset of a specific interval of x-values.

Table 4: *Specmine* package functions for data exploratory analysis and preprocessing. (Continued)

Function name	Description
sum_dataset	Returns a summary with its main features.
transform_data	Performs data transformation according to a method.
transmittance_to_absorbance	Converts transmittance values to absorbance values.
values_per_peak	Gets the number of values on each peak.
values_per_sample	Gets the number of values on each sample.
variables_as_metadata	Use one or more data variables as metadata variables.
xvalue_interval_to_indexes	Returns indexes corresponding to an interval of x-values (only to numerical values - spectra)
x_values_to_indexes	Returns the indexes corresponding to a vector of x-values (only to numerical values - spectra)

3.2.3 Univariate and unsupervised multivariate analysis

For univariate analysis, *specmine* offers a set of functions that cover various analysis types such as t-tests, [ANOVA](#), regression analysis, correlations and [FC](#) calculation. The package offers one-way [ANOVA](#), with the TukeyHSD post hoc test, and also multifactorial [ANOVA](#), with functions to summarize the main results, including p-values and the percentage of variation explained by the different factors. These p-values are already adjusted using the [FDR](#) method. Non-parametric tests include the Kruskal-Wallis and Kolmogorov-Smirnov tests.

For the linear regression analysis *specmine* offers functions that summarize the coefficients for the different factors and interactions and respective p-values. The correlations between variables or samples can be computed and the resulting matrix visualized as a heatmap. For [FC](#) analysis the package offers functions to calculate the fold changes of values considering two groups of samples, and the results can be visualized in both tabular and graphical forms, similarly to the t-tests results.

As for the unsupervised multivariate analysis *specmine* provides functions to perform [PCA](#) using two methods: classical and robust, where the latter makes use of the grid search algorithm to compute the desired number of principal components. The results of [PCA](#) can be visualized through a variety of graphs, including scree plots, scores plots, biplots and pairs plots.

The package also offers functions to perform k-means and hierarchical clustering methods. The distance method to use can be chosen as well as the method to use in the case of hierarchical clustering. K-means clustering results can be plotted in the form of a graph with the k clusters in different colors and hierarchical clustering results plotted as a dendrogram. The various *specmine* functions for univariate and unsupervised multivariate analysis are listed in [Table 5](#).

Table 5: *Specmine* package functions for univariate and unsupervised multivariate analysis.

Function name	Description
aov_all_vars	Perform analysis of variance of all variables in the dataset.
clustering	Perform cluster analysis on the dataset.
correlations_dataset	Calculate the correlations of all variables or samples in the dataset.
correlations_test	Performs correlations test to the whole dataset.
correlation_test	Performs correlations test of two variables or samples from the dataset.
dendrogram_plot	Plot dendrogram of hierarchical clustering results.
dendrogram_plot_col	Plot dendrogram of hierarchical clustering results with different colors
fold_change	Perform fold change analysis on the dataset.
fold_change_var	Fold change applied on two variables. Instead of having the difference of the variables on two groups, we have the difference of the groups on two variables.
heatmap_correlations	Plots a heatmap with the correlations.
hierarchical_clustering	Perform hierarchical clustering analysis on the dataset.
kmeans_clustering	Perform k-means clustering analysis on the dataset.
kmeans_plot	Plot for each formed cluster, in grey the values of all samples of that cluster and in blue the median of that samples.
kmeans_result_df	Show for each cluster from kmeans analysis the sample names belonging to them.
kruskalTest_dataset	Run Kruskal-Wallis Tests for each row of the data from the dataset.
ksTest_dataset	Run Kolmogorov-Smirnov Tests for each row of the data from the dataset.
linregression_onevar	Performs linear regression on one variable of the dataset.
linreg_all_vars	Performs linear regression analysis over the dataset with the selected metadata's variables.
linreg_coef_table	Gets a data.frame with the coefficient values.
linreg_pvalue_table	Gets the p-values table from the linear regression analysis.
linreg_rsquared	Gets the linear regression r-squared values.
multifactor_aov_all_vars	Perform multi-factor ANOVA on all variables with the selected metadata variables.
multifactor_aov_pvalues_table	Gets the p-values table from the multifactor ANOVA results.
multifactor_aov_varexp_table	Gets the variability explained table from the multifactor ANOVA results.
pca_analysis_dataset	Performs a classical PCA analysis over the dataset.
pca_biplot	Shows a PCA biplot.
pca_biplot3D	Shows a interactive 3D PCA biplot.
pca_importance	Gets the importance from the PCs.
pca_kmeans_plot2D	Groups the points with the clusters given by k-means in a 2D PCA scores plot.
pca_kmeans_plot3D	Groups the points with the clusters given by k-means in a interactive 3D PCA scores plot.
pca_pairs_kmeans_plot	Groups the points with the clusters from k-means in a PCA pairs plot.
pca_pairs_plot	Shows a PCA pairs plot.
pca_plot_3d	3D plot from 3 components
pca_robust	Performs a robust PCA analysis.
pca_scoresplot2D	Shows a 2D PCA scores plot of two principal componets.
pca_scoresplot3D	Shows a 3D PCA scores plot of three principal componets.
pca_scoresplot3D_rgl	Shows a interactive 3D PCA scores plot of three principal components.
pca_screepplot	PCA scree plot with the proportion and cumulative variance of the PCs.
plot_anova	Function for plotting the results from ANOVA. Usage
plot_fold_change	Function for plotting the results from fold change.
plot_kruskaltest	Function for plotting the results from Kruskal-Wallis tests.
plot_kstest	Function for plotting the results from Kolmogorov-Smirnov tests.
plot_regression_coefs_pvalues	Plots the linear regression coefficient and the p-values.
plot_ttests	Function for plotting the results from t-tests.
tTests_dataset	Run t-Tests for each row of the data from the dataset.

Table 5: *Specmine* package functions for univariate and unsupervised multivariate analysis. (Continued)

Function name	Description
volcano_plot_fc_tt	Volcano plot to intersect the results from t-tests and fold change.

3.2.4 Machine learning and feature selection

For machine learning, the package offers a number of functions to train, use and evaluate the different methods, covering both classification and regression approaches. It also includes validation methods to estimate the error metrics, including k-fold cross-validation, leave-one-out cross-validation and resampling methods, among others. The error metrics available include accuracy, [Area Under the ROC Curve \(AUC\)](#) and kappa statistic for classification, and [Root Mean Square Error \(RMSE\)](#) and the coefficient of determination (R^2) for regression. Functions to optimize model parameters (e.g. hidden nodes in a neural network) by testing and evaluating different values, according to the selected validation method and error metrics, are also available. This model optimization process gives the best model obtained and its performance, the variables' importance, the results of all tested combinations of parameters, the confusion matrices (in the case of classification), among other statistics. The resulting trained models can then be used for new data prediction.

Feature selection methods provided by *specmine* include both filter and wrapper methods, which can be combined with the machine learning methods. The package includes the most commonly used wrapper method [RFE](#), which tests different subsets of features, iteratively reducing the number of features and verifying which configuration provides the best performance. The various *specmine* functions for machine learning and feature selection are listed in [Table 6](#).

Table 6: *Specmine* package functions for machine learning and feature selection.

Function name	Description
feature_selection	Perform feature selection on the dataset.
filter_feature_selection	Perform selection by filter using univariate filters, from <i>caret</i> 's package.
flat_pattern_filter	Performs a flat pattern filter over the dataset.
multiClassSummary	Summary function for caret to compute AUC.
predict_samples	Predict new samples.
recursive_feature_elimination	Perform recursive feature elimination on the dataset using <i>caret</i> 's package.
summary_var_importance	Summary of variables importance of the models
train_and_predict	Train a model and predict new unlabeled samples with that model.
train_classifier	Train a specific classifier.
train_models_performance	Train various models.

3.3 PLATFORM ARCHITECTURE

The developed platform has modules which cover the main steps of a metabolomics data analysis workflow, of which the ones focusing on spectral data will be emphasized, as well as modules to handle public and private user's projects.

A graphical representation of the application's modules structure is shown in Figure 12, while the files structure is represented in

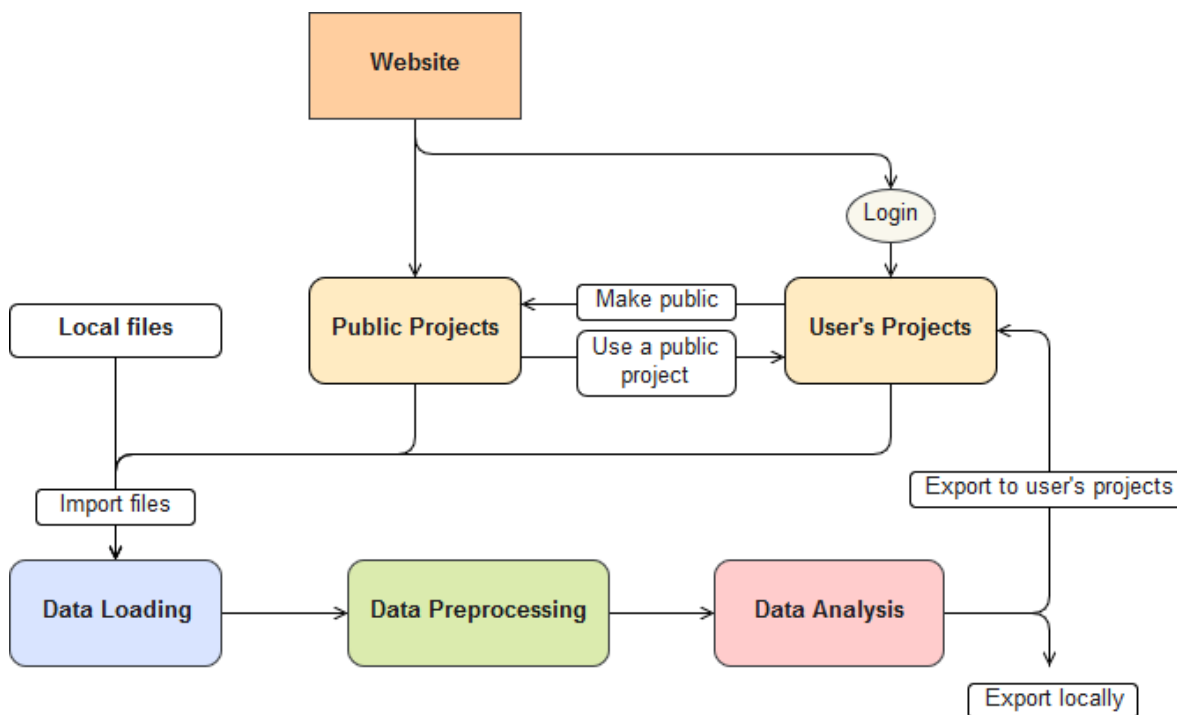


Figure 12: Graphical representation of the application's structure.

The application layout consists in a dashboard with three components: a header, the sidebar and the body, which was created using the R library *shinydashboard*.

The dashboard header gives access to the *Data Visualization*, *Pre-processing* and *Run Analysis* modules, as well as the *Saving* and *Loading Workspace* options. The header also includes the option to load a project, which is done through the *Choose Files* button if the user is logged in, or through the *New Project* button if the user is off-line. Lastly, the dashboard header also includes a button to handle the authentication of the user and his account options.

On the other hand, the dashboard sidebar includes three tabs: the *Home* tab, which represents the main page of the web application; the *My Projects* tab, which contains information about the user's stored projects, and the *Public Projects* tab where all user's shared projects

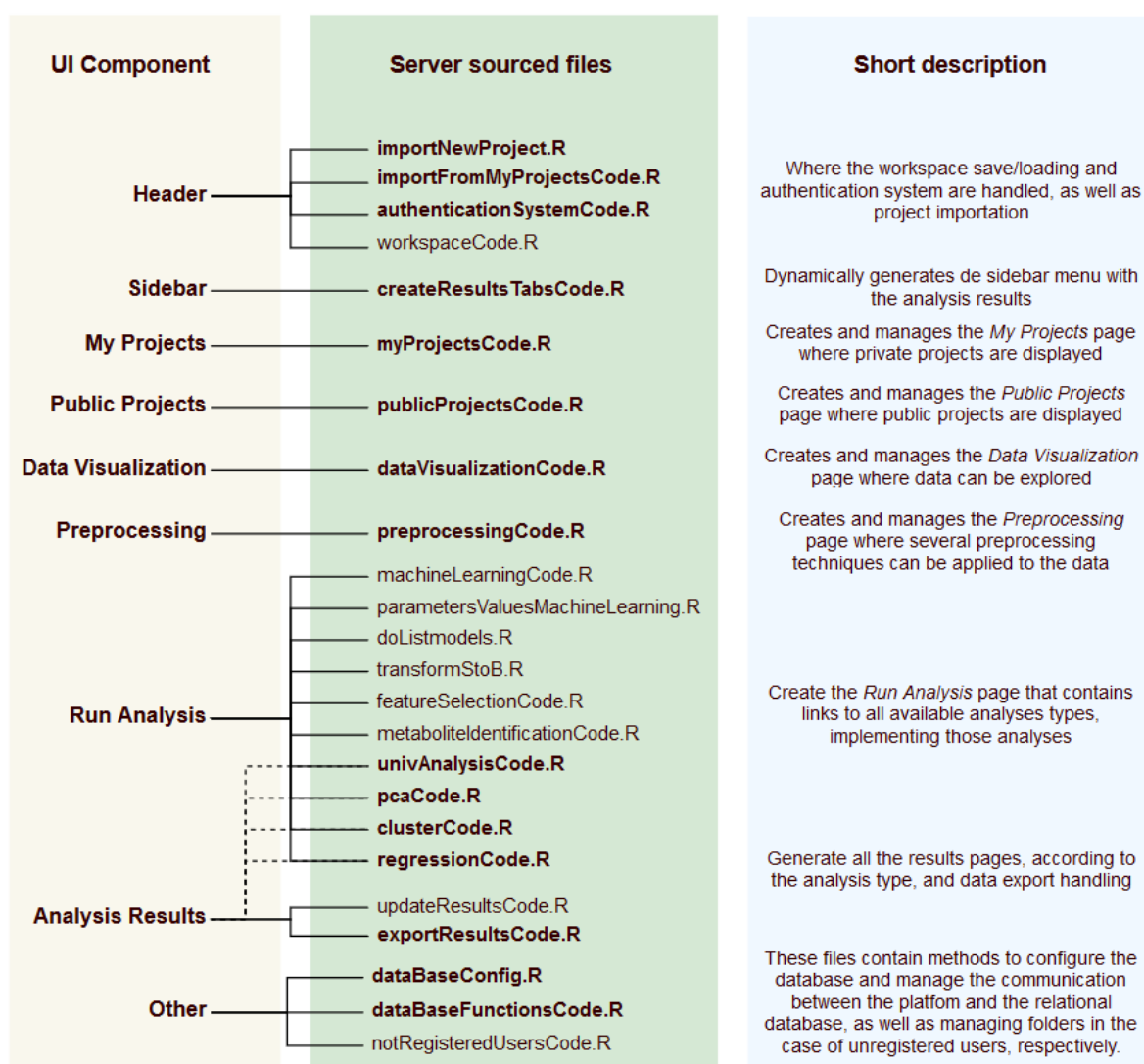


Figure 13: Graphical representation of the application's file structure. The filenames in bold represent the files to which I greatly or totally contributed to, given the scope of this work.

are shown. When a project is loaded the current dataset is also shown on the sidebar. The analysis results can be accessed in the *Analysis Results* menu.

The dashboard organization can be viewed in [Figure 14](#).

The different modules will be discussed in the following sections.

3.4 AUTHENTICATION SYSTEM

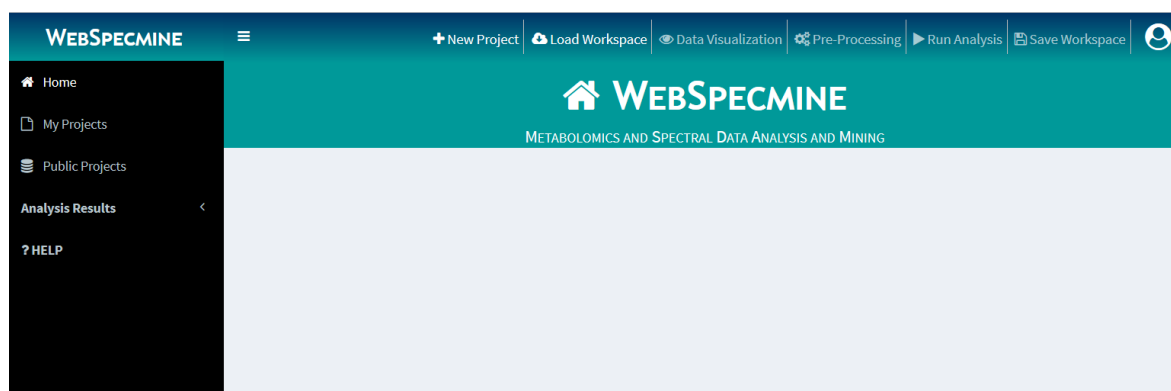


Figure 14: Main page of the web application.

In the web application, the authentication is made through the *user* button on the upper right corner. Here, the user can either login or register with his e-mail. The password encryption process is explained in [section 3.11](#).

Once logged-in, the user has the option to change his name and password, as well as deleting his account, in which case a warning about the account being permanently deleted is shown. The user also has the option to logout, which refreshes the page and the app is set to default values ([Figure 15](#)).

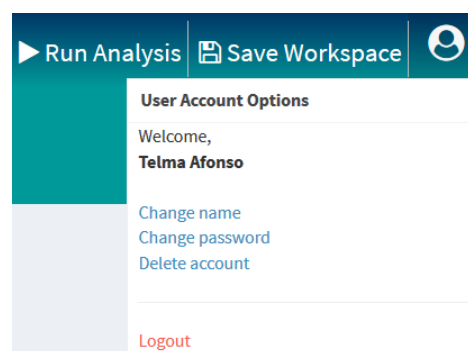


Figure 15: Authentication menu after successful login (detail).

3.5 PRIVATE AND PUBLIC PROJECTS

The modules that handle public and private user's projects are *Public Projects* and *My Projects*, respectively. In this web application, a project is associated with a data folder, which contains sub folders to store one or more datasets that can be of different data types; a metadata folder, which stores one or more metadata files; and a reports folder to store the reports generated during the analysis. A graphical representation of a project's structure is shown in [Figure 16](#).

To access the *My Projects* page, the user must be authenticated. Here, all the user's saved projects are displayed, including each project's description, datasets, metadata and report files.

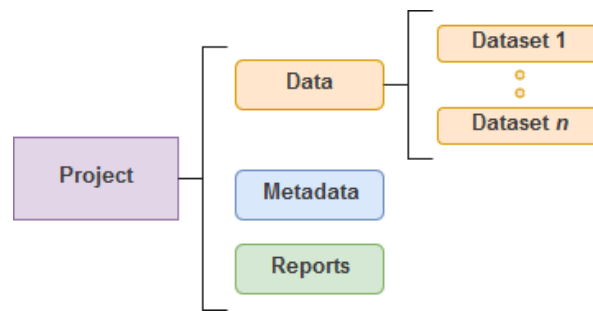


Figure 16: Graphical representation of a project's structure.

The user is able to create new projects and data folders for each project and also to edit their information, including the data type in the case of a data folder or name and description in both cases. Both projects and folders can be deleted.

For each data and metadata folder one or multiple files can be uploaded, which can be later used in an analysis. When creating a project the user can decide whether or not to make it public, in which case it will be available for every user. The project status can be changed at anytime using the circular button for the effect. All the user's stored files in this module can be viewed or downloaded at any time. A zoomed view of the *My Projects* page with the metadata tab selected is shown in Figure 17.

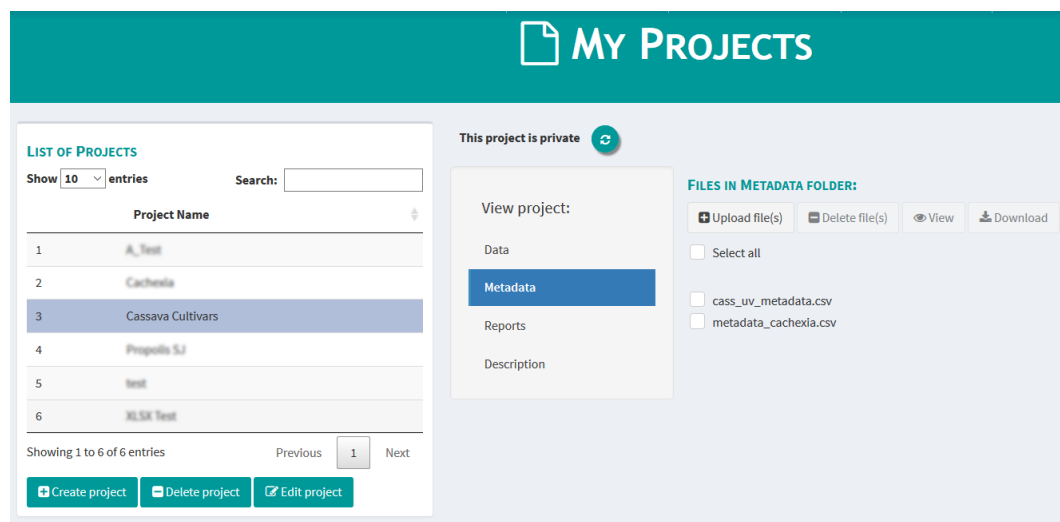


Figure 17: Zoomed view over *My Projects* page.

The *Public Projects* module can be accessed without any kind of authentication. Here, all projects that have been made public are displayed in table format, with information about the project name, author and data type. Any project can be imported into the user's private projects collection, given that the project itself is not owned by the user and he does not already own a project by that name. Each project in this module has a description, data,

metadata and reports files associated, as in the previous module, that can be viewed at any given time. To obtain the latest list of public projects a *refresh* button is provided.

An important feature also present in the web application is the ability to save and load the workspace. This way, all the data and results the user is currently working on can be saved into his account for later use, thus providing the ability to continue the analysis at any given time.

3.6 IMPORT FILES

Before any analysis can be made, the data must be loaded into the web application. This is done by clicking the *Choose Files* or *New Project* buttons on the dashboard header, which depends on whether or not the user is authenticated, respectively.

In the case of being authenticated, a window with three sections opens. These sections are related with the project, data folder and metadata file to be imported from the user's projects. After choosing the correct files a new window with options to create the dataset appears, according to the data type.

For spectral data, the data options consist in choosing the file type, that is, whether the selected folder has a single CSV file or multiple CSV, JDX, SPC or XLSX files, the field separator character, whether the samples are represented in columns or rows and if the file has row/column headers. Metadata options consist in choosing the field separator character and whether the file has row/column headers. Additionally, a short description of the data and the x and y axis labels may be provided.

On the other hand, if the user is not authenticated the files must be instead uploaded directly into the web application. However, the data and metadata options are the same as in the previous case. The *New Project* window for spectral data is shown in [Figure 18](#).

In both cases, the *specmine*'s functions used to implement the data reading process are described in [Table 3](#).

3.7 DATA VISUALIZATION

When a dataset is loaded, the data and some of its global statistics can be viewed in the *Data Visualization* page. Here, the data and metadata tables, the data summary, a boxplot of the variables and the spectra plot are shown.

Figure 18: Zoomed view over *New Project* window.

The *Data Summary* tab shows the summary of the loaded dataset, containing the description, type of data, number of samples, data points, metadata variables and missing values, x and y axis labels, mean, median and range of data values, standard deviation and quantiles of the dataset. The *specmine* function to retrieve the data summary is described in [Table 4](#).

In the *Data Table* tab, as the name indicates, a table with all data points is shown, with variables in the rows and samples in the columns. The *Metadata Table*, on the other hand, shows information regarding the metadata, with samples represented in rows and variables in columns. Both tables can be searched for a specific term and ordered by column.

When the loaded dataset is either of the type **NMR**, **MS**, **IR**, **UV-vis** or Raman spectra, an additional tab – *Spectra Plot* – is shown, containing the spectra plotted from the dataset, using *specmine*'s *plot_spectra* function. A number of options are available to adjust the plot, including the metadata variable to color the plot, the samples and the x axis range to plot, also including the option to reverse the x axis. [Figure 19](#) shows a zoomed view of the *Spectra Plot* tab in the *Data Visualization* page.

Lastly, the *Boxplot of the Variables* tab shows a boxplot that can have from one to the total number of variables plotted, which are selected using a *pickerinput* object from *shinyWidgets* library. The boxplot is plotted using a *specmine* function described in [Table 4](#).

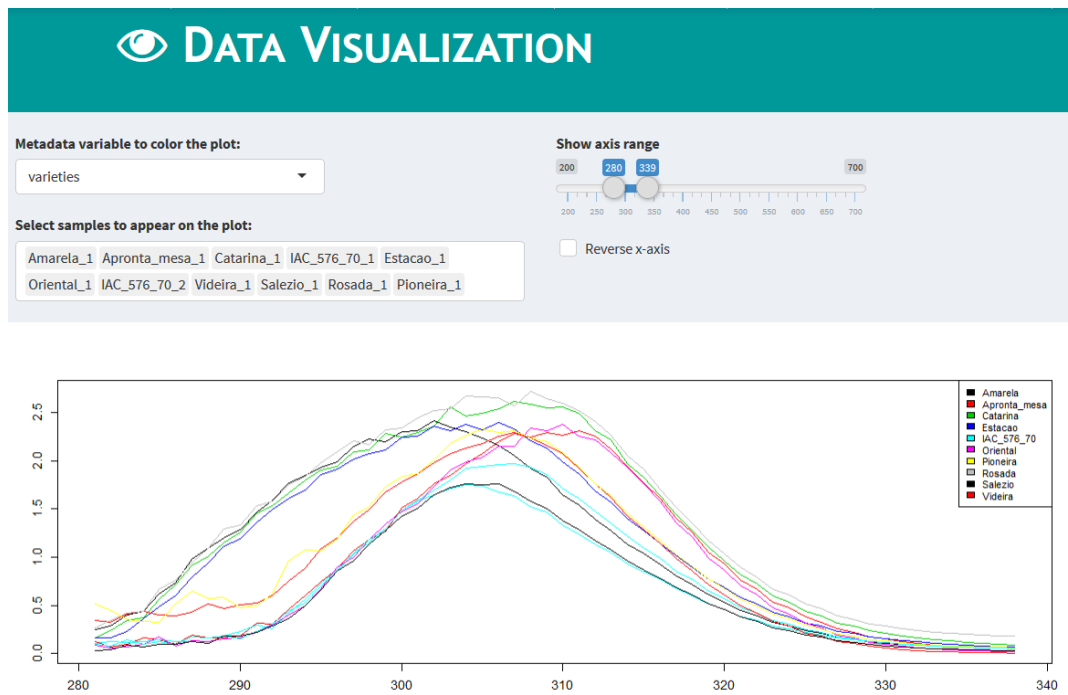


Figure 19: Zoomed view over *Spectra Plot* tab in the *Data Visualization* page.

An HTML report can be generated with the above information to be either downloaded or saved into the user reports folder of the selected project, in case he is authenticated.

3.8 PREPROCESSING

On the *Pre-Processing* page, a number of preprocessing approaches can be applied to the data. The page consists in a series of boxes to which a preprocessing technique is assigned to, displayed in two columns format. This way, various techniques can be applied sequentially with a simple mouse click. While some are straightforward to apply, others can be configured with different methods from which to choose from. Some techniques can only be applied to specific types of data or when a condition is met (e.g. missing values can only be treated if the dataset actually has some). After selecting the preprocessing techniques to apply to the data, the user must name the new dataset that is to be created, thus allowing to create multiple versions of the dataset.

A list with all the preprocessing approaches available in the web application and corresponding selectable methods (**M**), implemented using specmine functions described in Table 4, is presented below:

- **Data correction**
 - M Baseline; background; offset
- **Data normalization**
 - M Sum; median
- **Data transformation**
 - M Logarithmic; cubic root
- **Factor conversion**
- **First derivative**
- **Smoothing interpolation**
 - M Bin; loess; Savitzky-Golay
- **Flat patter filter**
 - M Interquartile range; relative standard deviation; standard deviation; median absolute deviation; mean; median
- **Mean centering**
- **Missing value handling**
 - M Mean; median; given value; KNN, linear approximation
- **Multiplicative Scatter Correction**
- **Scaling**
 - M Auto; pareto; range

In Figure 20, a zoomed view of the *Pre-Processing* page is shown, where some of the already mentioned methods, such as missing value handling, mean centering, data transformation, scaling and flat patter filter are emphasized.

Figure 20: Zoomed view over *Pre-Processing* page.

3.9 DATA ANALYSIS

Opening the *Run Analysis* page is the first step to perform data analysis using the web application. In this page, each analysis type (or group of analysis) is assigned to a panel with

the respective information and a button that leads into the corresponding analyses page, (Figure 21). Currently, there are available univariate analysis such as ANOVA, fold change analysis, T-Tests, Kruskal-Wallis and Kolmogorov-Smirnov tests. Unsupervised multivariate analysis include PCA, hierarchical and k-means clustering and correlation analysis. Other available supervised analyses are machine learning, regression analysis, feature selection and metabolite identification. Since the data types emphasized throughout this work (UV-vis, IR and Raman) are not usually employed in metabolite identification this type of analysis won't be here discussed.

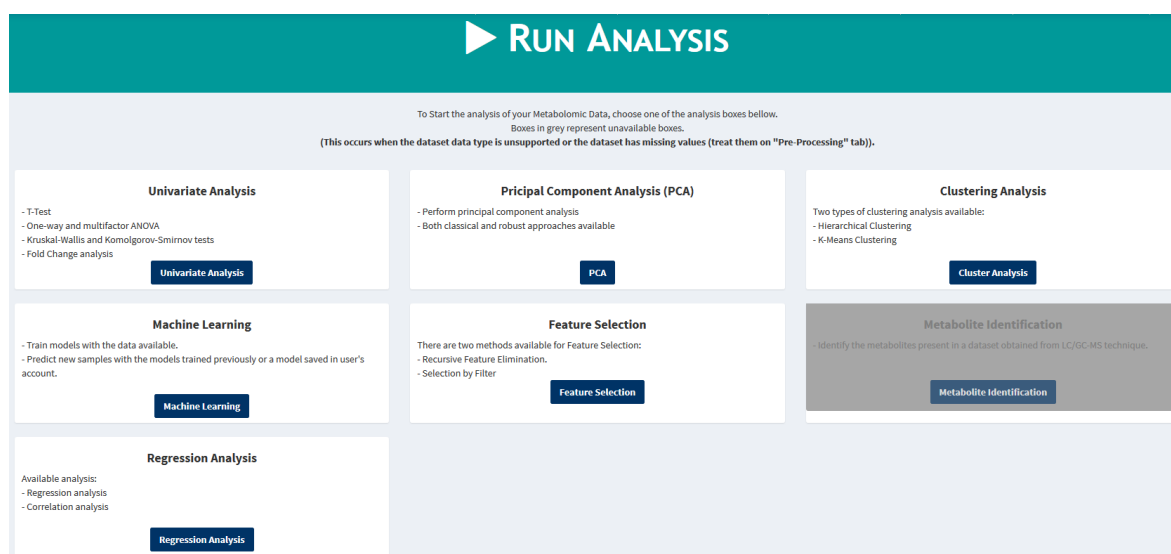


Figure 21: Zoomed view over the *Run Analysis* page.

To perform any type of the described methods, a name must be given to the analysis, this being the name that will appear under the corresponding analysis tab in the *Analysis Results* menu on the sidebar. Upon clicking an analysis name, the corresponding results page is opened. Every results page has a round button displayed on the top left corner, which can be clicked to reveal the options used to perform the analysis. It is also important to note that HTML reports and CSV files can be generated with the results at any time, which can then be downloaded and/or saved into the respective project's reports folder.

3.9.1 Univariate Data Analysis

Regarding univariate data analysis, the web application is able to perform either one-way or multi-factor ANOVA, T-Tests, Kruskal-Wallis and Kolmogorov-Smirnov tests, and fold change analysis. These analyses are implemented using *specmine*'s functions described in Table 5.

One-Way Analysis Of Variance

When performing a one-way **ANOVA** the user must select the metadata variable to use and whether the Tuckey's **HSD** test should be applied. Plot options include the p-value threshold and whether the x axis should be reversed. In the results page, a table with the p-value, logarithm of p-value, **FDR** and Tuckey's test results, if it was selected during the analysis, is shown. The table results are ordered by p-value, but can be also ordered by any other result type and searched for a specific term (Figure 22). For this type of analysis, a plot is also shown, with the negative base 10 logarithm of the p-value represented on the y axis and variables represented on the x axis.

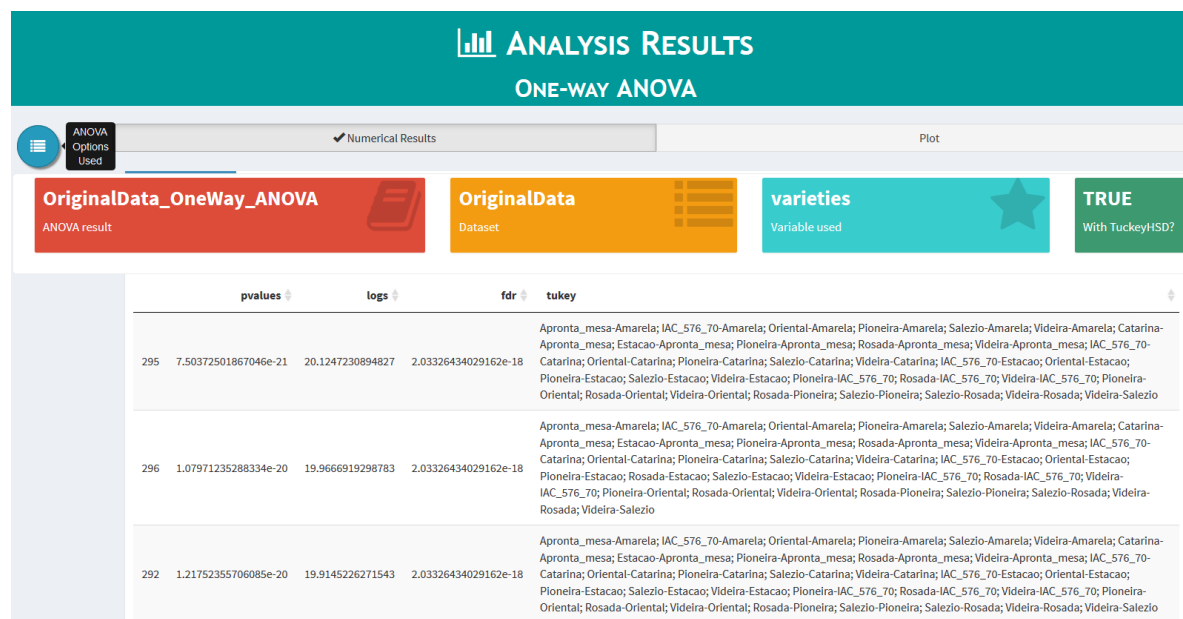


Figure 22: Zoomed view over *Analysis Results* page for one-way **ANOVA**, showing the numerical results tab and emphasizing the options used for the analysis.

Multi-factor Analysis Of Variance

For the multi-factor **ANOVA**, the metadata variables need to be selected and then a formula, using the selected variables, chosen. The results page for this type of analysis includes a table with the result for each variable on the data, with information regarding the degrees of freedom, sum of squares, mean square, F value, P-value and explained variability.

T-Tests, Kruskal-Wallis and Kolmogorov-Smirnov Tests

To run a T-Test, Kruskal-Wallis or Kolmogorov-Smirnov test for each variable from the dataset the user must start by choosing the metadata variable to create the groups of sam-

ples as well as the threshold value for the p-value to be considered significant. The results page for the three types of tests are similar, including a table with the p-values, $-\log_{10}$ of the p-values and the [False Discovery Rate](#), while also including a plot with variables in the x axis and the $-\log_{10}$ of the p-values in the y axis.

Fold Change Analysis

Two types of fold change analysis can be performed using the web application: either perform the analysis on the entire dataset or over two variables. In the latter case, instead of having the difference of the variables on two groups, the difference of the groups on two variables is calculated. In both cases, the metadata variable to use must be chosen.

The fold change analysis over the entire dataset requires the user to choose a reference value, namely a class of the metadata variable, while the analysis over two variables requires the user to choose the two variables to use. In the results page, a table with fold change values and the \log_2 of fold change is shown. It also includes a plot with these \log_2 of fold change values in the y axis and the variable names in the x axis, for the analysis over the entire dataset.

3.9.2 *Linear Regression Analysis*

To perform linear regression analysis, the metadata variables to use must be selected, as well as a formula specifying the model. The results page for this type of analysis includes tables with the p-values, coefficients, r-squared and adjusted r-squared values. It is also possible to plot the linear regression coefficient and the p-values for selected variables, with options to customize the color of the bars and font size.

3.9.3 *Unsupervised Multivariate Analysis*

Regarding unsupervised multivariate data analysis, the web application is able to perform either classical or robust [PCA](#), hierarchical and k-means clustering and correlation analysis using *specmine*'s functions described in [Table 5](#).

Principal Components Analysis

The simple form of [PCA](#) requires the user to decide if variables are to be scaled and/or centered, while the robust approach allows the centering and scaling methods to be chosen,

as well as the number of components. Centering can be done either by mean or median, while scaling methods include standard deviation ratio and mean absolute deviation.

The results pages for both approaches have three tabs: one with the numerical results, another to make the plots and finally a tab where the plots can be visualized. The numerical results tab includes tables with the component importance, the scores matrix and variable loadings for both approaches, while robust PCA results also include the order of the components.

Available plots are highly customizable, with options that range from selecting the variables to plot to more aesthetic options such as color palette selection (Figure 23A). The available plots in the web application are listed below:

- **Scree plot:** Shows the individual percentages of the explained variance of each principal component and cumulative;
- **Pairs plot:** Shows the pairs plot of the scores of the defined principal components, for a chosen variable (Figure 23B);
- **Scores plot:** Both 2D and 3D plots that show the scores of two different principal components;
- **Biplot:** Plot that displays samples as points, while the variables are displayed either as vectors, linear axes or nonlinear trajectories, considering PC1 and PC2 as axes;
- **K-means 2D and pairs plot:** Plots that combine some of the already mentioned plots with k-means results for coloring the points according to the cluster they belong.

Clustering Analysis

Both hierarchical and k-means clustering approaches are available. The former requires the user to select the distance measure (methods include Euclidean, Manhattan, Pearson correlation and Spearman correlation), the agglomeration method (complete, Ward, single, average, McQuitty, median and centroid methods available), and whether to perform the analysis over samples or variables. Additionally, a variable to color the leaves may be chosen. On the other hand, k-means clustering only requires the user to choose the number of clusters and whether to perform the analysis over samples or variables.

The results page for hierarchical clustering analysis includes the resulting dendrogram (Figure 24), as well as numerical results that comprise heights, order and the labels for the chosen variable to perform the analysis. The dendrogram is plotted using the respective *specmine* function described in Table 5, allowing the dendrogram to show colored leaves, according to the selected variable.

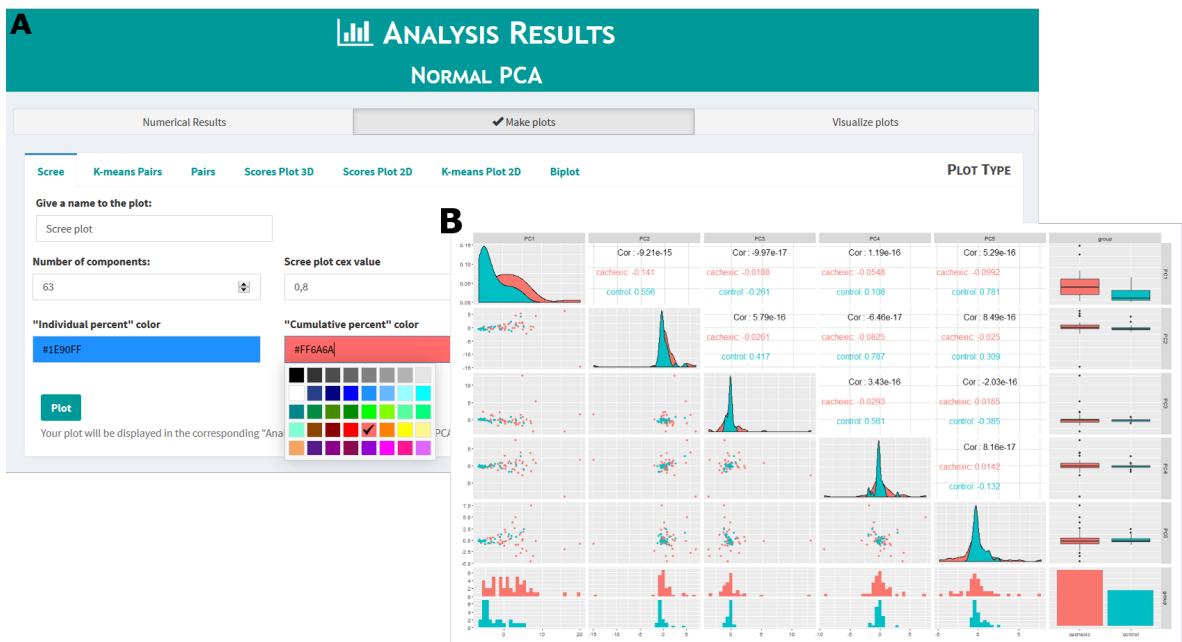


Figure 23: Zoomed view over the *Analysis Results* page for **PCA**, showing the *Make Plots* tab for the scree plot, emphasizing the customizable options (**A**) and example of a pairs plot made in the web application (**B**).

The k-means clustering results page shows information regarding each sample's cluster, the set of samples belonging to each cluster, the centers and the number of samples per cluster. For each cluster a plot is also available, showing in blue the median of the values of the samples in that cluster and in grey all the values of those samples. These plots are implemented using the functions described in [Table 5](#).

Correlation Analysis

To perform a correlation analysis between samples or variables the correlation method must be chosen. Three methods are available: Pearson, Kendall and Spearman. The color palette to use in the heatmap is also customizable, with a wide variety of colour gradients available to choose from.

Additionally, a correlations test can be performed over the entire dataset. In such a case, the alternative hypothesis to test must be chosen, and it can be two-sided, greater (for positive association) and less (for negative association).

The results page for this type of analysis includes the correlation matrix and, if a test was performed, the table with the correlation test results. A heatmap is also generated using the correlation matrix, with respective colour scale legend for easier interpretation (Figure 25).

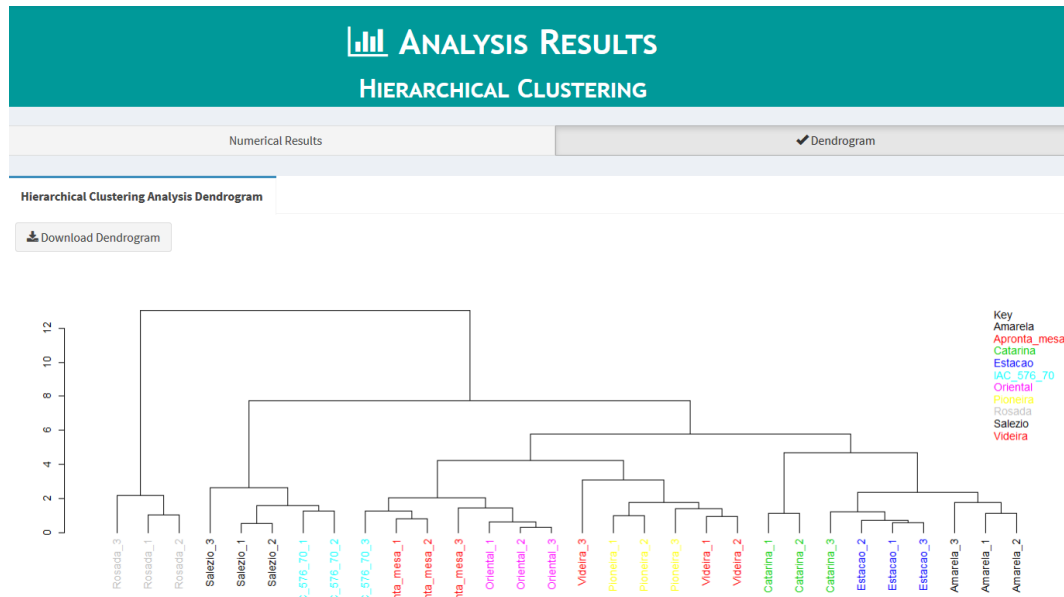


Figure 24: Zoomed view over *Analysis Results* page for hierarchical clustering, showing the clustering dendrogram.

3.9.4 Supervised Multivariate Analysis: Machine Learning

As mentioned before the platform development was shared and this module was not implemented by me, therefore this module will be briefly described.

Regarding machine learning analysis, the web application can perform both model training and prediction of new samples, by implementing *specmine* functions present in Table 6. Available models include PLS, LDA, decision trees (C4.5-like Trees), rule-based classifier (JRip method), SVMs with linear kernel, random forests and neural networks. Parameter optimization options are also included.

Model validation can be done using resampling, cross-validation, repeated cross-validation, leave-one-out cross-validation and leave group out cross-validation methods. The number of validation folds as well as the metric to test the models performance can also be chosen. These performance test metrics include accuracy and ROC curves.

3.9.5 Feature Selection

Similarly to the previous section, the feature selection module was not developed by me and, therefore, it will be briefly described.

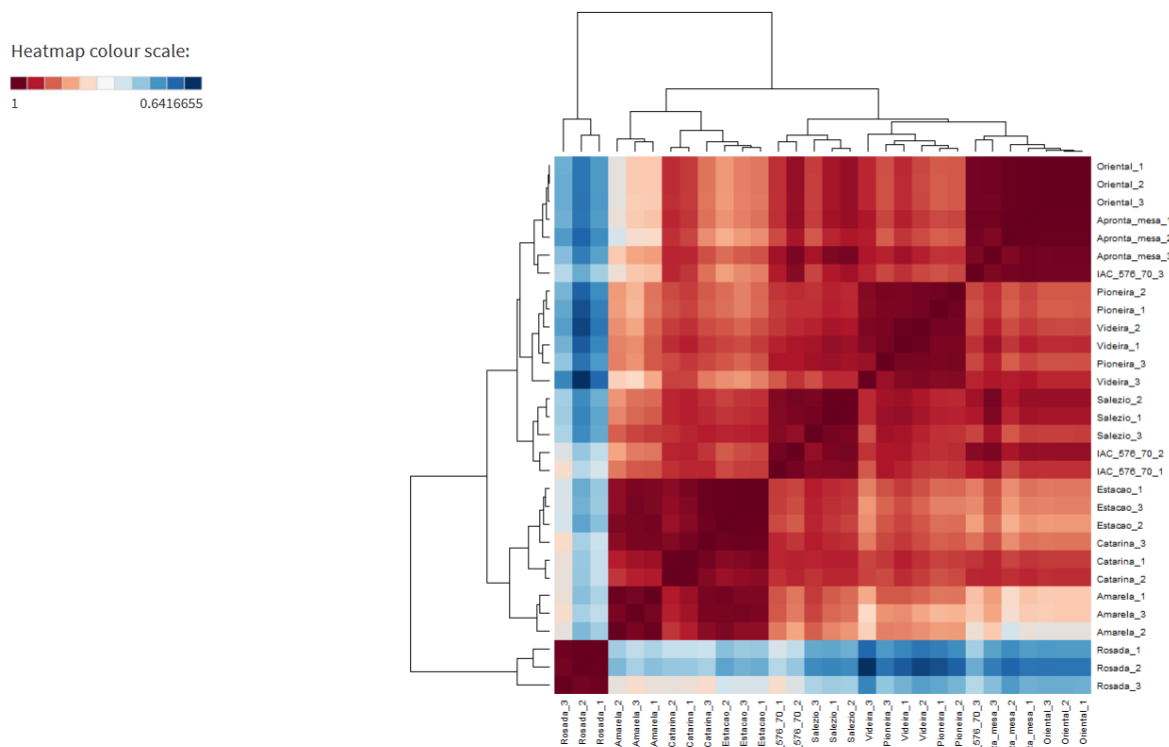


Figure 25: Zoomed view over the *Analysis Results* page for correlation analysis, emphasizing the correlation heatmap and respective colour scale.

Available feature selection methods include wrappers (RFE) and filters. Additionally, the metadata variable where the class to predict is must be chosen, as well as the function for model fitting, prediction and variable importance/filtering, which can be done using random forests, linear regression, bagged trees, LDA or the Naive-Bayes method. This type of analysis is implemented using *specmine*'s function described in Table 6.

3.10 DATA MODEL FOR PROJECT, DATASET AND USER MANAGEMENT

The data model used for the management of the different files and the authentication system will be explained in this and the following section, respectively.

A relational database for project, dataset and user management was created using the MySQL RDBMS. The tables that constitute the database are the *user*, *project*, *permission*, *dataset* and *datatype* tables. The MySQL model used in the platform for project, dataset and user management is shown in Figure 26.

The *user* table has a *user_id* attribute to store each user's unique identifier, a *user_name* attribute to store the user's first name, a *user_lastname* attribute to store the user's last

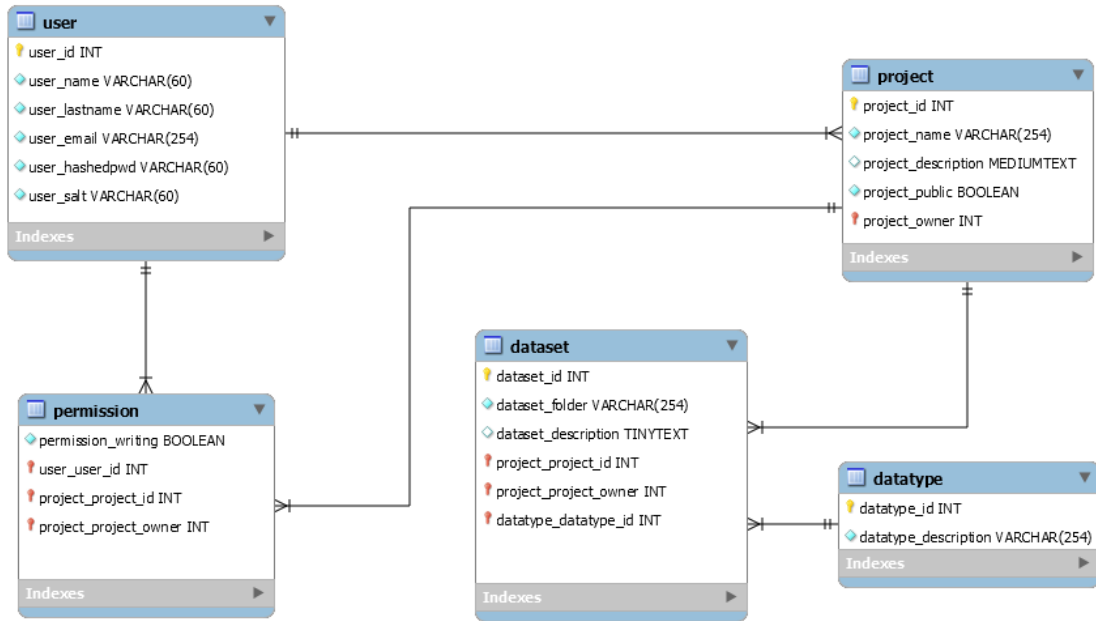


Figure 26: MySQL model used in the platform for project, dataset and user management.

name, a *user_email* attribute to store the user's email, a *user_hashedpwd* attribute to store the encrypted password and a *user_salt* attribute which stores the salt associated with the password. The password encryption process will be explained in the next section.

The *user* table has a 1:N relationship with the *project* table, meaning a user can have one or more projects. *Project* table has a *project_id* attribute to store each project's identifier, a *project_name* attribute to store the project's name, a *project_description* optional attribute to store the project's description, a *project_public* attribute that stores a boolean value indicating whether or not the project is public and a foreign key *project_owner* which references to the *user's* table *user_id* attribute.

Projects can either be public, and every user has access to them, or private, in which case only the creator has access. These permissions are handled in the *permission* table, which has a *permission_writing* attribute that stores a Boolean value according to the privacy of a project. Both *user* and *project* tables have a 1:N relationship with the *permission* table. The *user_user_id* foreign key references to the *user_id* attribute in the *user* table, whereas the *project_project_id* and *project_project_owner* reference to the *project_id* and *project_owner* attributes in the *project* table, respectively.

The dataset information is stored in the *dataset* table, which has a *dataset_id* attribute to store each dataset's unique ID, a *dataset_folder* attribute to store the dataset name and a *dataset_description* optional attribute to store the dataset description. Each dataset has an associated data type, which is stored in the *datatype* table. This table has a *datatype_id*

attribute that stores each data type unique ID and a *datatype_description* attribute to store the data type name.

Both *datatype* and *project* tables have a 1:N relationship with the *dataset* table. In this table, the *datatype_datatype_id* foreign key references to the *datatype_id* attribute in the *datatype* table, whereas the *project_project_id* and *project_project_owner* reference to the *project_id* and *project_owner* attributes in the *project* table, respectively.

While the management itself is done using the database, the files are stored in the local file system rather than the database, which only stores the paths to the files.

3.11 PASSWORD ENCRYPTION FOR AUTHENTICATION SYSTEM

For the password encryption process the *bcrypt* R package was used (<https://CRAN.R-project.org/package=bcrypt>). It consists in an R interface to the OpenBSD 'blowfish' password hashing algorithm, as described in [Provos and Mazieres \(1999\)](#).

Hashing is the transformation of a string of characters into a usually shorter fixed-length value or key that represents the original string. The hashing process is performed by a hash function, whose returned values are often called hash values, hash codes, digests, or simply hashes. This is a one-way function in which a hashed value cannot be reversed to obtain the original input value (i.e. the password). These functions generate random bytes or numbers from OpenSSL (<https://www.openssl.org/>). This provides a cryptographically secure alternative to R's default random number generator.

Bcrypt has the option to incorporate a salt, that is, a random data that is used as an additional input to the hash function, providing additional defense against dictionary attacks or against its hashed equivalent, a pre-computed rainbow table attack. *Bcrypt* is an adaptive function: over time, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power.

For each password chosen by the users a random salt is generated, using the *gensalt* function, which is stored in the database. The password string is then hashed with the generated salt using the *hashph* function, and the hash code stored in the database as well.

To authenticate a user, when the application receives a username and password, it performs the hashing operation using the password and stored salt and compares the resulting hashed value with the password hash stored in the database for the particular user. If the two hashes are an exact match, the user provided a valid username and password ([Figure 27](#)).

Below is a simple example of password hashing in R using the *bcrypt* package:

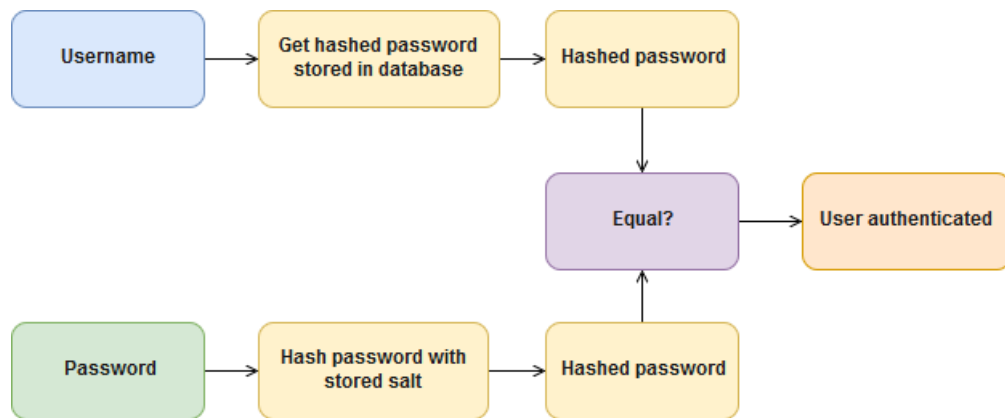


Figure 27: Graphical representation of the hashing process.

```

1 library(bcrypt)
3 password = '12345'
  salt = gensalt(log_rounds = 12)
5 salt
## "$2a$12$.RGbTpgZ8TyBpP.MJUFXMu"
7
  hash = hashpw(password, salt)
9 hash
## "$2a$12$.RGbTpgZ8TyBpP.MJUFXMubcbSTfHve1cnkHohULAIoDLWq58opNG"
11
  identical(hash, hashpw(password, salt))
13 ## TRUE

```

The prefix "\$2a\$12\$" in the hash string specifies a cost parameter of 12, indicating 2^{12} key expansion rounds. The random generated salt is ".RGbTpgZ8TyBpP.MJUFXMu" and the resulting hash for the "12345" password is ".RGbTpgZ8TyBpP.MJUFXMubcbSTfHve1cnkHoh-ULAIoDLWq58opNG". These are the values that would be stored in the database, rather than the password itself.

USE CASES

The main purpose of the current chapter is to demonstrate the functionalities of the web platform, by building analysis pipelines using real data from the literature. For this, two distinct datasets were used. The first use case is the discrimination of propolis samples from southern Brazil according to chemical profile (using UV-vis data) (Tomazzoli et al., 2015), while the second use case consists in the chemical and enzymatic composition screening in several genotypes of cassava roots during PPD (using IR data) (Uarrota et al., 2014).

4.1 PROPOLIS

4.1.1 Context

The propolis resinous substance is collected by honeybees *Apis mellifera* from various plant sources and added to salivary enzymes, beeswax, and pollen. They use it to seal openings and for protection against microorganisms and insects.

However, this substance also offers a broad spectrum of biological activities, including, for instance, cytotoxic, anti-herpes, free radical scavenging, antimicrobial, and anti-HIV activities, being used in both cosmetic and pharmaceutical markets. Therefore, the botanical origin of propolis is extremely important to guarantee raw materials of superior quality are supplied to said markets.

Since propolis quality depends, among other variables, on the local flora which is strongly influenced by (a)biotic factors over the seasons, the main scope of the study was to determine the harvest season effect on the chemical profile of this substance.

For this, propolis samples from *A. mellifera* were collected in Southern Brazil throughout 2014, and samples visually classified according to their color. The UV-vis absorbance values were recorded using a spectral window of 280-800 nm.

4.1.2 Data Overview

In Figure 28 the propolis dataset summary and metadata table is shown. The UV-vis dataset has no missing values, 5 metadata variables, a total of 165 samples and 521 data points. The last column of the metadata table (replicates) was omitted for forming reasons.

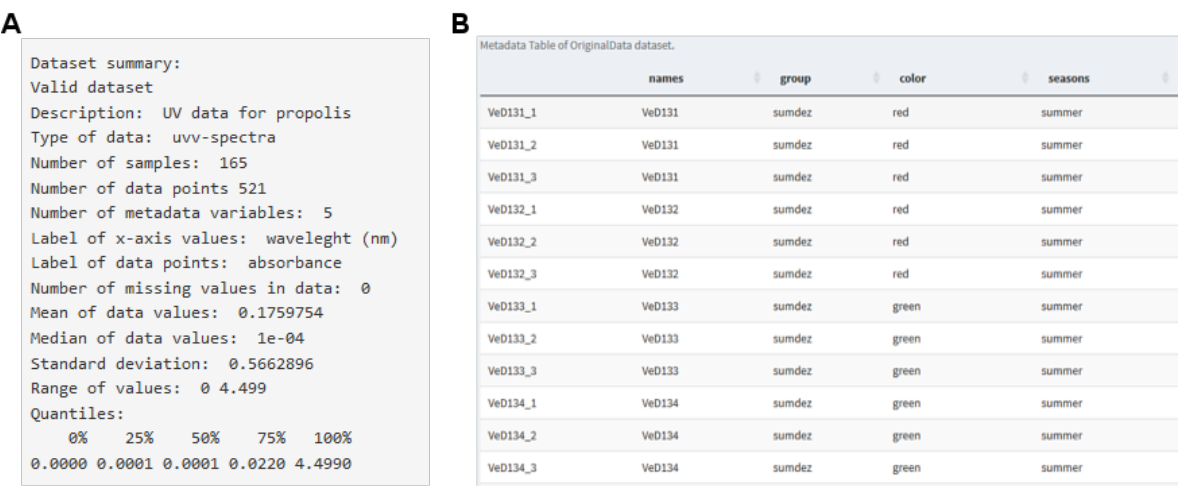


Figure 28: The dataset summary (A) and partial metadata table (B).

4.1.3 Preprocessing

Smoothing interpolation was initially performed over the propolis dataset, followed by background, offset and baseline corrections (Figure 29).

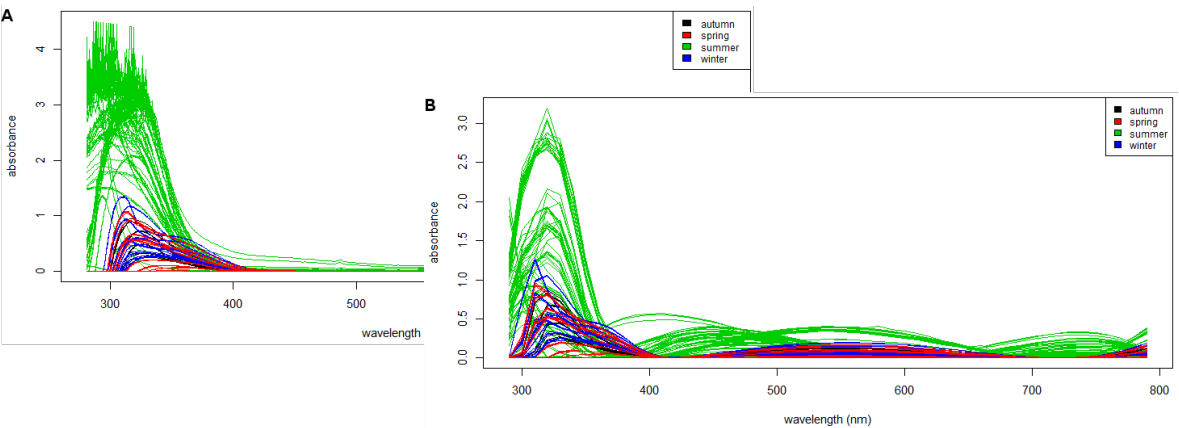


Figure 29: Propolis raw spectra (A) and spectra processed with smoothing interpolation, background, offset and baseline corrections (B), with samples colored according to season.

4.1.4 Univariate Analysis

4.1.5 Clustering

An **Hierarchical Clustering Analysis** with Euclidean distance measure and complete agglomeration method was performed over the propolis dataset. Samples were discriminated in two groups, the first one having samples collected in the four seasons and the second containing almost exclusively propolis samples produced in the summer.

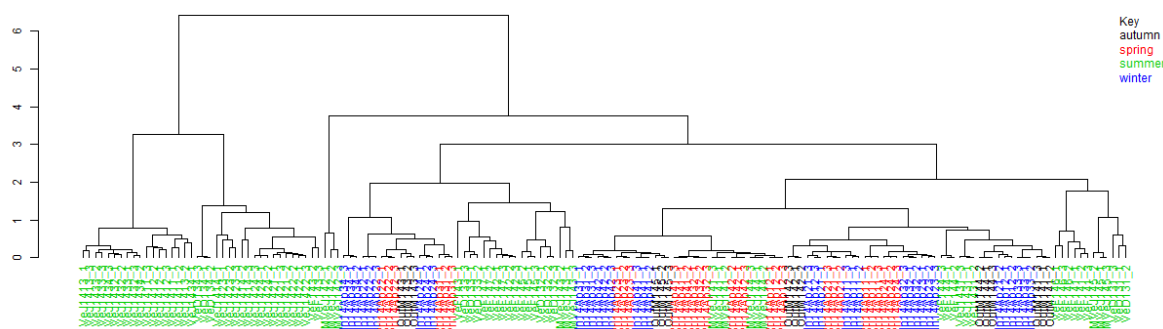


Figure 30: Hierarchical clustering analysis dendrogram of the propolis dataset. Euclidean distance and a complete agglomeration method were used.

4.1.6 Principal Components Analysis

A robust **PCA** was also performed, using data centered (by mean) and scaled (by standard deviation method) (Figure 31). With the first and second components explaining 49.9% and 25.0% of total variance, respectively, this results have confirmed the sample discrimination by seasons into two groups, as observed in the **HCA** (Figure 32).

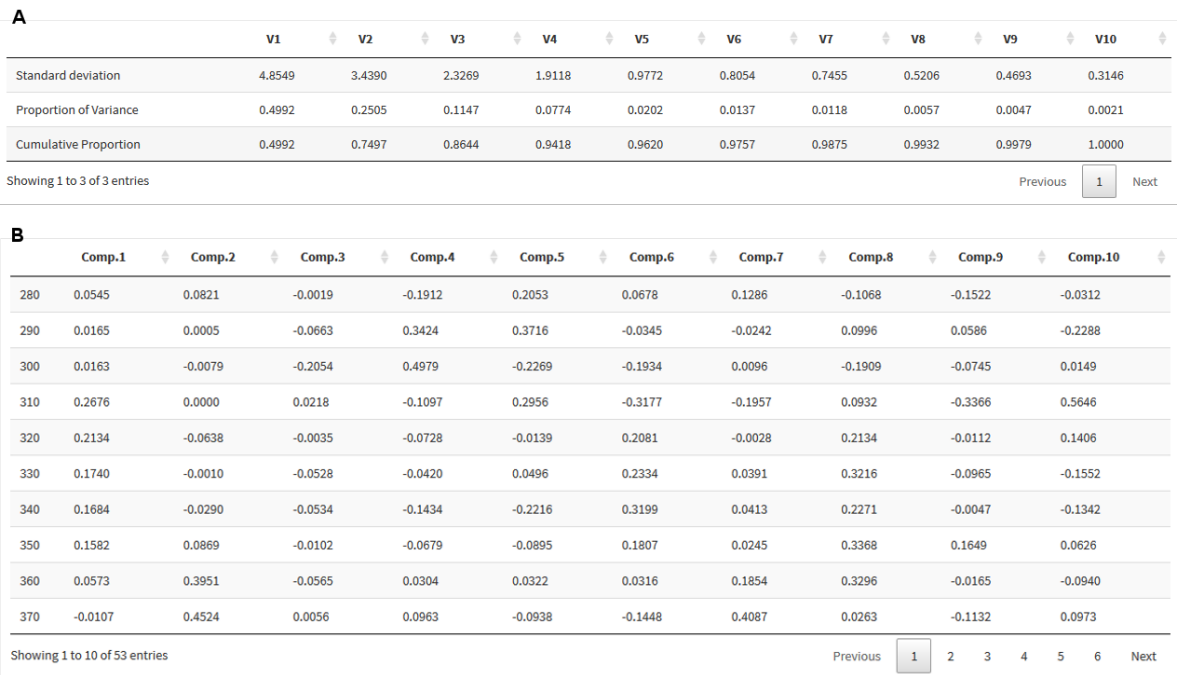


Figure 31: Principal components importance (A) and variable loadings (B) tables from the PCA performed over the propolis dataset.

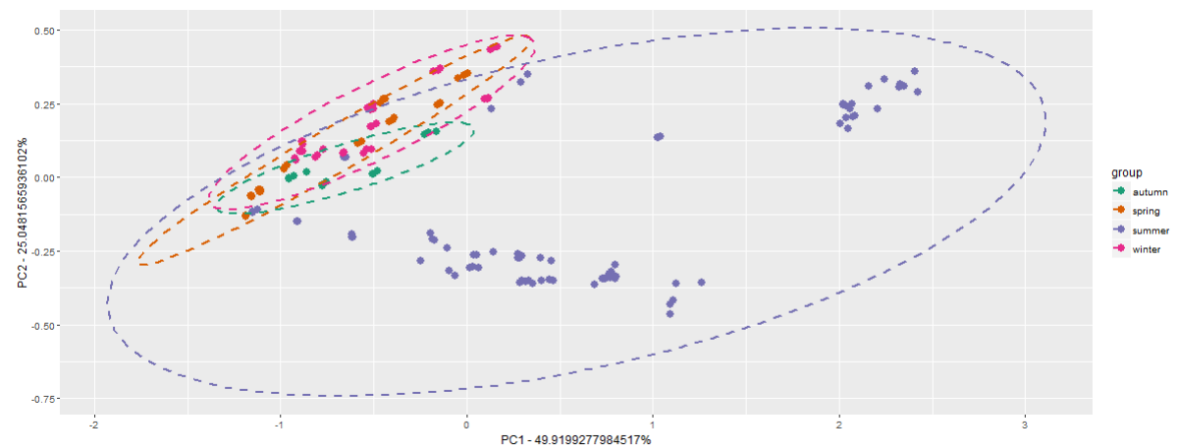


Figure 32: Scores scatter plot of the PCA performed over the propolis dataset for the first and second components.

CASE STUDY

The aim of this chapter is to present a case study using real data, providing a meaningful pipeline with the ability to demonstrate the application's features. This case study consists in the chemometric characterization of the carotenoid content in cassava roots tissue, using [UV-vis](#), CIELAB data and a [Low-Level Fusion](#) of the two.

5.1 INTRODUCTION

Cassava is the commonly used term to designate the *Manihot esculenta* species. This tuberous-root plant species offers a wide variety of agronomic advantages, being resilient to droughts, inexpensive, resistant to major diseases and pests, easy to grow and having flexible harvest times, allowing farmers to harvest the roots as needed. It is, therefore, a valuable source of energy for people living in the poorest regions. However, cassava roots are a poor source of provitamin A carotenoids, whose deficiency is a major problem in such regions ([La Frano et al., 2013](#); [Sánchez et al., 2014](#)).

Carotenoids are one of the most important natural pigments, having already been recognized benefits of carotenoid consumption such as the diminished risk of several degenerative disorders, including various types of cancer, cardiovascular or ophthalmological diseases, as well as their preventive effect associated with their antioxidant activity, protecting cells and tissues from oxidative damage ([Stahl and Sies, 2003](#)). However, only vitamin A precursors β -carotene, α -carotene and β -cryptoxanthin represent the major sources of carotenoids in the human diet.

With a broad range of colors, varying from yellow to dark-red, carotenoids confer color to many plant leaves, fruits and flowers, as well as birds, insects, fish, and crustaceans. The color of cassava's starchy root, which can vary from white to red, is strongly correlated to the presence and contents of several carotenoid pigments and their associations ([Sánchez et al., 2006](#)). However, the possibility of adopting the color of roots as an indirect criterion

for selection of higher carotene content is questionable, since color is a characteristic of difficult visual evaluation. Thus, the use of a standardized color measurement technique is of most importance.

The CIELAB color technique was adopted by the [Commission Internationale de L'Eclairage \(CIE\)](#) and is based on the Lab color space, which describes mathematically all perceivable colors in three dimensions: L^* for lightness and a^* and b^* for the color opponents green-red and blue-yellow. The values of these three variables are usually absolute, with the L^* value representing the darkest black at $L^* = 0$, and the brightest white at $L^* = 100$. On the other hand, the a^* value represents red and green opponents at positive and negative values, respectively, while the b^* value represents yellow and blue opponents at positive and negative values, respectively ([Brookes, 1982](#); [Schanda, 2007](#)). A visual representation of the CIELAB color space is shown in [Figure 33](#).

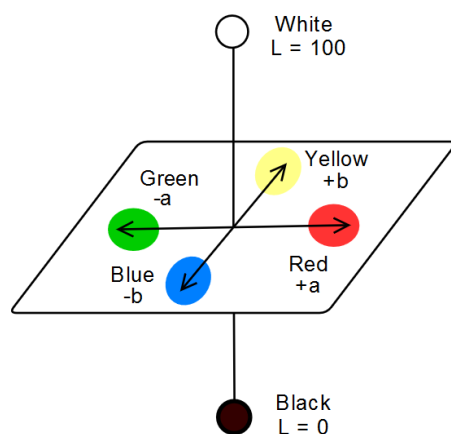


Figure 33: Representation of the CIE $L^* a^* b^*$ color space.

Currently, CIELAB is the most used system for quantitative color description of an object, given its uniformity, ease of acquisition, very low cost and device independence. Considering that this technique facilitates the acquisition of measurements directly on the field, while also avoiding the degradation of the compounds, it becomes an appealing approach in comparison to traditionally used methods such as [High Performance Liquid Chromatography \(HPLC\)](#) or [UV-vis](#). The CIELAB color technique has been applied for instance in the unique identification of skin color for clinical and scientific purposes ([Weatherall and Coombs, 1992](#)) and as an optimal color design approach for transforming patients' perception into color elements ([Liu et al., 2014](#)).

Combining [UV-vis](#) and CIELAB colorimetric data, the aim of the present case study is to validate a quantification method for carotenoid content estimation in roots of *M. esculenta*, assuming that the statistical and machine learning techniques can correlate these data types, to ultimately detect genotypes of *M. esculenta* with high contents of carotenoids.

Importantly, this study provides tools that can support the plant-breeding program at Epagri (Agricultural Research Company and Rural Extension of the State of Santa Catarina, <http://www.epagri.sc.gov.br/>) that aims to obtain genotypes with high levels of provitamin A carotenoids and superior nutritional traits.

5.2 DATA TYPES

5.2.1 Ultraviolet-visible Data

Fifty root samples of *M. esculenta* genotypes harvested in 2015/2016 season from the Epagri's germplasm bank were used due to their economic and social importance. Samples were visually selected for high content in carotenoids with provitamin A activity and lycopene.

After carotenoid extraction from the roots using organosolvents, extracts absorbances were then recorded using a spectral window from 200 to 700 nm. Aliquots of the extracts were also injected into a liquid chromatograph for carotenoid quantification purposes.

Considering that most carotenoids exhibit absorption in the visible region of the spectrum, between 400 to 500 nm, a subset of the original UV-visible dataset with samples belonging to this wavelength interval was used. Missing values contained within this dataset were replaced with the mean of the variables' values.

In order to accurately predict the carotenoid contents in cassava roots regression-derived statistical and machine learning models were used, including PLS, decision trees, SVMs, random forests and neural networks. Three response variables were used in the machine learning approach: the Total Carotenoid Content (TCC) determined by spectrophotometry (Lambert-Beer law), the TCC and the content of trans- β -carotene, the most abundant carotene in cassava roots, both determined by HPLC.

Models that showed best performance were selected and the variable importance calculated. Various pre-processing methods were applied to the datasets to see whether model performance could be improved, using these models. The data was also subject to filter-based feature selection (40%, 60% and 80% data filtering) to determine once again if it could improve model performance.

5.2.2 CIELAB Data

Immediately after harvest, color measurements of the root samples were made. For all fifty samples, three readings were performed at different sites.

The analysis pipeline was similar for the CIELAB dataset, however, models that included feature selection, the data pre-processing and filtering processes were excluded from the analysis pipeline, as it did not make sense to perform these, considering there are only 3 features in the dataset (L^* a^* and b^* parameters), while pre-processing was meant for spectral data.

5.2.3 Fusion Data

For the fusion dataset, which contained 104 variables (absorbance values + L^* a^* b^* parameters), the analysis pipeline was similar to that of the CIELAB dataset, while data filtering was also performed similarly to the [UV-vis](#) dataset.

5.3 UNIVARIATE ANALYSIS

To detect significant statistical differences (p-value below 0.05) derived from the effects of cassava's root colors on the spectral profiles a one-way [ANOVA](#) with Tukey's [HSD](#) analysis was performed for all wavelengths. [Figure 34A](#) shows the top ten results ordered by decreasing p-value. In [Figure 34B](#) the $-\log_{10}$ of p-values plot is represented, showing an horizontal line that corresponds to a p-value threshold of 0.05. More information about these methods is available in [subsection 2.2.2](#).

From [Figure 34](#) it appears that wavelengths around 500nm have a significant effect on the discrimination of cassava samples according to their color. This finding becomes more evident when looking at the $-\log_{10}$ of p-values plot.

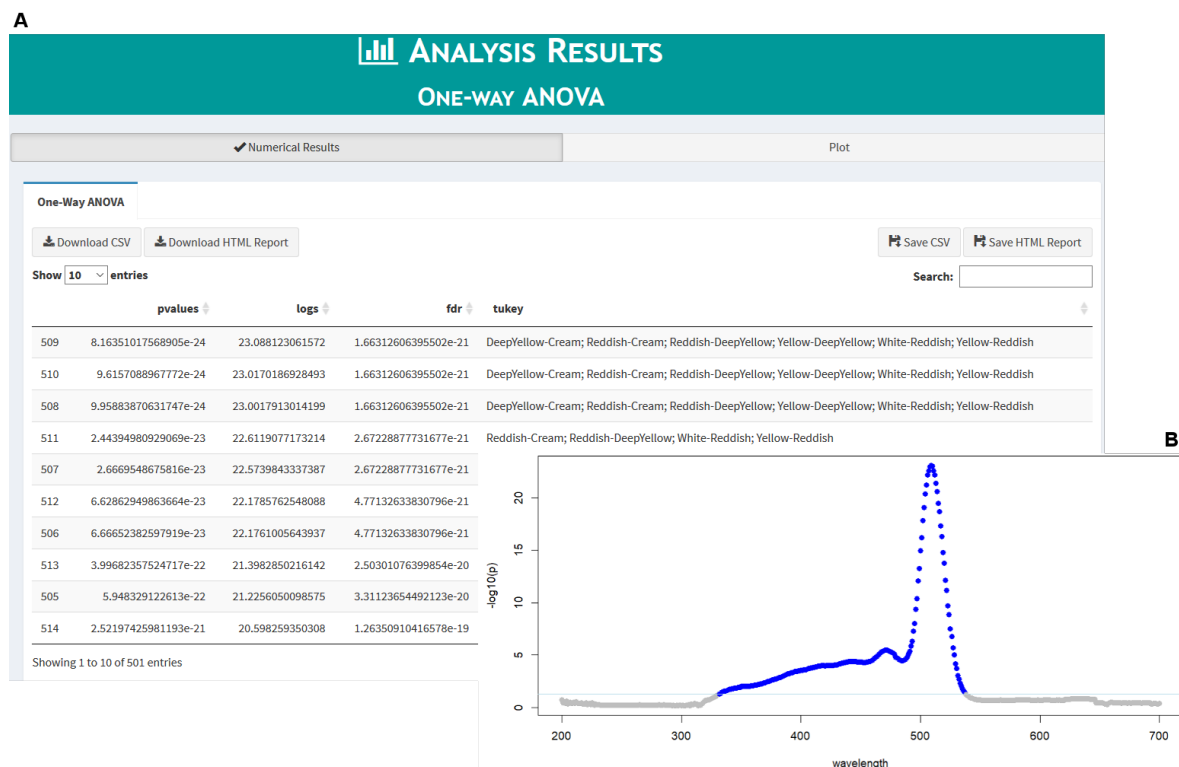


Figure 34: ANOVA results using the *colors* metadata variable (A) and respective plot of the $-\log_{10}$ of p-values with a p-value threshold value of 0.05 (B).

BIBLIOGRAPHY

- Barbosa-García, O., Ramos-Ortiz, G., Maldonado, J., Pichardo-Molina, J., Meneses-Nava, M., Landgrave, J., and Cervantes-Martínez, J. (2007). Uv-vis absorption spectroscopy and multivariate analysis as a method to discriminate tequila. *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, 66(1):129–134.
- Beleites, C. and Sergo, V. (2016). *hyperSpec: a package to handle hyperspectral data sets in R*. R package version 0.98-20161118.
- Brockes, A. (1982). The evaluation of whiteness. *CIE J*, (2).
- Cortez, P. (2016). *rminer: Data Mining Classification and Regression Methods*. R package version 1.4.2.
- Costa, C., Maraschin, M., and Rocha, M. (2016). An r package for the integrated analysis of metabolomics and spectral data. *Computer methods and programs in biomedicine*, 129:117–124.
- Cozzolino, D., Flood, L., Bellon, J., Gishen, M., and De Barros Lopes, M. (2006). Combining near infrared spectroscopy and multivariate analysis as a tool to differentiate different strains of *saccharomyces cerevisiae*: a metabolomic study. *Yeast*, 23(14-15):1089–1096.
- Darlington, R. B. and Hayes, A. F. (2016). *Regression analysis and linear models: Concepts, applications, and implementation*. Guilford Publications, New York.
- Daviss, B. (2005). Growing pains for metabolomics: the newest’omic science is producing results—and more data than researchers know what to do with. *The Scientist*, 19(8):25–29.
- De Lima, M. G., Moura, M. O., and Arízaga, G. G. C. (2011). Barcoding without dna? species identification using near infrared spectroscopy. *Zootaxa*, 2933:46–54.
- Fiehn, O. (2002). Metabolomics—the link between genotypes and phenotypes. *Plant molecular biology*, 48(1-2):155–171.
- Fourati, H. (2016). *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*, volume 1. CRC Press, Taylor & Francis Group LLC, Boca Raton, FL.
- Grissa, D., Pétéra, M., Brandolini, M., Napoli, A., Comte, B., and Pujos-Guillot, E. (2016). Feature selection methods for early predictive biomarker discovery using untargeted metabolomic data. *Frontiers in Molecular Biosciences*, 3.

- Hanson, B. A. (2016). *ChemoSpec: Exploratory Chemometrics for Spectroscopy*. R package version 4.3.34.
- Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*, volume 1. Springer-Verlag New York, 2 edition.
- Hu, S., Wang, J., Ji, E. H., Christison, T., Lopez, L., and Huang, Y. (2015). Targeted metabolomic analysis of head and neck cancer cells using high performance ion chromatography coupled with a q exactive hf mass spectrometer. *Analytical chemistry*, 87(12):6371–6379.
- Kansiz, M., Heraud, P., Wood, B., Burden, F., Beardall, J., and McNaughton, D. (1999). Fourier transform infrared microspectroscopy and chemometrics as a tool for the discrimination of cyanobacterial strains. *Phytochemistry*, 52(3):407–417.
- Khairudin, K., Sukiran, N. A., Goh, H.-H., Baharum, S. N., and Noor, N. M. (2014). Direct discrimination of different plant populations and study on temperature effects by fourier transform infrared spectroscopy. *Metabolomics*, 10(2):203–211.
- Kim, H.-Y. (2014). Analysis of variance (anova) comparing means of more than two groups. *Restorative dentistry & endodontics*, 39(1):74–77.
- Kind, T., Tolstikov, V., Fiehn, O., and Weiss, R. H. (2007). A comprehensive urinary metabolomic approach for identifying kidney cancer. *Analytical biochemistry*, 363(2):185–195.
- Kuhn, M., Wing, J., Weston, S., Williams, A., Keefer, C., Engelhardt, A., Cooper, T., Mayer, Z., Kenkel, B., Benesty, M., Lescarbeau, R., Ziem, A., Scrucca, L., Tang, Y., Candan, C., and Hunt, T. (2016). *caret: Classification and Regression Training*. R package version 6.0-73.
- Kumar, S., Panchariya, P., Prasad, B., and Sharma, A. (2013). Discrimination of indian tea varieties using uv-vis-nir spectrophotometer and pattern recognition techniques. 2(2):15–19.
- Kumar, V. and Minz, S. (2014). Feature selection: A literature review. *SmartCR*, 4(3):211–229.
- Kus, S., Marczenko, Z., and Obarski, N. (1996). Derivative uv-vis spectrophotometry in analytical chemistry. *Chern Anal*, 41:899–927.
- La Frano, M. R., Woodhouse, L. R., Burnett, D. J., and Burri, B. J. (2013). Biofortified cassava increases β -carotene and vitamin a concentrations in the tag-rich plasma layer of american women. *British Journal of Nutrition*, 110(2):310–320.
- Larsen, P. O. and Von Ins, M. (2010). The rate of growth in scientific publication and the decline in coverage provided by science citation index. *Scientometrics*, 84(3):575–603.

- Lee, K.-M., Herrman, T. J., Nansen, C., and Yun, U. (2013). Application of raman spectroscopy for qualitative and quantitative detection of fumonisins in ground maize samples. *Journal of Regulatory Science*, 1(1):1–14.
- Liland, K. H. (2011). Multivariate methods in metabolomics—from pre-processing to dimension reduction and statistical analysis. *TrAC Trends in Analytical Chemistry*, 30(6):827–841.
- Lin, M., Al-Holy, M., Chang, S.-S., Huang, Y., Cavinato, A. G., Kang, D.-H., and Rasco, B. A. (2005). Rapid discrimination of alicyclobacillus strains in apple juice by fourier transform infrared spectroscopy. *International journal of food microbiology*, 105(3):369–376.
- Liu, W., Ji, J., Chen, H., and Ye, C. (2014). Optimal color design of psychological counseling room by design of experiments and response surface methodology. *PloS one*, 9(3):e90646.
- MacFarland, T. W. and Yates, J. M. (2016). *Introduction to nonparametric statistics for the biological sciences using R*. Springer International Publishing AG, Switzerland.
- Martínez, A. M. and Kak, A. C. (2001). Pca versus lda. *IEEE transactions on pattern analysis and machine intelligence*, 23(2):228–233.
- Milburn, M. V. and Lawton, K. A. (2013). Application of metabolomics to diagnosis of insulin resistance. *Annual review of medicine*, 64:291–305.
- Owen, T. (1996). *Fundamentals of modern UV-Visible spectroscopy: A Primer*. Hewlett-Packard, Palo-Alto.
- Pereira, A. C., Reis, M. S., Saraiva, P. M., and Marques, J. C. (2011). Madeira wine ageing prediction based on different analytical techniques: Uv-vis, gc-ms, hplc-dad. *Chemometrics and Intelligent Laboratory Systems*, 105(1):43–55.
- Polshin, E., Aernouts, B., Saeys, W., Delvaux, F., Delvaux, F. R., Saison, D., Hertog, M., Nicolaï, B. M., and Lammertyn, J. (2011). Beer quality screening by ft-ir spectrometry: Impact of measurement strategies, data pre-processings and variable selection algorithms. *Journal of food engineering*, 106(3):188–198.
- Preisner, O., Lopes, J. A., Guiomar, R., Machado, J., and Menezes, J. C. (2007). Fourier transform infrared (ft-ir) spectroscopy in bacteriology: towards a reference method for bacteria discrimination. *Analytical and bioanalytical chemistry*, 387(5):1739–1748.
- Provos, N. and Mazieres, D. (1999). A future-adaptable password scheme. In *USENIX Annual Technical Conference, FREENIX Track*, pages 81–91.
- Roessner, U. and Bowne, J. (2009). What is metabolomics all about? *Biotechniques*, 46(5):363.

- Roggo, Y., Degardin, K., and Margot, P. (2010). Identification of pharmaceutical tablets by raman spectroscopy and chemometrics. *Talanta*, 81(3):988–995.
- Rosenblum, E., Viant, M., Braid, B., Moore, J., Friedman, C., and Tjeerdema, R. (2005). Characterizing the metabolic actions of natural stresses in the california red abalone, *haliotis rufescens* using 1h nmr metabolomics. *Metabolomics*, 1(2):199–209.
- Roussel, S., Bellon-Maurel, V., Roger, J.-M., and Grenier, P. (2003). 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(2):209–219.
- Sacré, P.-Y., Deconinck, E., Saelens, L., De Beer, T., Courselle, P., Vancauwenberghe, R., Chiap, P., Crommen, J., and De Beer, J. O. (2011). Detection of counterfeit viagra® by raman microspectroscopy imaging and multivariate analysis. *Journal of pharmaceutical and biomedical analysis*, 56(2):454–461.
- Sánchez, T., Ceballos, H., Dufour, D., Ortiz, D., Morante, N., Calle, F., Zum Felde, T., Dominguez, M., and Davrieux, F. (2014). Prediction of carotenoids, cyanide and dry matter contents in fresh cassava root using nirs and hunter color techniques. *Food chemistry*, 151:444–451.
- Sánchez, T., Chávez, A. L., Ceballos, H., Rodriguez-Amaya, D. B., Nestel, P., and Ishitani, M. (2006). Reduction or delay of post-harvest physiological deterioration in cassava roots with higher carotenoid content. *Journal of the Science of Food and Agriculture*, 86(4):634–639.
- Santos, P., Pereira-Filho, E., and Rodriguez-Saona, L. (2013). Rapid detection and quantification of milk adulteration using infrared microspectroscopy and chemometrics analysis. *Food chemistry*, 138(1):19–24.
- Savitzky, A. and Golay, M. J. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639.
- Schanda, J. (2007). *Colorimetry: understanding the CIE system*. John Wiley & Sons.
- Sikirzhytski, V., Sikirzhyskaya, A., and Lednev, I. K. (2012). Advanced statistical analysis of raman spectroscopic data for the identification of body fluid traces: semen and blood mixtures. *Forensic science international*, 222(1):259–265.
- Sikirzhytski, V., Virkler, K., and Lednev, I. K. (2010). Discriminant analysis of raman spectra for body fluid identification for forensic purposes. *Sensors*, 10(4):2869–2884.
- Smith, E. and Dent, G. (2005). *Modern Raman spectroscopy: a practical approach*. John Wiley & Sons, Chister, UK, 1 edition.

- Soderberg, T. (2010). Organic chemistry with a biological emphasis volume i. 1.
- Souto, U. T. C. P., Pontes, M. J. C., Silva, E. C., Galvão, R. K. H., Araújo, M. C. U., Sanches, F. A. C., Cunha, F. A. S., and Oliveira, M. S. R. (2010). Uv-vis spectrometric classification of coffees by spa-lda. *Food Chemistry*, 119(1):368–371.
- Stahl, W. and Sies, H. (2003). Antioxidant activity of carotenoids. *Molecular aspects of medicine*, 24(6):345–351.
- Stuart, B. H. (2004). *Infrared Spectroscopy: Fundamentals and Applications*, volume 8. Wiley Online Library, Chister, UK.
- Subari, N., Mohamad Saleh, J., Ali, Y., Shakaff, M., and Zakaria, A. (2012). A hybrid sensing approach for pure and adulterated honey classification. *Sensors (Basel, Switzerland)*, 12(10):14022–40.
- Thanasoulas, N. C., Parisis, N. A., and Evmiridis, N. P. (2003). Multivariate chemometrics for the forensic discrimination of blue ball-point pen inks based on their vis spectra. *Forensic science international*, 138(1):75–84.
- Thanasoulas, N. C., Piliouris, E. T., Kotti, M.-S. E., and Evmiridis, N. P. (2002). Application of multivariate chemometrics in forensic soil discrimination based on the uv-vis spectrum of the acid fraction of humus. *Forensic science international*, 130(2):73–82.
- Tomazzoli, M. M., Pai Neto, R. D., Moresco, R., Westphal, L., Zeggio, A. R., Specht, L., Costa, C., Rocha, M., and Maraschin, M. (2015). Discrimination of brazilian propolis according to the seasoning using chemometrics and machine learning based on uv-vis scanning data. *Journal of Integrative Bioinformatics (JIB)*, 12(4):15–26.
- Uarrota, V. G., Moresco, R., Coelho, B., da Costa Nunes, E., Peruch, L. A. M., de Oliveira Neubert, E., Rocha, M., and Maraschin, M. (2014). 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.
- Urbano, M., De Castro, M. D. L., Pérez, P. M., García-Olmo, J., and Gomez-Nieto, M. A. (2006). Ultraviolet-visible spectroscopy and pattern recognition methods for differentiation and classification of wines. *Food Chemistry*, 97(1):166–175.
- Varmuza, K. and Filzmoser, P. (2009). *Introduction to multivariate statistical analysis in chemometrics*. CRC press, Boca Raton, Florida, 1 edition.
- Virkler, K. and Lednev, I. K. (2010). Raman spectroscopic signature of blood and its potential application to forensic body fluid identification. *Analytical and bioanalytical chemistry*, 396(1):525–534.

- Weatherall, I. L. and Coombs, B. D. (1992). Skin color measurements in terms of cielab color space values. *Journal of investigative dermatology*, 99(4):468–473.
- Zhang, A., Sun, H., Wang, P., Han, Y., and Wang, X. (2012). Modern analytical techniques in metabolomics analysis. *Analyst*, 137(2):293–300.