UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE BIOLOGIA

ELISA RIBEIRO MIRANDA ANTUNES VEDOVATTI

Metabolomics and Machine Learning for Quality Control of Medicinal Plants

Metabolômica e Machine Learning para Controle de Qualidade de Plantas Medicinais

Campinas

2025

ELISA RIBEIRO MIRANDA ANTUNES VEDOVATTI

Metabolomics and Machine Learning for Quality Control of Medicinal Plants

Metabolômica e Machine Learning para Controle de Qualidade de Plantas Medicinais

Thesis presented to the Institute of Biology the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Plant Biology.

Tese apresentada ao Instituto de Biologia da Universidade Estadual de Campinas como parte dos requisitos exigidos para a obtenção do Título de Doutora em Biologia Vegetal.

Orientador:Profa. Dra. Alexandra Christine Helena Frankland Sawaya

Coorientador:Profa. Dra. Aurea Rossy Soriano Vargas

ESTE ARQUIVO DIGITAL CORRESPONDE À VERSÃO FINAL DA TESE DEFENDIDA PELA ALUNO ELISA RIBEIRO MIRANDA ANTUNES VEDOVATTI, E ORIENTADA PELA PROFA. DRA. ALEXANDRA CHRISTINE HELENA FRANKLAND SAWAYA.

Campinas

2025

Ficha catalográfica
Universidade Estadual de Campinas (UNICAMP)
Biblioteca do Instituto de Biologia
Mara Janaina de Oliveira - CRB 8/6972

Antunes, Elisa Ribeiro Miranda, 1993-
An89m Metabolomics and machine learning for quality control of medicinal plants / Elisa Ribeiro Miranda Antunes Vedovatti. – Campinas, SP : [s.n.], 2025.

Orientador: Alexandra Christine Helena Frankland Sawaya.
Coorientador: Aurea Rossy Soriano Vargas.
Tese (doutorado) – Universidade Estadual de Campinas (UNICAMP), Instituto de Biologia.

1. Plantas medicinais - Controle de qualidade. 2. Aprendizado de máquina. 3. Metabolômica. I. Sawaya, Alexandra Christine Helena Frankland, 1958-. II. Vargas, Aurea Rossy Soriano. III. Universidade Estadual de Campinas (UNICAMP). Instituto de Biologia. IV. Título.

Informações complementares

Título em outro idioma: Metabolômica e *machine learning* para controle de qualidade de plantas medicinais
Palavras-chave em inglês:
Medicinal plants - Quality control
Machine learning
Metabolomics
Área de concentração: Biologia Vegetal
Titulação: Doutora em Biologia Vegetal
Banca examinadora:
Alexandra Christine Helena Frankland Sawaya [Orientador]
Andréia de Melo Porcari
Ilara Gabriela Frasson Budzinski
Nilsa Sumie Yamashita Wadt
Sandra Eliza Fontes de Avila
Data de defesa: 17-03-2025
Programa de Pós-Graduação: Biologia Vegetal

Objetivos de Desenvolvimento Sustentável (ODS)
ODS: 12. Consumo e produção responsáveis

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0000-0001-9392-2677
- Currículo Lattes do autor: http://lattes.cnpq.br/5821110229054927

COMISSÃO EXAMINADORA

Profa. Dra. Alexandra Christine Helena Frankland Sawaya

Profa. Dra. Andréia de Melo Porcari

Profa. Dra. Ilara Gabriela Frasson Budzinski

Profa. Dra. Nilsa Sumie Yamashita Wadt

Profa. Dra. Sandra Eliza Fontes de Avila

A Ata da defesa com as respectivas assinaturas dos membros encontra-se no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa Biologia Vegetal da Unidade Instituto de Biologia

AGRADECIMENTOS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and by the program of Academic Excelency (PROEX) - Finance Code 88887.342848/2019-00.

"If I have seen further, it is by standing on the shoulders of giants."
— Isaac Newton

Scientia potentia est: Knowledge is power.

Resumo

Na última década, o interesse em plantas medicinais aumentou consideravelmente, tornando o controle de qualidade de fitoterápicos ainda mais importante. Embora a Farmacopeia Brasileira já sugira métodos para o controle de qualidade de alguns fitoterápicos listados, esses métodos não são isentos de limitações e geralmente se concentram em grupos de compostos ou compostos específicos. Dada a composição complexa das plantas medicinais e a o efeito sinérgico entre todos os seus compostos, uma metodologia mais holística seria ideal para esse controle de qualidade.

Nesse contexto, a Metabolômica *Untargeted* oferece uma alternativa interessante, uma vez que mede, simultaneamente, o máximo de metabólitos possível, sem focar em marcadores químicos específicos. No entanto, apesar de sua eficiência, a análise metabolômica geralmente gera grandes quantidades de dados altamente complexos que exigem ferramentas matemáticas, bioinformáticas e quimiométricas para processá-los e analisá-los.

Neste cenário, Machine Learning (ML) se destaca para dar suporte à análise e interpretação de dados metabolômicos, bem como para processar grandes conjuntos de dados. Portanto, o objetivo deste estudo foi desenvolver um método de controle de qualidade associando metabolômica e ML, focando nas espécies *M. ilicifolia* e *M. laevigata*, conhecidas popularmente como espinheira santa e guaco, respectivamente.

Para isso, 400 amostras de *Maytenus ilicifolia* e *Mikania. laevigata*, cada, foram analisadas por Cromatografia Líquida de Ultra Alta Performance acoplada à espectrometria de massa (UHPLC-MS) utilizando dois métodos analíticos distintos. Após a análise UHPLC-MS, os dados foram processados seguindo o fluxo de trabalho de estudos metabolômicos.

Com esses dados, dois modelos de ML foram desenvolvidos para cada espécie "oficial"(*M. ilicifolia* e *M. laevigata*) para classificar se novas amostras seguem os padrões de controle de qualidade e pertencem às especies-alvo. Para construir os modelos, três algoritmos foram testados: Support Vector Classification (SVC), K-Nearest Neighbours Classifier e Random Forest. Durante treinamento do modelo, a seleção de *features* foi realizada usando três métodos distintos: Mutual information,Recursive Feature Elimination e Boruta. *GridSearchCV* foi aplicado para otimização dos hiperparâmetros e os algoritmos foram avaliados usando o coeficiente de correlação de Matthews (MCC) e a métrica F1, tanto nas etapas de validação cruzada quanto de teste.

Os modelos finais obtidos apresentam valores de MCC de 94% e 97% para os modelos *M. ilicifolia* e *M.laevigata* respectivamente, comprovando o sucesso do método em identificar e diferenciar as espécies 'oficiais' de suas contrapartes. Embora o presente trabalho apresente limitações, a alternativa apresentada, que associa Metabolômica *Untargeted* com ML, oferece uma maneira eficiente, confiável e econômica de abordar os desafios do controle de qualidade de fitoterápicos.

Abstract

During the past decade, the interest in medicinal plants has increased significantly, enhancing the importance of quality control. Although the Brazilian Pharmacopoeia already suggests methods for the quality control of regulated medicinal herbs, these methods often focus on specific compounds or groups of compounds. Given the complex composition of medicinal plants and the synergistic effect between their compounds, a more holistic methodology would be ideal for quality control.

In this context, Untargeted Metabolomics offers an interesting alternative, as it simultaneously measures as many metabolites as possible, without focusing on specific chemical markers. However, despite its efficiency, metabolomic analysis usually generates large amounts of highly complex data that demand mathematical, bioinformatic, and chemometric tools to process and analyze them.

Herein lies the potential of Machine Learning (ML) to support analysis and interpretation of metabolomics data as well as to process large datasets. Therefore, the purpose of this study was to develop a quality control method that associates metabolomics and machine learning, focusing on two important brazilian medicinal species, *Maytenus ilicifolia* and *Mikania laevigata*, popularly known as "espinheira-santa" and "guaco", respectively.

To this end, 400 samples of *Maytenus ilicifolia* and *Mikania laevigata* each, were analyzed by Ultra High-Performance Liquid Chromatography coupled with mass spectrometry (UHPLC-MS) using two different analytical methods along with their counterparts. After UHPLC-MS analysis, the data were processed following the metabolomics workflow.

With such data, two machine learning models were developed for each 'official' species (*Maytenus ilicifolia* and *Mikania laevigata*) to classify if new samples follow quality control standards. To build the models, three algorithms were tested: Support Vector Classification (SVC), K-Nearest-Neighbors Classifier (KNN), and Random Forest (RF). In addition to model training, feature selection was performed using three distinct methods: Mutual information, Recursive Feature Elimination, and Boruta. The GridSearchCV was applied to find the optimal hiperparameter space and the algorithms were evaluated using Matthews correlation coefficient (MCC) and the F1 score, both at the cross-validation and testing steps.

The final models obtained present high MCC scores of 94% and 97% for the *M. ilicifolia* and *M. laevigata* models respectively, proving the success of the method in identifying and differentiating the 'official' species from their counterparts. Although the present work presents limitations, the alternative presented herein, which associates Untargeted Metabolomics with Machine Learning, offers an efficient, trustworthy, and economical way to approach the challenges of quality control of herbal medicine.

Contents

1 Introduction

1.1 Quality control of herbal medicines

In the past decade, the interest in medicinal plants has significantly increased and the global market for herbal drugs reflected this phenomenon. In 2019 the market was worth over US$ 83 billion partially driven by the coronavirus (COVID-19) pandemic, when there was an increased demand for immune-boosting herbal products (Ng et al., 2023). By 2030, this market is expected to reach US$ 550 billion, at a compound annual growth rate of 18.9% according to insightSLICE (insightSLICE, 2021; Ng et al., 2023).

Due to this increase, the quality assessment of medicinal plants is even more necessary as the lack of regulation of these products may result in adverse effects caused by poor quality, adulteration, and contamination of the herbal drug. Therefore, efficient quality control methods should be in place in every country where herbal medicines are used and regulated, which is the case for Brazil (World Health Organization, 2004).

In this regard, the Brazilian Pharmacopoeia already presents instructions for the quality control of regulated medicinal herbs with the use of methods such as Thin Layer Chromatography (TLC) and Total Phenolic, Flavonoid, and Tannin Contents (ANVISA, 2019). Such methods, however, present many limitations and usually focus on specific compounds or groups of compounds.

Thin Layer Chromatography, for example, is commonly used for an initial semi-quantitative evaluation (Liang et al., 2004) and even the most advanced techniques present low sensitivity (Loescher et al., 2014). Moreover, TLC methods also require a pre-purification step that can result in the loss of some constituents (Gilard et al., 2010).

Similarly, Total Phenolic, Flavonoid and Tannin Content are also methods with low sensitivity and specificity, as different species can present similar results by such techniques (Games, 2010). Additionally, the results from such methods can also be influenced depending on the presence of other interfering classes of compounds (Sánchez-Rangel et al., 2013).

Furthermore, medicinal plants present a complex composition, and their pharmacological activity is usually derived from a combined effect between many components (Rasoanaivo et al., 2011). According to Gilbert e Alves (2003), for some species, it is often observed that the isolated substances are less active than when presented in the mixture of the plant extract. In the psychotherapeutic field for example, isolated compounds from *Rauwolfia serpentina*, *Hypericum perforatum*, and *Passiflora incarnata* did not reproduce the activity observed from the crude extracts of the plants.

In addition to this, multiple studies demonstrated that the chemical composition of plants changes in response to environmental events (Flück, 1955; Qaderi et al., 2023). These changes are related to the plant acclimation and defense against such abiotic environmental stresses which can increase or decrease the amounts of primary and secondary metabolites within the

plants (Qaderi et al., 2023).

Anthocyanins, for instance, are produced at a higher rate under multiple stresses such as increased UVB exposure, heat stress, drought stress, nutrient deficiency, etc. (Wahid, 2007). Terpenoids, on the other hand, can increase during stress-induced stomatal closure, such as drought and heat stress (Singsaas, 2000). Glucosinolates, glycosides, and alkaloids have been shown to increase after cold stress while heat stress presented the opposite effect (Qaderi et al., 2023).

Therefore, since herbal medicine's activity is usually more efficient when considering the total composition of the extract, applying a quality control method that focuses on isolated compounds might not be ideal. Additionally, this composition might also change depending on the climate, and with that, the prevalence of given chemical markers.

Therefore, analyzing the whole composition and the combination of substances might be more accurate to ensure the quality and efficiency of herbal medicines. In this regard, quality control methods that apply a more holistic methodology are ideal and Untargeted Metabolomics is an interesting approach to achieve this goal (Lee et al., 2017).

1.2 Metabolomics

Metabolomics is an emerging 'omics' field that aims to identify and quantify small molecules known as metabolites in cells, tissues, or fluids of different species (Lee et al., 2017; Liebal et al., 2020). These small molecules are the by-products of metabolism, which, for the plant kingdom, can be divided into primary and secondary metabolism.

Primary metabolism is involved in a plant's fundamental processes such as growth, development, and reproduction and it is mostly formed by carbohydrates, lipids, and proteins. In consequence, primary metabolism is highly conserved among plant species. Secondary metabolism, on the other hand, is composed of a larger variety of metabolites, usually responsible for the plant's response to the environment. For this reason, different plant species present differences in their secondary metabolism, which is often used as a 'fingerprint' to identify the species (van Dam e van der Meijden, 2011).

According to Okada et al. (2010), in plant science, metabolomics is an effective approach to analyze the diversity of chemical compounds contained in plant cells. For medicinal plant research specifically, metabolomics has been widely used to evaluate and discriminate species and samples based on their metabolic profile, fingerprint, and even chemical markers (Okada et al., 2010).

Morad et al. (2023) for example, studied the effects and metabolic profile of *Calotropis procera* and A*triplex halimus*, two medicinal plants commonly found in different areas of Africa, Asia, and the Mediterranean regions. In this study, thanks to the application of metabolomics, 118 metabolites were identified that have been linked to multiple medicinal effects, such as anti-diabetic, antioxidant, and anticarcinogenic activity. As a result, these species can now be considered for future studies as a source of molecules for the treatment of medical conditions

related to such effects.

Another study, by Zanatta et al. (2023), leveraged mass spectrometry and nuclear magnetic resonance spectroscopy techniques for the metabolic analysis of *Terminalia catappa*, a Brazilian medicinal species. In the study, the authors were able to characterize the metabolic profile of this plant as well as understand how seasonal variation and environmental conditions interfere with its metabolic production. By using metabolomics, this study helped establish appropriate quality criteria for the standardization of this herbal medicine.

With similar approaches, previous studies from our research group also leveraged the power of metabolomics to aid the quality control of medicinal species. Mokochinski et al. (2018), for example, used metabolomics to analyze samples of Eucalyptus species and detected 88 polar primary metabolites and 625 semi-polar secondary metabolites. Additionally, the study was able to detect that soluble sugars and polyphenols were affected by different temperature regimes. This information will also help standardize the quality control for herbal medicines based on Eucalyptus samples.

Additionally, Galbiatti et al. (2021), also with the aid of metabolomics techniques, detected changes in the chemical composition of *Plectranthus neochilus (Coleus neochilus)* that coincided with environmental changes. Their study was the first attempt to correlate metabolic changes and environmental factors for this species and it was a first step toward a better quality control of this herbal medicine.

These studies used a specific type of metabolomic technique called untargeted metabolomics. Usually, metabolomics studies can be separated based on two distinct approaches: targeted and untargeted metabolomics. The targeted approach aims to measure a specific set of metabolites, typically focusing on one or more pathways of interest. Untargeted metabolomics, on the other hand, aims to simultaneously measure as many metabolites as possible, without focusing on specific compounds or groups of compounds (Patti et al., 2012).

In both approaches, the most popular analytical method is Mass Spectrometry (MS) due to its high sensitivity, selectivity, and throughput (Dunn e Ellis, 2005). For untargeted metabolomics, mass spectrometry is usually associated with Liquid Chromatography (LC-MS), as it allows the detection of thousands of peaks from biological samples, covering a wide range of compounds (Commisso et al., 2013; Kenny et al., 2005; Patti et al., 2012). For such reason, over the past decades, LC-MS has been applied in the analysis of herbal medicines (Liang et al., 2004) and it is ideal for the quality control of such products.

However, despite its efficiency, metabolomic analysis usually generates large amounts of highly complex data (Liebal et al., 2020). This data can contain hundreds or thousands of data points and, therefore, high-performance mathematical, bioinformatic, and chemometric tools are essential to process and analyze the data (Kenny et al., 2005; Alonso et al., 2015). In this regard, Machine Learning could offer great support to the Metabolomics field (Liebal et al., 2020).

1.3 Machine learning and its applications

Machine Learning (ML) is a subfield of Artificial Intelligence and, according to Géron (2019), can be defined as "the science and art of programming computers so they can learn from data". ML approaches learn from data to identify patterns and relationships and achieve a predictive analysis (Tian et al., 2018). The ML methods are commonly divided into two categories: supervised learning and unsupervised learning.

In supervised learning the goal is to create a regressor (for continuous outputs) or classifier (for discrete outputs) from a 'training' dataset and apply it to a 'test' set, to verify its performance (Zhang, 2020). Unsupervised learning, on the other hand, consists of exploring patterns in an unlabeled dataset using clustering approaches and/or multidimensional projections. The goal is to find the patterns on its own, in order to separate the data (Zhang, 2020).

The development of a supervised learning model typically begins with framing the problem and selecting an appropriate performance metric. This metric is chosen based on the model type (regressor or classifier), the problem being addressed, and the data configuration (balanced or imbalanced). Following these initial steps, data is gathered and split into training and testing sets.

This data split is performed to evaluate a model's performance. One possible way to evaluate a model involves deploying the model on real-world data and checking its predictions. However, if the model performs poorly, the results may be unreliable and potentially problematic. Therefore, an alternative approach involves dividing the initial dataset into training and testing sets.

The training set is used to train the model, while the testing set evaluates its performance by measuring the prediction error rate, known as the generalization error. This error provides insight into how the model might perform on unseen data.

During model training, various parameter values within the algorithm can be tested. However, training and testing the model multiple times with different parameter values can lead to the model 'memorizing' patterns specific to the train-test split, known as 'overfitting'. To solve this problem, a method called 'Cross-validation' can be used.

Cross-validation is performed by dividing the training set into complementary subsets, training the model on different combinations, and testing it on the remaining subsets. The best-performing model is then retrained on the full training set and tested on the test set to assess its generalization error. Figure 1 illustrates this development cycle.

Currently, ML is already being applied alongside metabolomics and chemometrics but it is mainly limited to the preliminary stages of the analysis. The most well-known and used algorithms, in this case, are hierarchical clustering analysis (HCA) and principal component analysis (PCA), both unsupervised learning methods (Nazarenko et al., 2019).

The applications of machine learning along with plant metabolomics, however, are still new. Few studies were developed on this subject and the application of ML for quality control of medicinal plants is even scarcer. The studies either focus on chemical markers or species

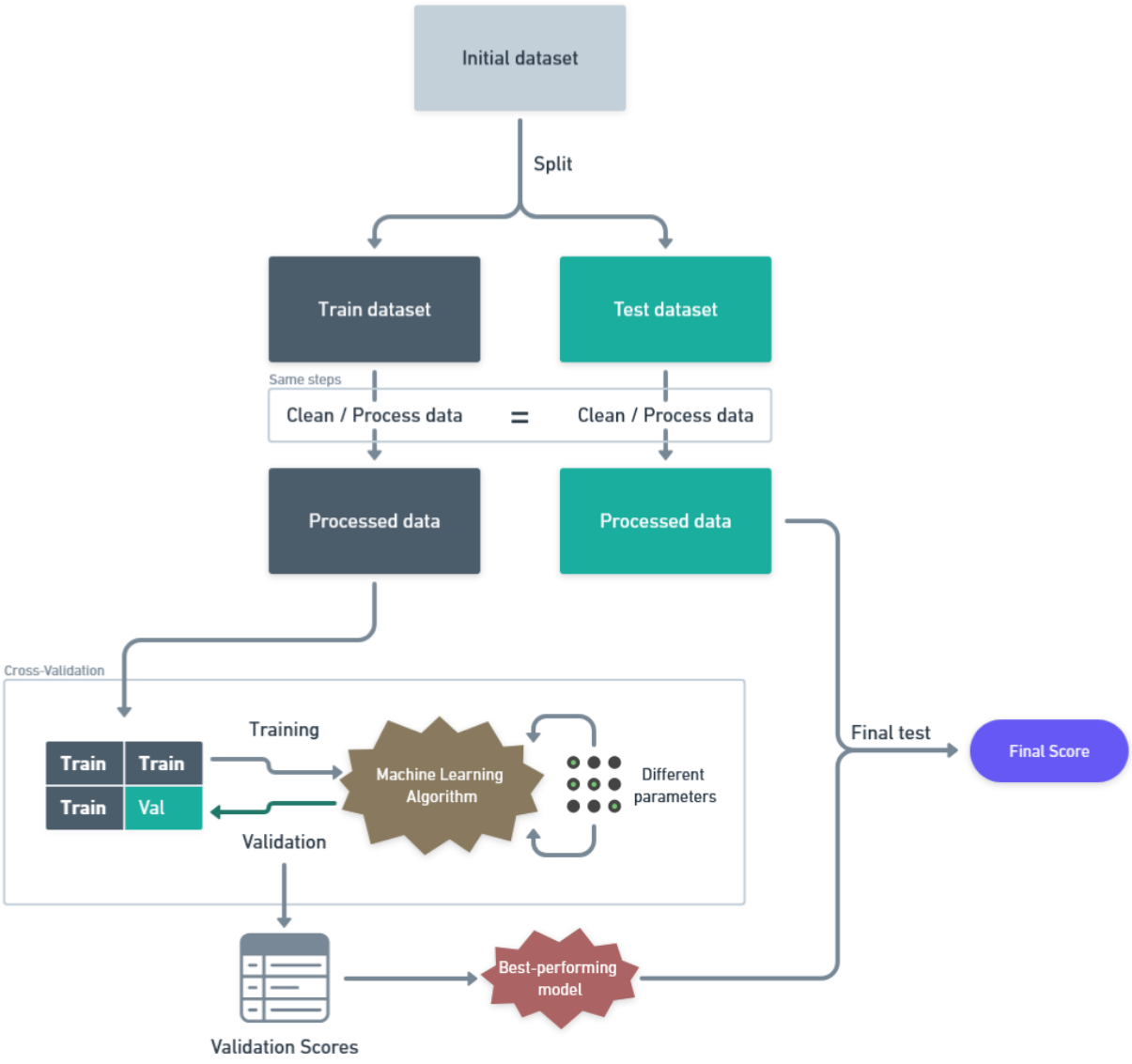

Figure 1: Machine Learning model development cycle

identification (Kharyuk et al., 2018; Li et al., 2020; Nazarenko et al., 2016) or do not apply LC-MS methods (Ramírez-Meraz et al., 2020).

Kharyuk et al. (2018), for example, applied LC-MS fingerprinting and ML to help identify 74 medicinal species. In their study, only one ML model and LC-MS methods were created to classify all samples. The drawback of their solution is that due to the diversity of metabolites in all 74 samples, creating one LC-MS method that detects all metabolites properly is a challenge.

Although the authors achieved a high accuracy score on the models, from 68% above, depending on the approach and algorithm, subtle differences between samples caused misclassification. The pairs *Bidens tripartita* - *Anethum graveolens* and *Aerva lanata* - *Salvia officinalis*, which were even from different families, demonstrated very similar chemical profiles on their LC-MS method, which resulted in poor classification scores by the machine learning model, that consistently mistook the samples in the pairs (up to 30% for some of the tested algorithms) (Kharyuk et al., 2018).

According to Srirama et al. (2017), there are herbal medicines adulteration practices, which only recently have been revealed by the use of 'omics' techniques such as metabolomics. In some African countries, for example, adulterated herbal drugs were up to 80% of the total market, due to misidentification or species substitution.

Therefore, effective quality control techniques must be developed, especially for cases of chemical and morphological similarities between two species. For this reason, a model like the one developed by Kharyuk et al. (2018) would not be enough for the quality control of medicinal plants, since it could present low scores for samples with similar compositions but different biological effects.

Another study by the same research group also applied Metabolomics and ML to identify plant species, but the drawbacks of this study are similar to the previous one. In this study, 36 species were analyzed via UHPLC-MS, and three algorithms were trained with the metabolomics data obtained: logistic regression (LR), support vector machine (SVM), and random forest (RF) (Nazarenko et al., 2016).

This study also achieved high accuracy scores of around 95% for most of the algorithms tested, but similarly to the previous study, only one model was trained to classify all 36 species. As the authors state, the next step of their work would be to test the applicability of their approach on a much larger number of species. However, as mentioned before, to achieve this goal, a universal UHPLC-MS method would need to be developed, capable of detecting and ionizing hundreds of metabolites present in a diverse set of samples, which is a challenging feat.

Therefore, even for those studies that applied LC-MS techniques along with ML classification algorithms, further research is necessary to verify their applicability to quality control processes, which demand techniques with higher sensitivity. Furthermore, at the moment, no attempt has been made to improve the quality control of Brazilian herbal medicines applying such methods.

1.4 Brazilian medicinal plants

In this regard, two important Brazilian medicinal species widely used by the population are *Maytenus ilicifolia* and *Mikania laevigata* popularly known as "espinheira-santa" and "guaco", respectively. Both species are included in the Brazilian Phytotherapy Formulary and clear instructions for their quality control are provided in the Brazilian Pharmacopoeia (ANVISA, 2019, 2021).

Maytenus ilicifolia Mart. ex Reiss, from the *Celastraceae* family, is popularly used due to its gastroprotective and antiulcerogenic effects (Ferreira et al., 2004; Gonzalez et al., 2001; Souza-Formigoni et al., 1991). The general population, however, occasionally substitutes this species with *Maytenus aquifolium* Mart., from the same family, possibly due to misidentification or intentional substitution (Santos-Oliveira et al., 2009).

Both plants, however, present some differences in their activity and composition. According to the Brazilian Pharmacopeia, "espinheira santa" samples should present at least 2.8 mg of epicatechin per gram of dried leaves. However, according to Duarte et al. (2022), *Maytenus aquifolium*, presented low amounts of epicatechin, which did not meet the quality control requirements and did not present significant amounts of catechin. *M. ilicifolia* samples, on the other hand, presented significant amounts of both catechin and epicatechin, which surpassed the minimum requirement of the Brazilian Pharmacopeia. Therefore, such differences point to the crucial need for the correct identification of the species (Antunes et al., 2019).

In terms of morphology, both species occur as large shrubs or trees, with glabrous leaves, and serrated margins with thorns that can vary in size. The main difference between *M. ilicifolia* and *M. aquifolium* is that the first usually reaches a maximum height of 5 meters, has monospermic fruits, and has larger marginal thorns on the leaves. In comparison, the second has a maximum height of 12 meters, presents dispermic fruits, and slightly smaller marginal thorns on the leaves (Reis, 2004). When the leaves are separated from the rest of the plant this difference is harder to detect, which could be one reason behind the substitution practiced by the general population, as shown in the Figure 2.

Mikania laevigata Sch. Bip ex Baker, and another similar species called *Mikania glomerata* Spreng., both from the Asteraceae family, are popularly used for the treatment of asthma, bronchitis, and cough (Borghi et al., 2023; de Lazzari Almeida et al., 2017b; Ueno e Sawaya, 2019). Both species present multiple morphological and anatomical similarities and therefore, are also frequently used without distinction by the population, as with the Maytenus species. (de Lazzari Almeida et al., 2017b; Ueno e Sawaya, 2019).

M. laevigata leaves are around 79.8 - 95.5 mm long, described as petiolate, oblong-lanceolate, glabrous, and coriaceous with entire margins and occasional lobes. In contrast, the leaves of *M. glomerata* are described as 29 - 69 mm long, oval lanceolate to deltoid with lobed margins, presenting four lobes that differentiate its leaves from those of its counterpart species. If cultivated in shaded environments, however, both species frequently present similar leaves and become indistinguishable. See Figure 3 below (Costa et al., 2018).

Additionally, both species also present differences in their composition, especially related

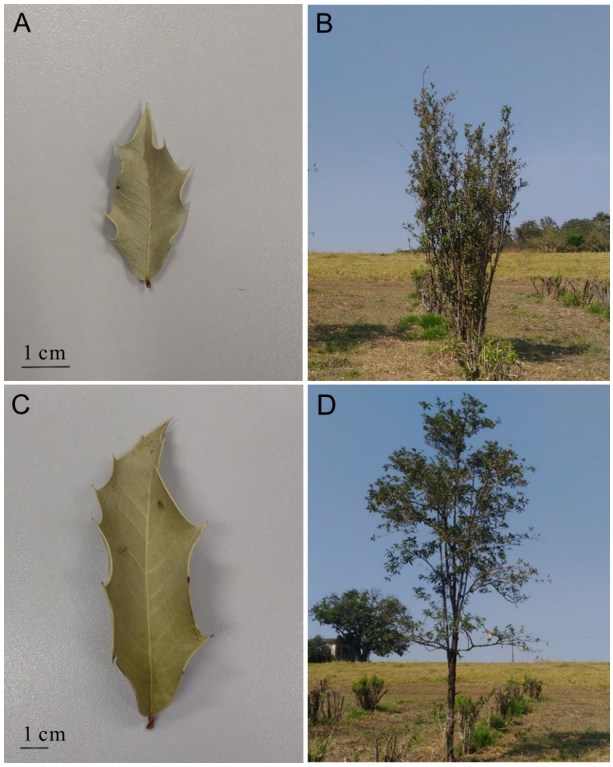

Figure 2: A-B: *M. ilicifolia* leaf and whole plant, respectively. C-D: *M. aquifolium* leaf and whole plant, respectively.

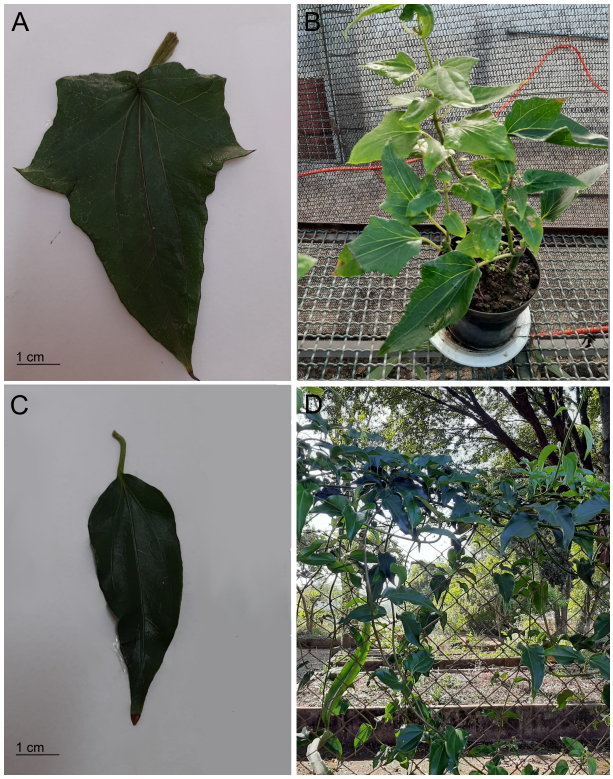

Figure 3: A-B: *M. glomerata* leaf and whole plant, respectively. C-D: *M. laevigata* leaf and whole plant, respectively.

to the presence of coumarin, the chemical marker of 'guaco'. According to Melo e Sawaya (2015), *M. laevigata* presents significant amounts of coumarin, while *M. glomerata* produces a higher level of caffeoylquinic acids and practically no coumarin. Therefore, in a similar case to that of the Maytenus species, both *M. laevigata* and *M. glomerata* need to be efficiently identified and clear quality control methods need to be applied to differentiate samples of both species.

Given the importance of "espinheira-santa" and "guaco" in Brazil's primary healthcare, both have been included in Brazil's reference list of medicine provided to the general population (da Saúde, 2009). This reinforces the critical need and increased interest to ensure the quality of herbal medicines derived from these species. Additionally, as substitutes for both *M. ilicifolia* and *M. laevigata* are often used interchangeably, the need for efficient and sensitive quality control methods is further emphasized.

Therefore, using both species as models for a proof-of-concept, this study aimed to develop a quality control method that associates metabolomics and machine learning. For this, *M. ilicifolia* and *M. laevigata* were used as target species while their respective counterparts, *M. aquifolium* and *M. glomerata*, were part of the negative class, to ensure the method would be able to differentiate similar samples.

All samples were analyzed via Ultra High-Performance Liquid Chromatography coupled with Mass Spectrometry (UPLC-MS). The resulting data, after properly processed by Metabolomics methods, served as a training set for two Machine Learning algorithms, one for each 'official' species.

The final models obtained present high MCC scores of 94% and 97%, respectively, for the *M. ilicifolia* and *M. laevigata* models, proving the success of the method in identifying and differentiating the 'official' species from their counterparts. Even though the present work has limitations, the alternative presented herein, which associates Untargeted Metabolomics with Machine Learning, offers an economical and more modern way to approach the challenges of quality control of herbal medicines.

1.5 Objectives

The primary objective of this study is to develop a robust quality control method for Brazilian medicinal plants by integrating untargeted metabolomics with machine learning. Specifically, the study aims to ensure the accurate identification and differentiation of *Maytenus ilicifolia* and *Mikania laevigata* from their commonly used counterparts, *Maytenus aquifolium* and *Mikania glomerata*, respectively. This approach seeks to address the limitations of current quality control methods that focus on isolated compounds and are often insensitive to the complex chemical compositions of medicinal plants.

1.6 Research Question

The main research question that this work aims to answer is: How can untargeted metabolomics be combined with machine learning to improve the quality control of Brazilian

medicinal plants, specifically *Maytenus ilicifolia* and *Mikania laevigata*, and accurately distinguish them from their similar species, *Maytenus aquifolium*, and *Mikania glomerata*?

1.7 Contributions

We summarize the main contributions of this study as follows:

1. We introduce a novel integration of untargeted metabolomics and machine learning for the quality control of Brazilian medicinal plants, addressing the limitations of traditional methods.

2. We provide a practical quality control framework for two widely used Brazilian medicinal plants, with MCC scores of 94% and 97% for *M. ilicifolia* and *M. aquifolium*, respectively, demonstrating the effectiveness of machine learning in distinguishing between target samples.

3. We reveal that the most important features for classification are not necessarily the traditional chemical markers, proving that focusing solely on these compounds is inefficient for quality control. This highlights the advantage of using untargeted metabolomics, which captures a broader range of metabolites and provides a more holistic view of the sample's chemical composition.

1.8 Outline

The remainder of this text is structured as follows:

In the Material and Methods section we describe the samples, equipment and tool used to create the quality control methods for both target species. In the Results and Discussion we describe the development cycle of both models which started with the Metabolomics Experiment and XCMS parameter optimization and preprocessing. After data treatment, the Exploratory Data Analysis was performed only on the train set, and the first baseline model was performed, described on the First Model Evaluation section.

All further steps taken to achieve the final models were performed based on the conclusions of the previous steps and the evaluation of previous scores. The baseline model, for example, demonstrated suspiciously high scores, which motivated the further separation of the train set into train and validation sets, before the XCMS preprocessing. Data Leakage Prevention section describes the steps performed to avoid this issue.

Afterwards, the models were fine-tuned using GridsearchCV, described in Model Tuning with GridSearchCV section. The scores obtained were still suspiciously high, therefore, in section "Incorporating further data", further data was incorporated into the Maytenus model, by using samples collected and analyzed previously by the research group. The hypothesis was that the model was over-adjusting to the specific equipment conditions of the samples obtained and analyzed during the development of this work.

After evaluating both models, the next steps were Normalization of the feature's relative intensities , Feature Selection and testing the Model Robustness and Validation Methods. Finally,

the models were trained once more and the best-performing models for each target species were selected, tested and analyzed, to obtain the most important features on each model (sections Final Model, Final Model Testing, and Final Model Analysis)

The final Machine Learning Application Proof-of-concept section, described the machine learning application that was created to demonstrate how these models could be used without the need to install and configure a system by unfamiliar and non-technical users, without much programming expertise.

2 Material and Methods

2.1 Experimental design and sample extraction

Maytenus ilicifolia, *Maytenus aquifolium*, *Mikania glomerata* and *Mikania laevigata* leaf samples used in this study were obtained from the Pluridisciplinary Center for Chemical, Biological and Agricultural Research (CPQBA). In total, 600 leaf samples from 600 different plant individuals were collected, of which 200 belonged to *M. ilicifolia* and *M. laevigata* each, and 100 were collected from *M. aquifolium* and *M. glomerata*, each. The samples were collected in September and December of 2021 and after harvest, the samples were frozen at -80°C, lyophilized, and ground.

Additional data used on *M. ilicifolia* model training was also obtained from samples harvested at CPQBA during a previous study from the research group. These samples were harvested during a year from November 2016 to October 2017, being collected once per month. In total, 60 samples were harvested from *M. ilicifolia* and *M. aquifolium* specimens, each. These samples were also frozen at -80°C, lyophilized, and ground and were kept in storage for around 5 years, before extraction for the present study (Antunes et al., 2019).

Table 1 demonstrates how the samples were separated for each experiment. Samples from the negative class on each model were obtained from the same individuals in the other model. For example: For the Maytenus model, as observed, 50 samples belonged to *M. laevigata.* The plant individuals that provided these samples were also harvested to create the Mikania model. This was done due to logistics and the availability of dried materials, which was enough to cover both models.

Table 1: Experiment design, specifying the amount of samples separated for each algorithm and each group.

Maytenus algorithm		**Mikania algorithm**	
Species	**Number of samples**	**Species**	**Number of samples**
Maytenus ilicifolia	260	*Mikania laevigata*	200
Maytenus aquifolium	160	*Mikania glomerata*	100
Mikania laevigata	50	*Maytenus ilicifolia*	50
Mikania glomerata	50	*Maytenus aquifolium*	50

As the project aimed to create one ML model for each 'target' species, two extraction methods were performed, one for the *M. ilicifolia* model and another for the *M. laevigata* model. All samples in each experiment were extracted by their respective method.

The Brazilian Phytotherapy Formulary and previous studies were used for sample extraction as references. As stated in the Formulary, *M. ilicifolia* is used in the form of herbal tea (ANVISA, 2019), therefore, for the *M. ilicifolia* model, the samples were extracted using water, in a proportion previously defined in other studies. The extract was carried out using 2 mg of freeze-dried ground leaves and 1 mL Milli-Q purified water in an ultrasonic bath for 30 min at 20 ∘C (Antunes et al., 2019, 2020).

For the *M. laevigata* model, as the Formulary preconizes its use as a hydroethanolic tincture,

the samples were extracted using 70% ethanol (v/v). The extraction procedure was also carried out following a method previously determined in the research group, in which 3 mg of dried plant powder was extracted using 1 mL of 70% ethanol (v/v), also assisted by an ultrasonic bath, for 30 min, then centrifuged at 13,000× g for 5 min (Borghi et al., 2020).

2.2 Metabolomic analysis

Two untargeted metabolomic experiments were carried out, one for each of the target species. The first analysis was performed with *M. ilicifolia* (200 samples), *M. aquifolium* (100 samples), *M. laevigata*, and *M. glomerata* (50 samples each). The second analysis was performed with *M. laevigata* (200 samples), *M. glomerata* (100 samples), *M. ilicifolia*, and *M. aquifolium* (50 samples each). All samples were analyzed in triplicate.

For both experiments, two types of quality control samples (QC) were used to track instrument and extraction variations in the analysis. The first type of QC sample was created by mixing 2 mg of each dried sample, within each experiment, resulting in two pools of dried samples that were, later, used to create 57 other QC samples by also weighing 2 mg of such pool and extracting it according to each experiment. The second method was to extract 10 μL from each extract creating one single sample that was analyzed at regular intervals of every 66 samples, resulting in 19 injections, also within each experiment.

Both analyses were carried out using an Ultra-High-Performance Liquid Chromatographer (UPLC® Acquity from Waters) coupled with a TQD Acquity mass spectrometer, with an ESI source. The analytical column was a C18 BEH Acquity Waters (1,7 μm × 2,1 mm × 50 mm), with an oven temperature of 30°C. The chromatographic methods applied were optimized in previous studies (Antunes et al., 2019, 2020; de Lazzari Almeida et al., 2017c). The solvents used were purified water with 0.1% of formic acid (A) and acetonitrile (B).

For *M. ilicifolia* analysis, the gradient began with 95% A and 5% B, ramping to 7% A and 25% B in 4 min, 50% A and 50% B in 6.10 min, 1% A and 99% B in 6.20 min, returning to 95% A and 5% B in 8.50 min, stabilizing until 10 min, at 200 μL/min flow rate. For *M. laevigata* analysis, the gradient started with 90% A and 10% B, ramping to 75% A and 25% B in 4 min, 0% A and 100% B in 8 min, and returning to 90% A and 10% B in 8.51, stabilizing until 10 min, also at 200 μL/min flow rate.

The mass spectra for both analyses were acquired with electrospray ionization at negative mode (ESI-), at full scan, with the following conditions: capillary at 3.00 kV, cone at 35 V, extractor at 1 V, source temperature at 150 °C, and desolvation temperature at 300 °C. The QC samples were also analyzed at regular intervals throughout the LC-MS analysis, with other samples randomized and analyzed in triplicate, for both experiments.

2.3 Data processing and feature extraction

After the metabolomic analysis, the data processing and feature extraction steps were carried out. First, the raw data (.raw) was converted into mzXML format (Pedrioli et al., 2004) using

msconvert (Chambers et al., 2012). Subsequently, the files were separated into train, validation, and test folders using the Python split-folders library (0.5.1, Python version 3.9.7).

For the *M. ilicifolia* model, samples acquired during a previous study were also divided into train, validation, and test folders and included in the experimentation. The samples were acquired using the same LC-MS and extraction methods but the acquisition occurred at the end of 2017. The conversion of these files to mzXML was done using the same msconvert code.

Next, using only the training samples, the parameters for data extraction were optimized using the R package IPO (1.14.0, R version 4.0.5) (Libiseller et al., 2015). The optimization was done separately for each model.

Finally, the data was processed using the xcms and CAMERA R packages (3.10.1, 1.44.0, R version 4.0.5), using their respective optimized parameters (Smith et al., 2006; Kuhl et al., 2012). The preprocessing generated a table of mass feature intensity, with retention time and *m/z* values for each sample analyzed. The code used for file conversion, IPO, and xcms steps is available in the Supplementary Material section.

2.4 Algorithm selection and model construction

Before the algorithm selection and model construction, an initial exploratory analysis was performed with the tabular data obtained by the XCMS preprocessing. For this, both R and Python languages were used, with their respective packages and libraries. The code for all exploratory analyses is also available in the Supplementary Material section.

For the ML model construction, all the experiments were implemented in Python programming language using the scikit-learn package (version 1.3.2). The experiments followed a supervised approach, where the goal was to classify the samples as *M. ilicifolia* and *M. laevigata*, in comparison to their counterparts *M. aquifolium*, and *M. glomerata*. Well-known classification algorithms were used: Support Vector Classification(SVC), K-Nearest Neighbours Classifier, and Random Forest (Pisner e Schnyer, 2020; Steinbach e Tan, 2009; Cutler et al., 2012).

Along with model training feature selection was performed using three distinct methods: Mutual Information, Recursive Feature Elimination, and Boruta. Additionally, the data was normalized using the Normalized method of the scikit-learn package. Finally, to aid the selection of the best hyperparameter settings and feature selection method for each algorithm and determine the best model, GridSearchCV was used along with inter-dataset validation and intra-dataset validation.

To verify the models performed within the GridsearchCV, a validation curve was plotted for each hyperparameter tested. This allowed tracking the impact of each hyperparameter on the models and determine when to stop the experimentation.

The algorithm performance was measured by the Matthews correlation coefficient (MCC) and the F1 score, both during the cross-validation and testing steps. The MCC score measures the quality of binary and multiclass classification and considers true and false positives and negatives. The MCC score ranges from -1 to 1 in which 1 is a perfect prediction, 0 is an average

random prediction, and -1 is an inverse prediction. MCC score is unaffected by the unbalanced datasets issue and is defined as follows:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}, \tag{1}$$

where TP is True Positives, FP False Positives, TN True Negatives, and FN False Negatives, obtained from a confusion matrix.

The F1 score is given by the harmonic mean of the precision and the recall with equal weights (importance) for both metrics, ranging from 0 to 1. The formula for the F1 score is:

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{2}$$

2.5 Application

With the final ML models for each species, a web application was developed using the Streamlit framework. Streamlit is an open-source Python library that permits the creation of web applications for machine learning and data science.

The application architecture was designed to encapsulate the entire process after the file conversion to mzXML. The goal was to serve as a proof-of-concept and user-friendly interface to perform the quality control of *M. ilicifolia* and *M. laevigata*. It performs various preprocessing steps on mzXML files and returns the probability of a given sample being from one of the 'target' species. The user does not need to run any code and all the necessary configurations, processing steps from metabolomics, and data preparation for the machine learning model are done on the "back-end".

The core components of the application include target species selection, data uploading, Metabolomics preprocessing, data preparation for the machine learning model, and sample classification.

The application was deployed on a cloud-based server on Streamlit itself, ensuring accessibility from any device with an internet connection. This allows users to easily interact with the application and obtain classification results without requiring local installations of complex software or programming knowledge.

2.6 Data availability

All the code is available at `https://github.com/ElisaRMA/ML_metabolomics` and the repository for the application is at `https://github.com/ElisaRMA/quality_control_app`

3 Results and Discussion

3.1 Metabolomics Experiment

While there is no universally used workflow in the field of metabolomics (Patti, 2011), some steps are common and essential for all studies. Metabolomic experiments usually begin with the biological question, experimental design, and sample analysis, which follow a very different process depending on whether the study follows a targeted or untargeted metabolomics approach.

Following sample analysis, a preprocessing tool is needed to extract the data from the resulting chromatograms, especially for untargeted metabolomics data. The first step is usually peak picking (or deconvolution) to later align and integrate peak data across multiple samples. Upon completing preprocessing, a matrix of 'metabolite features' (mass-to-charge ratio and retention time pairs) is obtained, and the results can be analyzed using different techniques (Di Guida et al., 2016). The present study followed such workflow, which is presented in Figure 4.

As previously mentioned, for the data acquisition step, two experiments were performed for each 'target' species. Table 1 in the Material and Methods section demonstrates how the samples were separated for each experiment. The LC-MS methods used for each species were previously developed and validated (Borghi et al., 2020; Duarte et al., 2022). Below are the representative chromatograms for each species in both experiments (Figure 5 and 6)

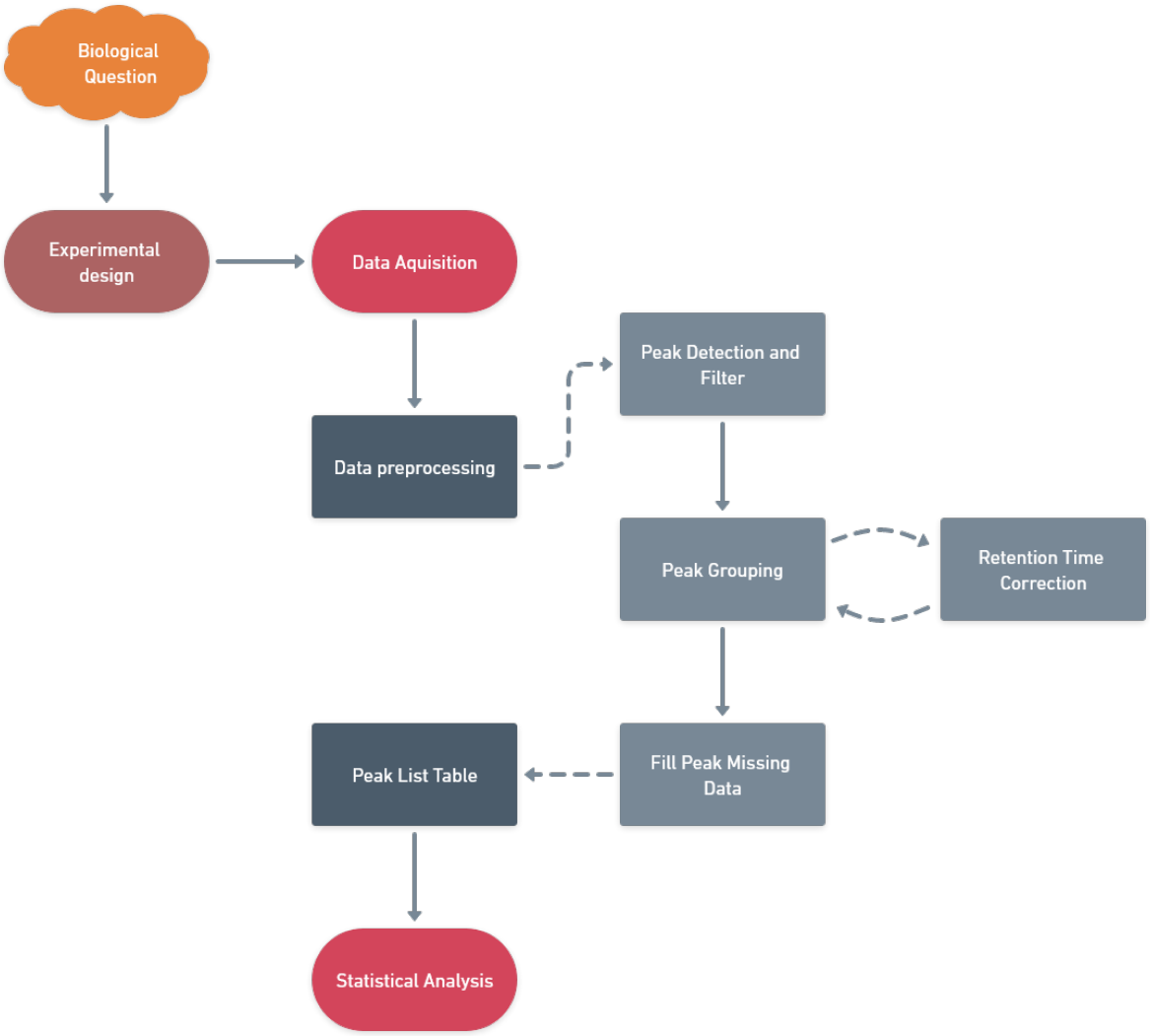

Figure 4: General metabolomics workflow.

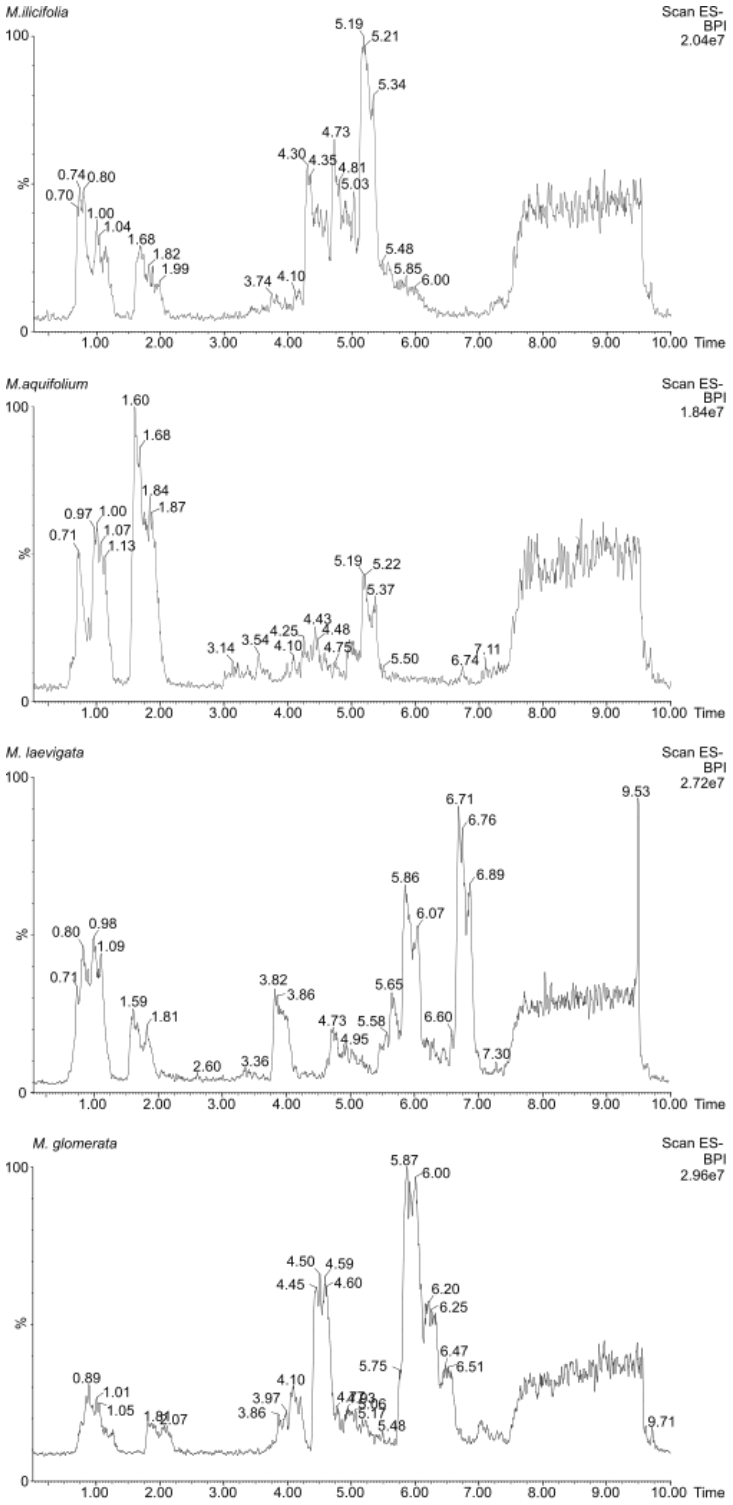

Figure 5: Chromatograms using the UHPLC-MS conditions for *Maytenus* samples. A) *M. ilicifolia*, B) *M. aquifolium*, C) *M. laevigata*, D) *M. glometara*

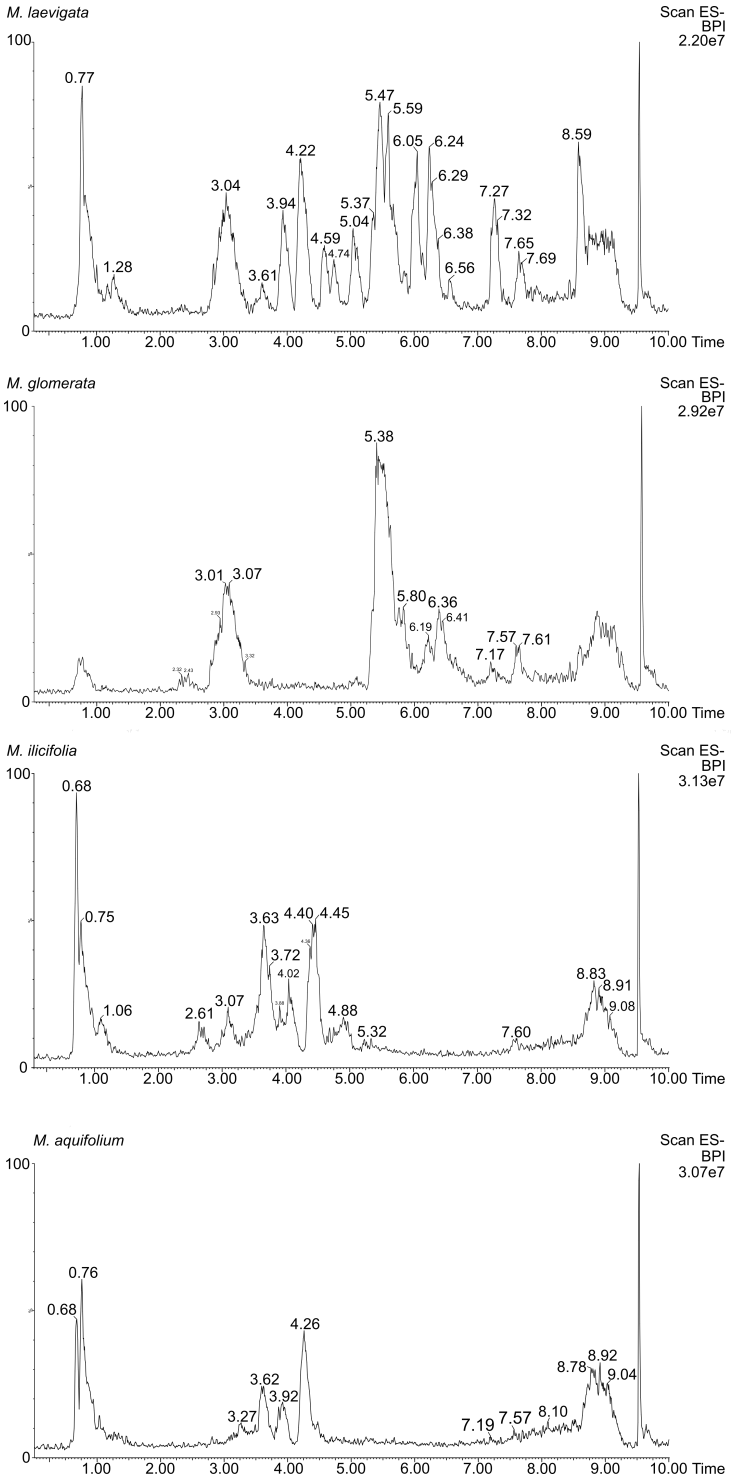

Figure 6: Chromatograms using the UHPLC-MS conditions for *Mikania* samples. A) *M. laevigata*, B) *M. glometara*, C) *M. ilicifolia*, D) *M. aquifolium*

3.2 XCMS parameter optimization and preprocessing

After the chromatographic analysis, the files were converted to mzXML format. The mzXML format is an open-source XML representation of mass spectrometry (MS) data that allows the comparison of results obtained from different equipment. By converting the propri-

etary file formats to mzXML and standardizing their use, it is possible to process and analyze the chromatographic data using open-source tools instead of the software provided by the vendors. A widely used preprocessing tool created for this purpose is XCMS (Pedrioli et al., 2004).

XCMS is a framework for processing and visualizing LC-MS data. The software is written in R language and it is freely available under the open-source license, which enables the analysis of LC-MS data derived from instruments of different manufacturers, as mentioned before. The preprocessing strategy used by XCMS comprises four main steps: peak detection, peak matching, retention time alignment, and filling of missing peaks (Smith et al., 2006).

The peak detection step starts by cutting the LC/MS data into slices with the width measured in the mass-to-charge ratio (*m/z*). These slices are combined and filtered using a second-derivative Gaussian as the model peak shape. Peaks that match this shape are then filtered once more using a signal-to-noise ratio cutoff, to be classified as 'valid' peaks.

Subsequently, these peaks are matched across samples to allow retention time correction and relative ion intensity comparisons. The algorithm calculates the overall distribution of peaks and identifies the boundaries where many peaks have similar retention times. These peaks are analyzed and the algorithm defines an interval in which all peaks inside would be placed into a group, representing an analyte. Finally, these groups are filtered based on their occurrence within the samples, and peaks that are not present in a significant amount of sample classes are removed. Alternatively, consistently present peaks across the samples are considered 'well-behaved' peak groups .

After the peak matching and filtration, the retention time alignment can be performed. First, the well-behaved groups determined previously are used as standards, and for every group, the algorithm calculates the median and deviation of retention time on every sample in that peak group. Then, the retention time drift is corrected using a local regression fitting method. This correction is done simultaneously on all samples.

Finally, the last step is filling the missing peaks. For this, XCMS first identifies which samples are missing from each peak group. Then, using data collected on peak detection about where peaks started and ended, along with their aligned retention times, the raw LC/MS data are integrated to fill in intensity values for each missing data point.

After all steps, the XCMS preprocessing returns a peak table with the relative intensity of each analyte along with the median ('mz'), minimum ('mzmin'), and maximum ('mzmax') *m/z* of peaks in the groups, as well as median ('rt'), minimum ('rtmin') and maximum ('rtmax') retention time of peaks in the group. In this context, the metabolite features are two-dimensional vectors, formed from the combination of these *m/z* and rt values. For example, one feature could have an *m/z* of 289 and an rt of 2.5 min, while another could have the same *m/z* but an rt of 3.4. In such a case, both analytes would be considered different features as the *m/z* and rt combinations differ. Table 2 illustrates an example of a partial table of features extracted from XCMS preprocessing.

Table 2: Features table example obtained after XCMS preprocessing

mz	mzmin	mzmax	rt	rtmin	rtmax	npeaks	sample1	sample2	sample3
106.26	105.89	106.62	559.48	554.90	566.20	898	4.38E+08	4.06E+08	4.13E+08
105.93	105.88	106.46	466.36	456.49	502.44	872	4.29E+08	3.54E+08	4.20E+08
113.77	113.13	114.12	53.00	36.06	116.69	147	2.85E+07	2.55E+07	2.65E+07
115.63	115.12	116.12	57.61	49.19	129.09	504	3.08E+07	2.84E+07	2.85E+07
116.40	116.12	117.11	55.73	49.40	176.88	112	2.82E+07	2.54E+07	2.68E+07
116.87	116.62	117.60	439.50	430.69	445.26	669	5.64E+07	6.30E+07	5.43E+07
117.62	117.13	118.12	109.92	75.06	177.31	133	2.12E+07	2.00E+07	2.23E+07
127.71	127.12	128.11	565.11	558.21	570.22	118	6.64E+07	6.10E+07	6.96E+07
128.65	128.12	129.12	562.51	555.60	568.50	255	8.22E+07	8.14E+07	8.58E+07

It is important to note that for each of the steps mentioned, XCMS takes a range of parameters to process the LC-MS data. Such parameters must be modified based on the analysis itself and to enhance the XCMS performance, these parameters can be optimized using another open-source software called IPO ('Isotopologue Parameter Optimization') (Libiseller et al., 2015).

IPO is another R package, that was created to determine the best set of parameters for the XCMS experiments, increasing the reliability of the results of peak picking, retention time correction, and grouping. The optimization for the peak picking parameters is the first step, and afterward, the retention time correction and grouping parameter optimization are done simultaneously, as grouping requires the correction of retention time, which in turn, can improve the grouping as a whole Libiseller et al. (2015).

According to the authors, the parameters are determined by a design of experiments approach (DoE) in which specific modifications are made to the input variables (parameters) to optimize or explain the changes in the response variable (feature table). For each parameter, three different values are tested. The two outer values (smaller and larger values) determine the range and the middle value determines the center. The values are tested and the result of the DoE is evaluated based on two scores determined by the authors: one for peak picking and one for retention time correction and grouping (Libiseller et al., 2015).

In summary, the parameters converge to increase the number of reliable peaks and reliable groups and simultaneously decrease the shift in retention time between peaks in a peak group. Reliable peaks are defined as peaks belonging to an isotopologue (13C isotope peaks) and reliable groups, according to the authors, "are assumed to show exactly one peak from each injection of a pooled sample" (Libiseller et al., 2015).

The present study used the IPO package and the training samples to determine the XCMS parameters for both species. In both cases, a slight modification was done to the optimized parameters to better represent the data. The XCMS final preprocessing generated 306 features for the *Maytenus* experiment and 148 features for the *Mikania* experiment that were used for model training. The parameters and code for IPO and XCMS steps are available in the Supplementary Material.

3.3 Exploratory Data Analysis

Before and during the algorithm selection and model construction, data exploration was performed to better understand if further data processing was necessary. According to Géron (2019), the initial steps of creating a Machine Learning model involve understanding the data and studying the problem. Therefore, the first analysis was the MS total useful signal (MSTUS) to verify if the samples suffered intensity drifts.

As LC-MS methods are subject to instrumental variations, differences in peak intensities are common. According to Jiang et al. (2020) such variations can normally go beyond 10% during an LC-MS series of runs. The reasons may include changes in detector sensitivity, and variations in the electrospray process, among others (Jiang et al., 2020). If such variations are too accentuated, the result of the analysis can be affected demanding further processing to reduce the systematic variation. In the present work, a large number of samples were analyzed, therefore, an inspection of the MSTUS plot was necessary.

The MSTUS plot uses the total intensity of all features for each sample, across the LC-MS run and plots these intensities arranged by the injection order. If the points on the plot form a pattern other than a random distribution, intensity drifts could have occurred in the analysis. Some researchers use such analysis to clean the data, removing a sample with an MSTUS higher or lower than three standard deviations from the mean (Rodríguez-Coira et al., 2019).

In the case of the present work, fortunately, no significant changes occurred, as observed in Figures 7 and 8, where all data points are evenly distributed along the plots and contained within three standard deviations of the mean. For the MSTUS of the *Mikania* experiment in Figure 8, the only difference in intensities was shown by some *Maytenus* samples. This difference, however, was not exacerbated, and, as mentioned, the *Maytenus* samples were still contained within 3 standard deviations from the mean.

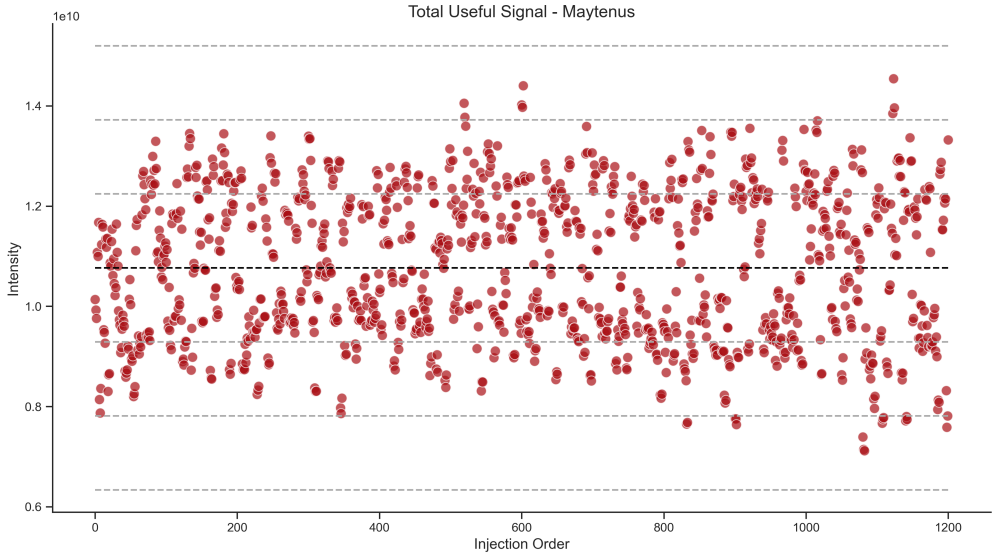

Figure 7: Total Useful Signal for *Maytenus* experiment. The dashed lines represent the mean (black) and the standard deviations (grey)

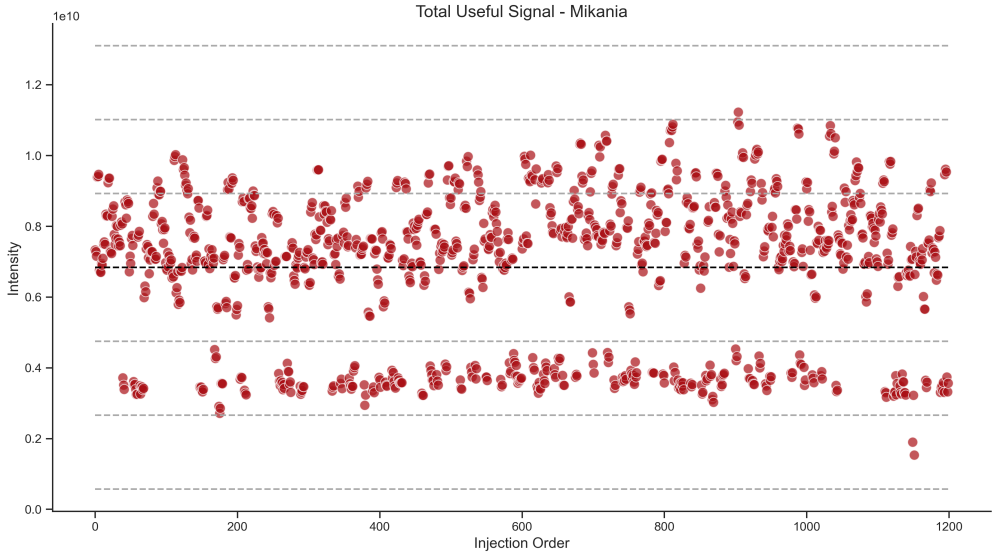

Figure 8: Total Useful Signal for *Mikania* experiment. The dashed lines represent the mean (black) and the standard deviations (grey)

However, later in the experimentation, during the model training and fine-tuning, the validation scores obtained were too high, indicating to the possibility of overfitting, as will be discussed later. To address this issue, a normalization step was added in the processing pipeline and a comparison between non-normalized and normalized data was done using the MSTUS plot. This will be explained in more detail in the section 3.4.5

It was observed that for each experiment, samples from the other genus presented lower feature intensities, dividing the MSTUS plot into 'groups'. For instance, in the *Maytenus* experiment, samples from the *Mikania* genus were significantly lower in the MSTUS plot in comparison with samples from the *Maytenus* genus. This could have occurred due to the analytical and extraction methods not being optimized for the *Mikania* genus, resulting in lower intensities for these samples. The same effect was observed for the *Mikania* experiment, in which *Maytenus* samples presented lower intensities.

Such differences in feature intensity could be one of the reasons behind the high scores obtained in later stages of the model training and fine-tuning. This is because the model created could be over-adjusting to classify the samples based on general intensity instead of the features themselves and the biological differences between the classes.

Figures 9 and 10 show the comparison between the previous MSTUS, color-coded by their classes and their normalized counterparts, for both experiments. It is possible to notice, after normalization, that the samples from the two classes were evenly distributed within the plot. The consequence of the normalization on the model's performance will be demonstrated in the next sections.

Next, to verify how the classes were separated from each other, a PCA was plotted for both experiments (Figures 11 and 12) just to inspect how the samples clustered. As observed, by labeling the samples according to their species it is evident that the samples were clustered together based on their genus, due to their higher similarity. For this reason, it was expected that the machine learning algorithms would not have difficulties classifying the samples.

For the *Maytenus* experiment, *M.laevigata* and *M. glomerata* samples clustered together in the upper left corner on the PCA scores plot in Figure 11 and for the *Mikania* experiment, *M. aquifolium* and *M. ilicifolia* samples clustered together on the right lower corner of the PCA from Figure 12. When labeling the samples according to their class (positive or negative) some overlap is observed, especially in the case of the *Mikania* experiment, indicating that the classification model for this experiment might present lower scores.

Such overlap could have occurred due to human errors while identifying the species during the harvesting step. As both species present many morphological similarities, identification errors are common, and depending on the result from the machine learning algorithm, it is possible that the *M. laevigata* samples clustered with *M. glomerata* (right upper corner of the first PCA on Figure 12) may, in fact belong to such a group.

The PCA plot was also generated with normalized data, to compare the distribution after the normalization step. Figures 13 and 14 show the PCA scores plot on the normalized data, the same data used for the MSTUS shown previously. As observed, the grouping was maintained after the normalization and, for the *Mikania* data, the normalization resulted in a PCA scores plot with groups further apart.

Additionally, both PC1 and PC2 values for the normalized plots were higher than for the unprocessed data. For *Maytenus*, the PC1 and PC2 for the unprocessed data were 29.66% and 11.7%, and for the normalized data, the PC1 and PC2 were 29.83% and 13.76%, respectively. For

Figure 9: MSTUS plot of non-normalized (Top) and normalized (bottom) *Maytenus* data. The colors indicate the different classes (POS: samples from *M. ilicifolia*, NEG: samples from *M. aquifolium*, *M. laevigata* and *M. glomerata*)

Figure 10: MSTUS plot of non-normalized (Top) and normalized (Botton) *Mikania* data. The colors indicate the different classes (POS: samples from *M. laevigata*, NEG: samples from *M. glomerata*, *M. aquifolium*and *M. ilicifolia*)

Mikania, the unprocessed data yielded a PC1 and PC2 of 29.46% and 16.81%, respectively, while the normalized data yielded a PC1 and PC2 of 38.37% and 20.48% respectively.

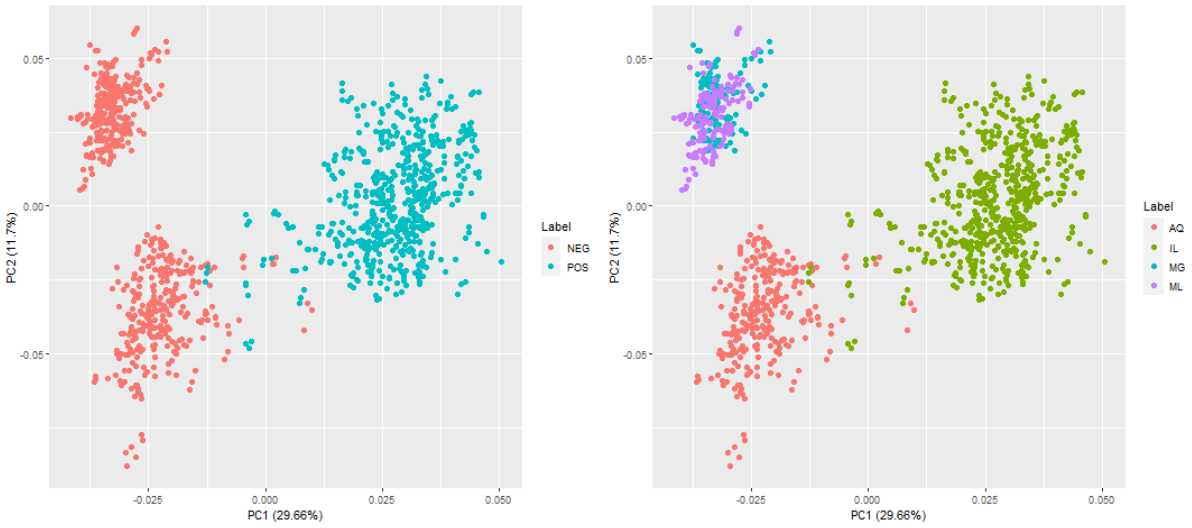

Figure 11: PCA scores plot for the *Maytenus* model unprocessed data. Label NEG and POS represent the group used for the machine learning model. POS is the target class, in this case, samples from *M. ilicifolia*, while NEG is all other samples. AQ, IL, MG, and ML are the abbreviations for the species name, *M. aquifolium M. ilicifolia*, *M. glomerata* and *M. leaevigata*, respectively.

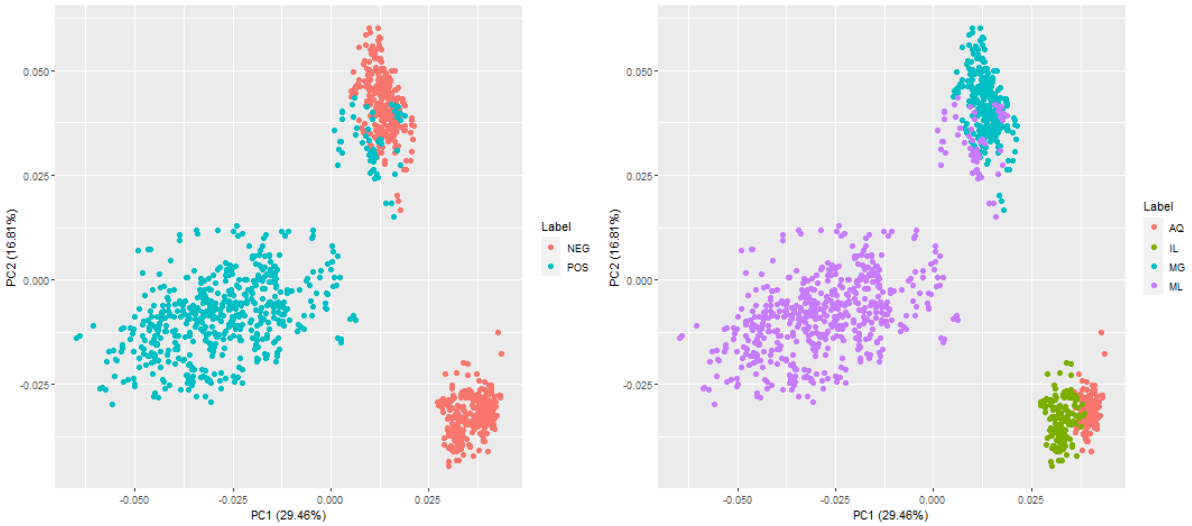

Figure 12: PCA scores plot for the *Mikania* model unprocessed data. Label NEG and POS indicate the group used for the machine learning model. POS is the target class, in this case, samples from *M. laevigata*, while NEG is all other samples. AQ, IL, MG, and ML are the abbreviations for the species name, *M. aquifolium M. ilicifolia*, *M. glomerata* and *M. leaevigata*, respectively.

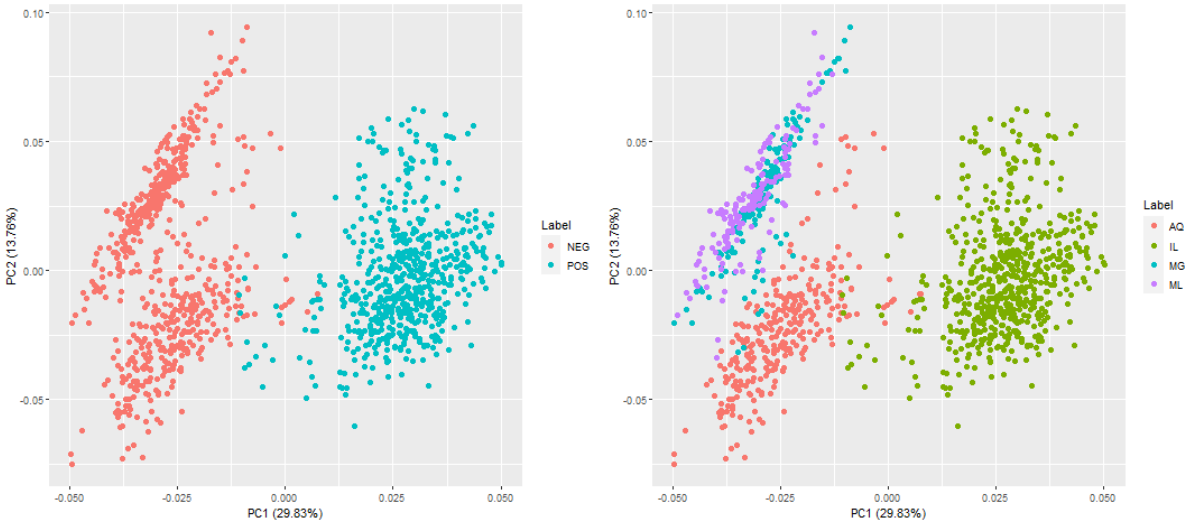

Figure 13: PCA for scores plot for the *Maytenus* model normalized data. Label NEG and POS indicate the group used for the machine learning model. POS is the target class, in this case, samples from *M. ilicifolia*, while NEG is all other samples. AQ, IL, MG, and ML are the abbreviations for the species name, *M. aquifolium M. ilicifolia*, *M. glomerata* and *M. leaevigata*, respectively.

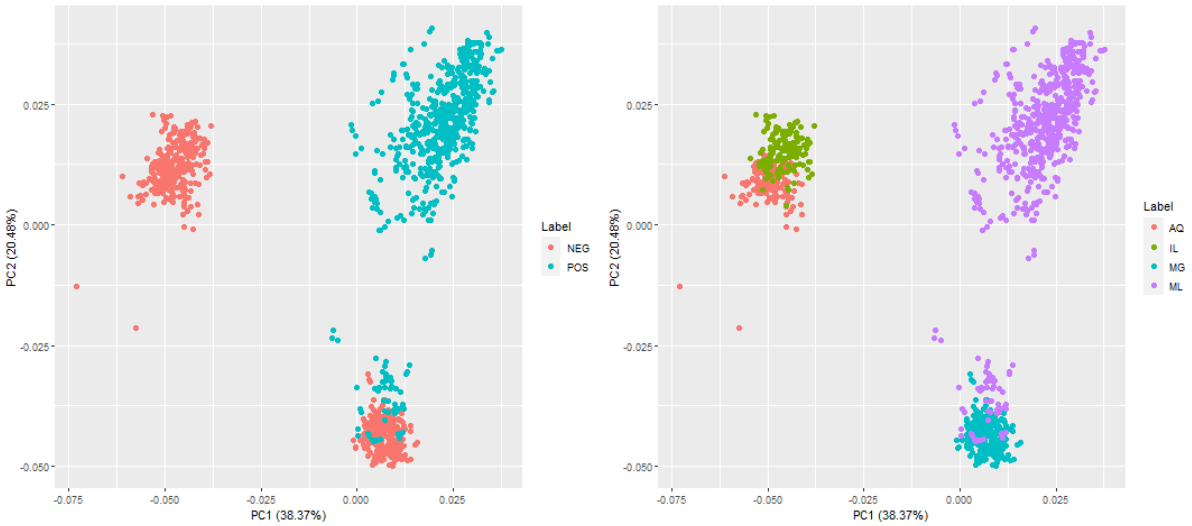

Figure 14: PCA scores plot for the *Mikania* model normalized data. Label NEG and POS indicate the group used for the machine learning model. POS is the target class, in this case, samples from *M. laevigata*, while NEG is all other samples. AQ, IL, MG and ML are the abbreviations for the species name, *M. aquifolium M. ilicifolia*, *M. glomerata* and *M. leaevigata*, respectively.

3.4 Algorithm selection and model construction experiments

After the initial data exploration and in possession of the feature table obtained in the preprocessing step, the model training and fine-tuning could begin. As mentioned, the goal was to create two models, one for each 'target' species (*M. ilicifolia* and *M. laevigata*). Many experiments were performed to achieve this, with different hyperparameters, features, and strategies.

However, it's important to note that during experiments, a lot of back and forth was performed between the XCMS preprocessing, data exploration, and model training. Since different parameters on XCMS would return a different dataset, small modifications to its parameters were performed, to find the best final model with the highest reasonable score.

3.4.1 First Model Evaluation

In order to train the model, for all experiments, the mzXML files were separated into training and test folders and the test was put aside. For the first experiment, however, the files in the training folder were all used for IPO optimization and XCMS preprocessing, and, only afterward, the resulting table of features was split into training and validation sets. Therefore, this split was initially performed in the traditional way, during the model training itself.

In this experiment, the model training began by using three initial classifiers with minimal changes to algorithm parameters. The objective was to verify how these algorithms would perform on the data, without any fine-tuning. The scores obtained would serve as a baseline for further models. The classifiers used were: Support Vector Classification (SVC), K-Nearest Neighbours Classifier (KNN), and Random Forest.

The SVC algorithm works by finding a line of separation, known as hyperplane, between data from two classes. This line seeks to maximize the distance between the closest points concerning each of the classes (Pisner e Schnyer, 2020). In this study, the classification was binary, so the classes were either 1 or 0 for samples that were from the target species or not, respectively.

The KNN algorithm works by initially finding the k-nearest neighbors of a given instance. Then, the class of the given instance is determined based on the class that occurs most frequently among its k neighbors. This algorithm has two parameters: the number of neighbors (k) and the dissimilarity measure used to find the nearest neighbors. Euclidean distance is the most widely used measure to determine neighbors, although there are several options (Steinbach e Tan, 2009).

Finally, the Random Forest is an ensemble of decision trees combined with bagging. When using bagging, different trees see different portions of the data, therefore neither tree sees all the training data. This means that each tree is trained with different data samples for the same problem. By combining their results, the errors of some trees are compensated for with the results of others, leading to a prediction that generalizes better (Cutler et al., 2012).

Table 3 summarizes the results of the first experiments obtained for both species with the

three classifiers. The scores shown in the table are related to the training set. As observed, all classifiers presented high scores, on both experiments, suggesting that overfitting or data leakage could be happening.

Table 3: Scores for the first experiment

	Model 1: Identification of M. ilicifolia	**Model 2: Identification of *Mikania* laevigata**
SVC		
MCC	0.962	0.937
F1	0.980	0.967
RandomForestClassifier		
MCC	0.981	0.927
F1	0.990	0.959
KNeighborsClassifier		
MCC	1.0	0.994
F1	1.0	0.997

According to James et al. (2023), overfitting means the model is following the errors or noise in the data too closely. In this situation, the score of the model is usually very high, and a perfect fit, as observed in Table 3, will almost certainly indicate it is overfitting the data. According to the authors, even though is possible to perfectly fit the data in a high dimensional setting, the model will perform poorly on new independent data. Since for this initial experiment, only small modifications were done on the XCMS parameters and no feature selection was performed, it's possible that the model was following errors or noise in the form of irrelevant features.

In addition, as mentioned, the high score may be due to a certain amount of data leakage. According to the Pedregosa et al. (2011), and Walker (2022), data leakage occurs when information not available at prediction is used to train a model. In other words, data from the test or validation sets 'leaks' into the training set.

This problem usually happens during data preprocessing, for example, during missing values imputation by using the mean. In this case, if imputation is done before the split, the mean values will be calculated using data from both the training and validation/test sets. In this situation, the model validation will not be efficient and its performance will be overly optimistic during training.

In the present study, the preprocessing using XCMS, which served as a feature engineering step was applied before the model training. As discussed previously, the XCMS preprocessing method involves matching chromatographic peaks across samples to correct the deviation in retention time and group such peaks to create features.

In this situation, if samples from training and validation are kept together for XCMS preprocessing, data from both sets would be matched with each other to determine the peak groups. Additionally, the retention time correction would take into account the drift on both sets and as a result, this validation set wouldn't serve as a reliable 'simulation' of the test set. Hence, the first step taken to improve these scores and avoid data leakage was to split the mzXML files into

training, validation, and test folders before any metabolomics preprocessing step.

3.4.2 Data Leakage Prevention

After splitting the mzXML files into training, validation, and test folders, the IPO parameter optimization was performed again only on the training set, to determine the best parameters for XCMS. Afterward, the training and validation sets were processed separately by XCMS, resulting in two different feature tables. It is important to note that, at this point, the test set was still put aside and the XCMS preprocessing for this data happened only at the end of the study, which will be discussed in the next subsections.

However, since this XCMS preprocessing was done separately for train and validation, the peak groups and retention time drifts were slightly different on both sets; therefore, the feature names were not the same. As the process of validating a model requires the same features used in training (Pedregosa et al., 2011), an expert system for feature correspondence had to be created.

The expert system created considers the feature names from the training set as a standard and applies the same name to the validation's set features based on conditionals. These feature names are obtained by concatenating the 'mz' and 'rt' values into one term ('mz_rt'), used to represent a given compound ('mz') at a given retention time ('rt'), which is the feature itself in the context of metabolomics, as mentioned before.

The first conditional test on the expert system is performed on the 'mz' values. If the 'mz' value of a given sample from validation is between 'mzmax' and 'mzmin' from the training set, then, for those features that pass the condition, the evaluation continues to check the rt values. In such a case, if any of the rt values from validation are between the 'rtmin' and 'rtmax' from training, then the name of the corresponding feature is used on validation. Figure 15 illustrates the expert system created.

It is possible that, during this conditional testing, a given feature on the validation set could receive the name of more than one feature from the training set if their mz and rt values are within the constraints. In the case of such ties, the 'npeaks' column is used to determine which feature name will be used. The 'npeaks' column on the feature table provides information on the number of samples that presented such a feature Smith et al. (2006). Consistent and valid peaks are present in most of the samples of a given class; therefore, peaks with a higher 'npeak' value are given preference in naming the validation feature.

After passing the data through the expert system pipeline and naming the validation features, the model training could begin. In this second experiment, the hyperparameters of the models were also not optimized and Table 4 summarizes the score for the models.

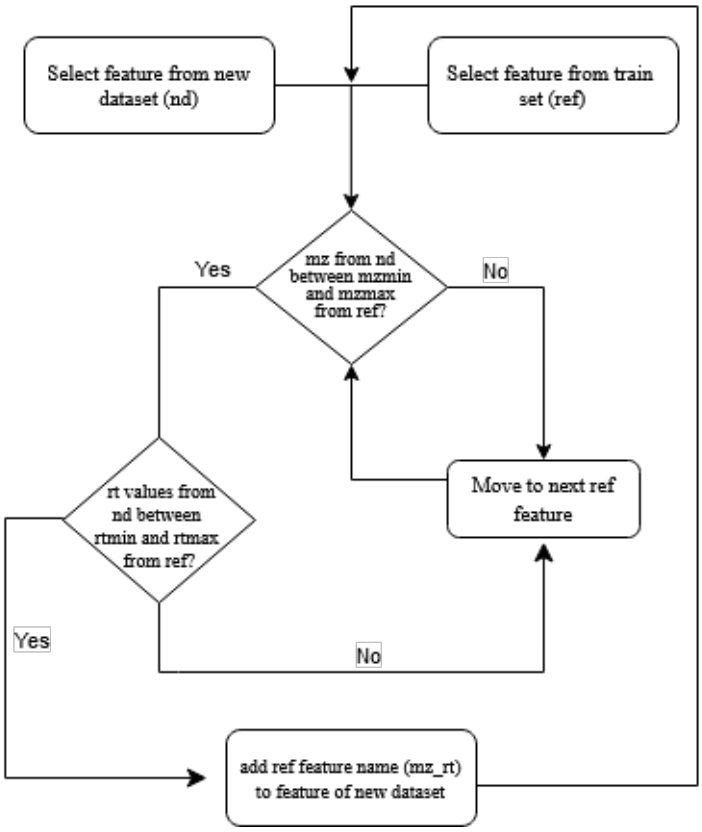

Figure 15: Expert System to create feature names (mz_rt) for the validation and test sets

Table 4: Scores for the second experiment

	Model 1: Identification of *M. ilicifolia*	**Model 2: Identification of *Mikania* laevigata**
SVC		
MCC	1.0	0.404
F1	1.0	0.439
Random Forest Classifier		
MCC	1.0	0.0
F1	1.0	0.0
K-Neighbors Classifier		
MCC	0.969	0.802
F1	0.985	0.903

As observed, for all classifiers *M. ilicifolia* still presented a high score, indicating possible overfitting. For *M. laevigata,* the score on KNN was still very high, but it significantly decreased for SVC and Random Forest, indicating that splitting the data into training and validation had a high impact on the model's performance and the next step would be to improve the model itself.

3.4.3 Model Tuning with GridSearchCV

Considering the results obtained in the previous experiment, 'GridSearchCV' with 'PredefinedSplit' was implemented in the third experiment. It is important to note that, since the training and validation were separate datasets, the function 'PredefinedSplit' was used to apply the train and validation sets in the cross-validation process itself, instead of allowing the function to split the training set.

In addition, to determine the best parameters and find the best model, the MCC score was used under the 'refit' parameter of GridSearchCV. For cross-validation, since the PredefinedSplit was used, the inter-dataset validation was applied, instead of the traditional K-fold cross-validation.

At the end of the experiment, Random Forest was chosen as the best algorithm for *M. laevigata* with a final score of 0.945 and KNN was chosen for *M. ilicifolia* with a score of 0.963. Below, Table 5 summarizes the results of this experiment, with the mean scores returned by the GridSearchCV. As observed, the scores were still very high and could still indicate overfitting, so the next experiment aimed to test if these scores were overestimating the quality of the model.

3.4.4 Incorporating further data

To test for the overfitting hypothesis, *M. ilicifolia* sampler results obtained during a previous study were used as preliminary testing to verify how the *M. ilicifolia* model would perform on unseen data (Antunes et al., 2019). According to Jiang et al. (2020), LC-MS analysis suffers from instrument variations that can affect the signals detected by the equipment over time. These variations can include shifts in the electrospray process, changes in detector sensitivity,

Table 5: Mean scores returned by GridSearchCV

	Model 1: Identification of *M. ilicifolia*	**Model 2: Identification of *Mikania laevigata***
SVC		
MCC	0.953 ± 0.047	0.937 ± 0.033
F1	0.975 ± 0.025	0.968 ± 0.017
Random Forest		
MCC	0.950 ± 0.038	0.941 ± 0.028
F1	0.974 ± 0.020	0.969 ± 0.015
KNN		
MCC	0.963 ± 0.037	0.920 ± 0.019
F1	0.981 ± 0.019	0.959 ± 0.009

changes in ion transfer, etc.

Since all the samples were analyzed at a given moment, all in the same equipment, the model could be over-adjusting to classify samples in these specific conditions. As a result, since LC-MS equipment are subject to variations, samples analyzed at different times and conditions, or with different equipment, could be more challenging for the model to classify.

In addition, all samples were collected on a specific day, between September and December of 2021. Although no seasonal metabolic changes were detected for *M. ilicifolia* or *M. laevigata*, many studies indicate that their chemical composition changes in response to specific environmental events (Antunes et al., 2019; de Lazzari Almeida et al., 2017a; Reis, 2004; Yariwake et al., 2005; Flück, 1955). Hence, the model could also be over-adjusting to classify samples harvested in such specific environmental conditions, present in the harvest day, and providing the model with samples from different periods could expose such problems.

Therefore, the samples used for this preliminary testing were harvested from the same location (CPQBA), but in a different period, montly between November 2016 and October 2017, and were analyzed with the same equipment and chromatographic method, during the same period. For *M. laevigata*, this test was not performed since there were no previous samples analyzed using similar extraction and chromatographic methods.

These *M. ilicifolia* samples were converted and processed by XCMS using the same parameters of the training data, and the testing was performed using the model and parameters determined with GridSearchCV. After applying all necessary treatments to these samples, the MCC score obtained was 0.0.

This score confirmed the overfit for the *M. ilicifolia* model and, therefore, the next step was to retrain the model and add more variation into the training set. For this, the samples used for the previous preliminary test were included in the training, and other samples from the preliminary study were added to the validation and test folders.

With these additional samples, the model would have a more diverse set of examples to train, therefore, it would be able to generalize better. Table 6 clarifies the sample distribution for this experiment.

Table 6: Sample distribution for each group with date of collection and analysis

Species	**Train**	**Validation**	**Test**	**Date of collection and analysis**
Maytenus ilicifolia	384	96	120	September, December of 2021
Maytenus ilicifolia	76	20	24	November 2016 - October 2017
Maytenus aquifolium	189	48	63	September, December of 2021
Maytenus aquifolium	76	20	24	November 2016 - October 2017
Mikania glomerata	108	21	21	September, December of 2021
Mikania laevigata	87	28	36	September, December of 2021

The ideal scenario, in this case, would be to also add samples collected from other locations and analyzed with different LC-MS equipment, at different labs and research groups, so this variation between examples would be even higher. Due to logistic limitations, however, this was not performed in this study.

With the new additional data, the pipeline from preprocessing to training followed the same steps as before. First, the data was converted using MSConvert, and then, the XCMS parameters were optimized with IPO. For this optimization, the training data included the previous samples, and afterward, training and validation were processed, separately, using XCMS.

The model training was also performed using the same steps as before, using GridSearchCV and PredefinedSplit applying inter-dataset validation, then testing the SVC, Random Forest, and KNN algorithms. In this experiment, the model for *M. ilicifolia* with the best performance was the Random Forest with a final MCC score of 0.963, as shown in Table 7 with the mean score of each model tested, returned by the GridSearchCV.

Table 7: Mean scores returned by GridSearchCV for the *Maytenus* experiment with previous data

	Model 1: Identification of *M. ilicifolia*
	SVC
MCC	0.839 ± 0.026
F1	0.924 ± 0.010
	Random Forest
MCC	0.803 ± 0.177
F1	0.934 ± 0.056
	KNN
MCC	0.796 ± 0.126
F1	0.904 ± 0.055

As mentioned, the addition of new data to the *M. laevigata* model was not possible, as no previous samples were available to add to the training and validation sets. Therefore, *M. laevigata*'s final model, at this point, was also the Random Forest with an MCC score of 0.965, as mentioned before. However, for both models, the scores were still high, and the next steps involved normalizing the data and feature selection.

3.4.5 Normalization of the feature's relative intensities

After evaluating the results obtained from the previous experiments and the possible reasons behind the high scores, the normalization step was tested. As observed in the first MSTUS, especially for *M. laevigata* model, some samples belonging to the negative classes presented lower intensities and the reason for such differences could lie in the analytical methods.

As mentioned before, the analytical methods used herein were validated for their respective target species, and, therefore, samples with a very distinct chemical composition could present lower ion intensities due to a lower ionization. Therefore, it is possible that the models obtained so far were over-adjusting to classify the samples mostly based on the ion intensities, with the positive samples in each experiment presenting higher intensities and the negative samples presenting lower intensities.

According to Katajamaa e Orešič (2007) the goal of the normalization step in the context of metabolomics is to remove systematic bias in ion intensities while retaining the biological variation. According to Pedregosa et al. (2011), normalization is the process of scaling samples to have a unit norm. This means adjusting the values of a dataset so that the Euclidean norm (also known as the L2 norm) of each sample is equal to one.

Normalization ensures that the ion intensities of negative and positive classes within both experiments would be more comparable. Without normalization, as mentioned before, a model might classify samples based primarily on intensity differences, ignoring more meaningful patterns. By applying normalization, the focus shifts to the relative differences in the compound profiles rather than their absolute intensities, which could lead to a more robust classification result.

Therefore, before training, the data was normalized within the 'Pipeline' object, by using the Normalizer class, also from the scikit-learn package, which applies an l2 normalization to the data. by default. Afterward, the model training on this step was also performed using GridSearchCV and PredefinedSplit, with inter-dataset validation, testing the SVC, Random Forest, and KNN algorithms.

As mentioned before, Figures 9 and 10 show the MSTUS before and after normalization, and it is possible to observe how the normalization affected the distribution of the samples for both experiments. After normalization, the samples from different classes were more evenly distributed, and the impact of this process is observable in the final scores obtained.

After this experiment, the final score for the *Mikania* experiment was 0.947, also with the Random Forest selected as the best-performing model. For the *Maytenus* experiment, the final score was 0.956 and the best-performing model was obtained by KNN. In both cases, the scores dropped if compared with previous experiments. Table 8 illustrates the mean scores for both experiments and all models.

For the *Mikania* experiment, the mean scores for SVC dropped for the normalized data while for the *Maytenus* experiment, the drop was for the Random Forest algorithm. Therefore, it is noticeable that the normalization affected the overall score of the models, especially the best-performing models chosen by the GridsearchCV but the difference was not exacerbated and

Table 8: Mean scores returned by GridSearchCV on the normalized data for both experiments

	Model 1: Identification of *M. ilicifolia*	**Model 2: Identification of *M. laevigata***
SVC		
MCC	0.808 ± 0.122	0.919 ± 0.020
F1	0.912 ± 0.052	0.958 ± 0.010
Random Forest		
MCC	0.844 ± 0.050	0.945 ± 0.023
F1	0.924 ± 0.021	0.971 ± 0.012
KNN		
MCC	0.833 ± 0.107	0.922 ± 0.009
F1	0.925 ± 0.045	0.960 ± 0.005

the scores remained high. The next experiment involved selecting a subset of features, as a last attempt to generate more realistic scores and try to avoid any overfitting of the models.

3.4.6 Feature Selection

When working on plant metabolomics it is important to note the differences between primary and secondary metabolism. Plant primary metabolism is formed by a group of compounds such as carbohydrates, lipids, and proteins which are the basic components involved in fundamental processes such as plant growth, development, and reproduction. As a result, the primary metabolism presents many similarities among different plant species and these compounds can be detected in metabolomic analysis(van Dam e van der Meijden, 2011).

Secondary metabolism, on the other hand, is mostly responsible for the survival and interaction of the plants with their biotic environment, serving as a response, and defense mechanism. Consequently, different plant families, genera, species, and individuals often present many differences in metabolite profiles. The secondary metabolome is, therefore, often considered the 'fingerprint' of plant species (van Dam e van der Meijden, 2011).

Since the metabolomic analysis can capture compounds of the primary metabolism, which is common to plants in general, the grouping and filtering steps of XCMS are not enough to prune out features related to this metabolism. In addition, since LC-MS equipment presents high sensitivity, ions present in the extraction liquid, mobile phase, and general sample contaminants can also be detected.

According to Walker (2022), unnecessary features can lead to overfitting and generate a model/system that is computationally expensive. Therefore, a separate feature selection step needed to be implemented to ensure that only relevant features would be used for sample classification. For this, three methods were chosen: Mutual Information, Recursive Feature Elimination, and Boruta.

Mutual information is a univariate feature selection method that works by selecting features that perform better on a univariate statistical test. This method measures how much information one variable provides about another and is classified as a "filter method". In this context,

for example, if two variables are independent, the mutual information score would be 0. During feature selection, features with a higher score are kept, as they are the ones that provide the most information on the target variable. (Walker, 2022; Pedregosa et al., 2011)

Mutual information is a good starting point for feature selection, but most of the time we often deal with multivariable relationships between the features and the target. For this reason, other methods such as "wrapper methods" might be more efficient in selecting relevant features. Three common types of wrapper methods are the forward, backward and exhaustive methods (Walker, 2022).

While the forward feature selection recursively adds features to the training step if the score of the estimator improves, the backward feature selection removes the feature that affected the score negatively (Pedregosa et al., 2011). Their disadvantage is that the removed or added features are not re-evaluated afterward, even though their importance and impact on the model might change depending on the feature combination. A solution to this problem is applying other methods classified as "exhaustive feature selection methods" (Walker, 2022).

These methods evaluate a model on all possible combinations and select the best subset of features but at the cost of system resources and time. Wrapper methods in general, including the forward, backward, and exhaustive methods, all tax the system resources as they need to train the model at each iteration and the more complex an algorithm is, the more this is an issue. To solve this problem, Recursive Feature Elimination (RFE) can be applied (Walker, 2022).

RFE is also a wrapper feature selection method but uses the simplicity of filter methods, while providing better information, much as the other wrapper methods mentioned before. RFE works by removing the feature with the lowest importance measure and repeats this process until the best-fitting model is found. When a feature is removed, it receives a rank score reflecting the point at which it was removed (Walker, 2022). For scikit-learn's implementation, the feature ranking corresponds to the feature importance, and the best features are assigned a rank 1 (Pedregosa et al., 2011). The benefit of the RFE is that it is easier to train than exhaustive methods, but another popular method with a different and efficient approach is Boruta (Walker, 2022).

Boruta is a feature selection method that also presents similarities with wrapper methods and it was originally developed as an R package. For each feature, Boruta creates a 'shadow feature' with its original values shuffled. To evaluate a feature, it compares the information provided by the original and the 'shadow' version and gradually removes those that provided less information than their artificial counterparts. The final output classifies the features into confirmed, tentative, and rejected groups (Kursa e Rudnicki, 2010). The advantage of Boruta is that if a feature is selected, then it probably does provide information on the target. The problem, however, is that, just as the exhaustive methods, Boruta is also computationally expensive (Kursa e Rudnicki, 2010; Walker, 2022).

As mentioned before, in this study, Mutual Information, RFE, and Boruta were the chosen methods to help select the relevant features. Mutual Information was selected as an initial

filter method, RFE was chosen as it is midway between filter and wrapper methods, providing more input than Mutual Information but without taxing the system resources and Boruta was applied for its unique approach. Together, all three techniques provide a good overview of relevant features increasing the chances of finding a subset of features that can represent the data in a lower dimensionality.

When applying these methods to select the features for both models, for the *Maytenus* model, out of 306 features in total, 194 were selected by at least one of these methods. In contrast, for the *Mikania* model, out of 148 features in total, 115 were selected by at least one of the methods.

For the *Maytenus* model, Mutual Information and RFE selected 153 features, while Boruta selected 154. All three methods selected 156 unique features, and their Jaccard similarity was around 58%, with RFE and Boruta's methods being the most similar, with 80% similarity.

When comparing the feature sets for the *Maytenus* model with a previous study developed by the group, most of the features detected in the study were also selected by at least one of the methods (Antunes et al., 2019).

For the *Mikania* model, Mutual Information selected 74 features, RFE selected 103, and Boruta selected 98 features in total. All three methods had 97 unique features and their Jaccard similarity was around 56%. For this experiment, RFE and Boruta also present the highest similarity, around 76%.

Comparing these results with previous studies with *Mikania* species, Umbelliferone, Coumaric acid, Kaurenoic acid, Grandifloric acid, Chlorogenic acid, Dicaffeoylquinic acid, and Mellilotoside were possibly selected as important features by these methods. Even though it is not possible to identify the compounds detected in this study, it is possible to infer their presence based on their *m/z*, in comparison with previous data. These compounds were detected in previous studies from the research group that utilized the same equipment and method applied herein, so even with different retention times, their *m/z* and its relative position to other compounds were maintained (Borghi et al., 2020; Costa et al., 2018). Tables 14 and 15 on the Supplementary Material section illustrate the features selected for each method, for both experiments.

Another way to evaluate the feature selection methods is by analyzing the data in a reduced dimensionality space by applying the Uniform Manifold Approximation and Projection (UMAP) technique. UMAP, like any dimensionality reduction technique, transforms the data into a lower-dimensional representation while attempting to preserve its essential topological structure. According to the creators of the method, "UMAP is a flexible non-linear dimension reduction algorithm based on manifold learning techniques and ideas from topological data analysis" (McInnes et al., 2020).

Since feature selection methods aim to identify a subset of the relevant features from the original set, the impact of this reduction can be visualized in a plot such as UMAP. Ascensión et al. (2022) for instance, applied UMAP to evaluate their feature selection techniques and the clusters formed by UMAP. In their work, to assess the quality of a feature selection represen-

tation on UMAP plot they verified whether different groups appeared as different clusters on UMAP. If two different groups appear mixed within UMAP they concluded that some of the important features to differentiate the groups were not selected by the method.

In the present work, a similar approach was applied. Comparisons between the feature selection representation on UMAP versus the original set representation were made to evaluate the feature selection methods. If the clusters were maintained, the feature selection method was considered efficient, if not, it was concluded that the method changed the topological structure of the data and, therefore, another iteration was done, especially on the wrapper methods. Figures 16 and 17 illustrate the UMAP plots on the original and reduced feature spaces for each feature selection method.

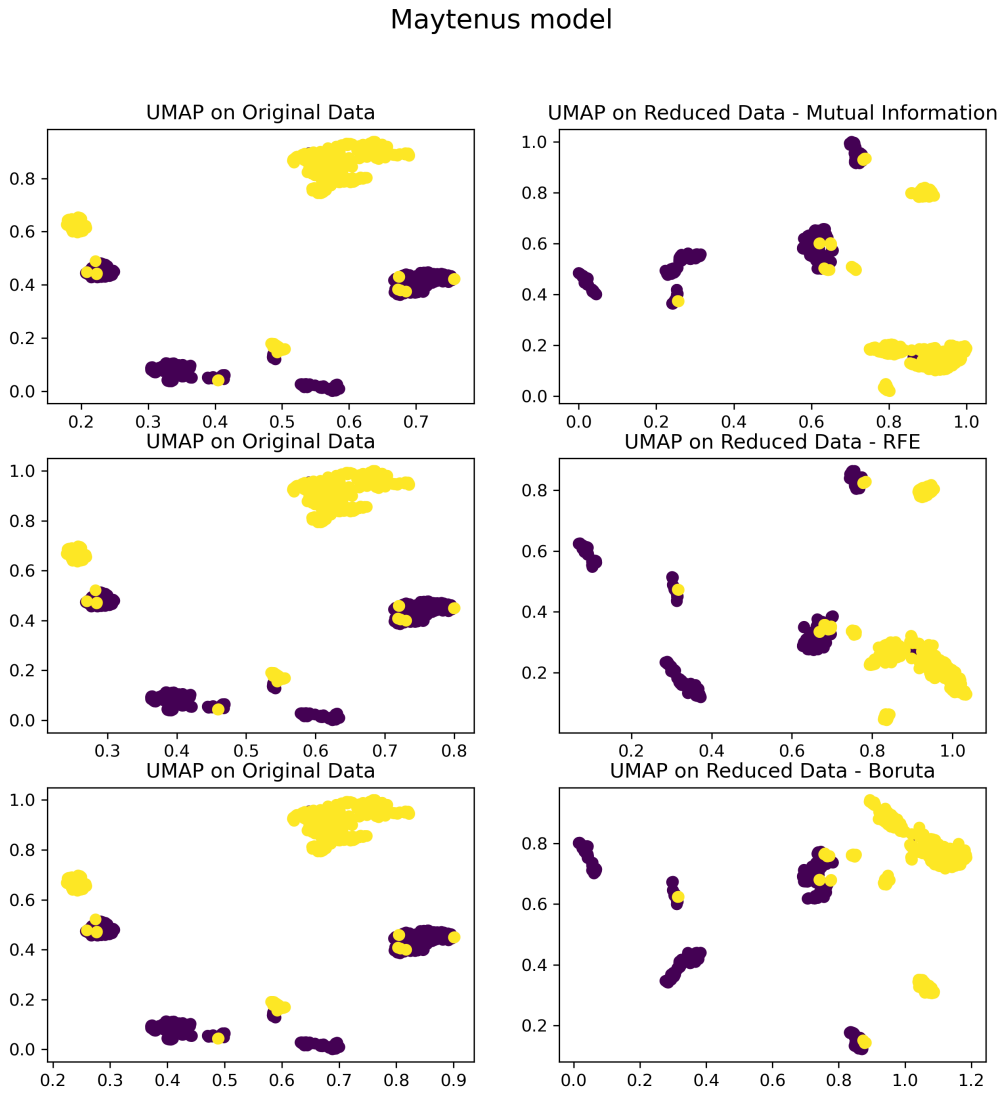

Figure 16: UMAP on original and reduced *Maytenus* data for each feature selection method.

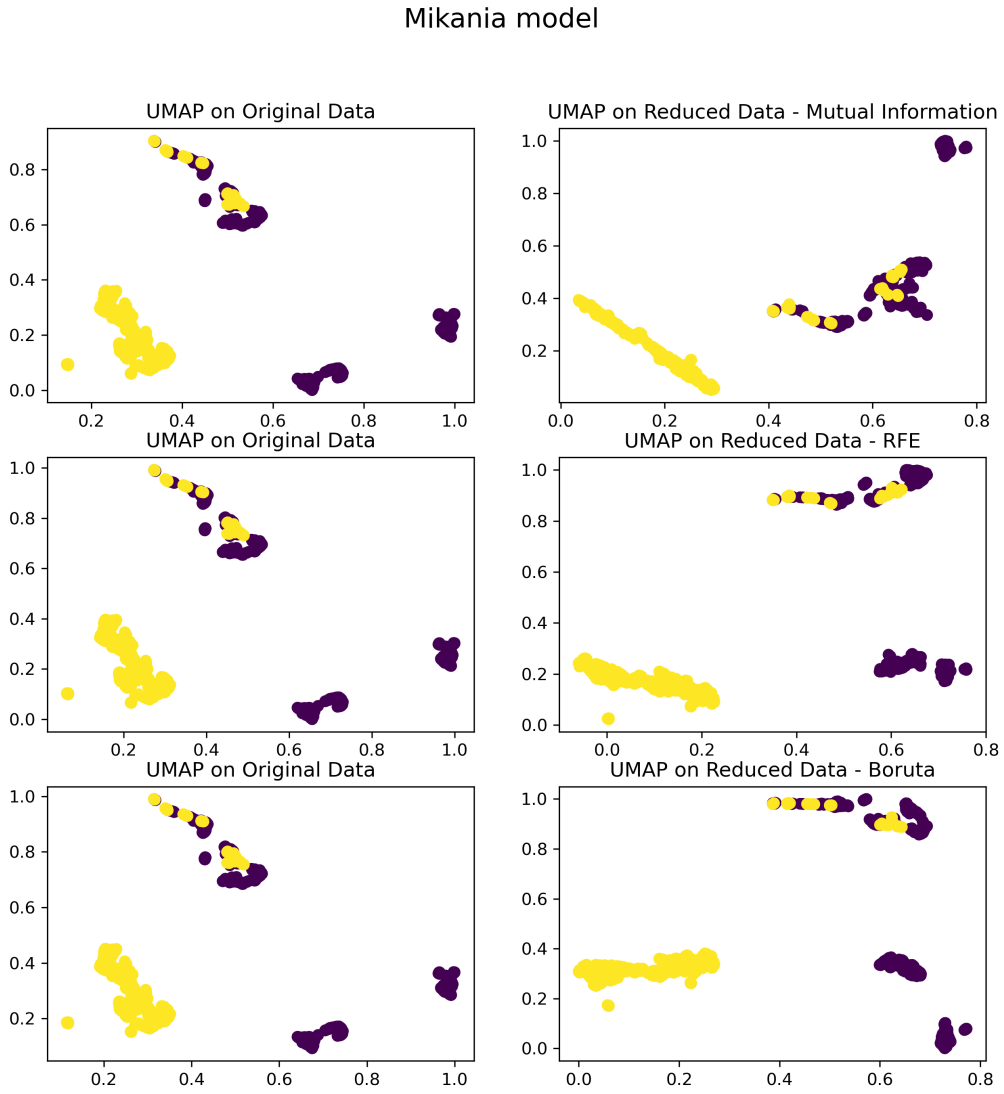

Figure 17: UMAP on original and reduced *Mikania* data for each feature selection method

By analyzing the plots for *Maytenus* it is possible to observe that the groups present in the original data set were maintained, with a slight change in shape and orientation of the groups within the plot 16. For *Mikania*, the shapes of the groups changed more drastically but the data points remained grouped in the same manner 17.

Therefore, the UMAP plot for both species confirmed that the feature selection methods were able to select a subset of the features without changing the overall characteristics of the dataset, as both spaces, original and reduced, still present a lot of similarities and the same clusters. All methods, after iterations, presented similar results on the UMAP plot, with no method being substantially worse or better in reducing the dimensionality space without information loss.

However, by only evaluating the feature selection methods with UMAP, it is impossible to predict which methods would better suit the model training and yield a better model. Therefore, applying the feature selection methods on the GridSearchCV is a good strategy as they would be evaluated alongside all other model parameters. Hence, this was the next experiment applied, alongside the comparison of different validation techniques.

3.4.7 Model Robustness and Validation Methods

During the model training and validation, two approaches are usually taken to evaluate the model: intra-dataset validation or inter-dataset (or cross-dataset) validation. Intra-dataset validation is done by evaluating the model in a set taken from the training data, while the inter-dataset is the evaluation performed on a separate dataset, in which the model had no previous contact. (Nadimpalli e Rattani, 2022)

Traditionally, intra-dataset validation is more commonly applied, usually with the aid of a k-fold cross-validation technique, in the basic approach (Pedregosa et al., 2011). For this technique, the training set is divided into k smaller sets and, at each iteration of the model training, a different set (fold) is used for validation while the remaining are used for training. The score reported is the average of the values obtained in all iterations (Pedregosa et al., 2011).

However, a good number of studies already demonstrated that a higher performance can be achieved with this validation method, but at the cost of a poor generalization across datasets (Nadimpalli e Rattani, 2022; Mohammadi et al., 2020). Additionally, according to Huang e Zhang (2021), inter-dataset evaluation is a more challenging task and has not been regularly applied in statistics or learning based papers, even though it presents a more realistic scenario.

For the present work, as mentioned, after some initial experiments, both training and validation sets were separated before XCMS preprocessing. Since both datasets were completely separate, during hyperparameter tuning using the GridsearchCV, the Predefined split function had to be applied during cross-validation, and as a result, an inter-dataset validation technique was used.

In the final steps of model training, however, as was done by Huang e Zhang (2021), a comparison between intra-dataset and inter-dataset validation was performed to verify the model's robustness. The goal was to compare the model's performances for both validation techniques

and verify how they differed from each other.

Therefore, three experiments of model training and validation were performed for each model. One intra-dataset validation was performed with the training set, another with the validation set, and the final training was performed using the inter-dataset validation, with the training set and validation sets used in their respective roles, all within the GridsearchCV.

Table 9 and 10 depict the mean scores and standard deviations between intra-dataset and inter-dataset validation techniques for both models. As observed, the standard deviation between the techniques was very low, for both scoring functions, which demonstrates that the model obtained in the inter-dataset validation technique is robust. If the standard deviation were high, it would indicate the model over-adjusting to specific datasets, which was not the case herein. The technique chosen to continue the experiments was the inter-dataset validation as it yields models with good generalization, as mentioned before.

Table 9: Mean scores +- standard deviation for all three validation techniques and each algorithm for the *Maytenus* model

Model	**MCC**	**F1**
SVC	0.954 +- 0.047	0.975 +- 0.026
RFE	0.925 +- 0.052	0.962 +- 0.026
KNN	0.958 +- 0.037	0.978 +- 0.019

Table 10: Mean scores +- standard deviation for all three validation techniques and each algorithm for the *Mikania* model

Model	**MCC**	**F1**
SVC	0.768 +- 0.170	0.858 +- 0.122
RFE	0.924 +- 0.028	0.956 +- 0.014
KNN	0.905 +- 0.028	0.947 +- 0.017

After verifying the model's robustness by comparing the different validation methods, the next step involved using this verification along with the GridSearchCV and Validation curves to obtain the final model. So far, three feature selection and validation methods have been applied but, as mentioned before, there was no guarantee that the hyperparameter space tested on the grid would yield the best-fitting models for both species. Therefore, a final implementation into the code was the creation of validation plots, to keep track of the hyperparameters tested on Gridsearch.

3.5 Final Model

According to Pedregosa et al. (2011), hyperparameters are parameters that are not learned in the estimators. They are passed as arguments to the algorithm function implementation on scikit-learn and control how the model is trained. Some examples include "C", for Support Vector Classification, "model_weights" and "n_neightbors" for K-Nearest Neighbors, and "min_samples_leaf" and "n_estimators" for Random Forests, which are some of the hyperpa-

rameters tested on the present study.

According to Arnold et al. (2023) hyperparameters critically affect the model's performance. As a result, to develop a robust model, is it essential to search for the best hyperparameter settings (Hoque e Aljamaan, 2021). Hoque e Aljamaan (2021), for example, demonstrated the impact of hyperparameter tuning on forecasting models, and concluded that this step could be the best choice for improving the model's performance without overfitting.

There are multiple techniques to search the hyperparameter space for the ideal setting (Hoque e Aljamaan, 2021). The classical method is the grid search, implemented on scikit-learn by the GridSearchCV function. The grid search algorithm exhaustively searches for all possible combinations of hyperparameters, intending to improve a specific performance metric, selected by the user. (Pedregosa et al., 2011; Hoque e Aljamaan, 2021). Once a hyperparameter setting is found, there is a certainty that, from that hyperparameter space, the best combination to yield the best-performing model was found.

The disadvantage of the grid search is that since it searches for all possible combinations if the hyperparameter space to be tested is too large, the exhaustive search will take too long. An alternative technique is called Random Search, implemented on scikit-learn under the function RandomSearchCV. This technique searches the hyperparameter combinations at random and as opposed to grid search, however, it might not find the best settings for that specific hyperparameter space (Bergstra e Bengio, 2012).

In the present work, as mentioned, the hyperparameter tuning technique of choice was the grid search. According to Liashchynskyi e Liashchynskyi (2019), grid search is still state of the art as it supports parallelization, it finds the best combinations of hyperparameters and if both the dataset and hyperparameter spaces are not too large, it presents a good performance.

To help in the process of testing different hyperparameter spaces, without inputting too many options at once, a good supporting strategy is to plot the validation curve. The validation curve is a plot that shows the influence of a single hyperparameter on the training and validation scores, to determine if the hyperparameter value is causing the model to overfit or underfit (Pedregosa et al., 2011). The hyperparameter value is plotted on the X-axis while the score is plotted on the Y-axis.

If both the training and validation scores are low, the estimator is underfitting. If the training score is high but the validation score is low, the estimator is overfitting. And, finally, if both scores are high and similar, with the validation only a bit lower, since a higher validation score is usually not possible, the estimator is performing well (Pedregosa et al., 2011).

Figure 18 and 19 show the validation curve for both *Maytenus* and *Mikania* models and all hyperparameters tested. For almost all the numeric hyperparameters a plateau on the MCC score was achieved, which demonstrates that larger values would possibly not increase the final score. For the hyperparameters "C" and "min_samples_leaf" for the SVC and Random Forest algorithms, respectively, as their values increased, the validation score also increased, until it reached a constant value.

If the validation score decreased too suddenly, this would be an indication of overfitting,

while if it increased higher than the training score, it would indicate underfitting of the model. Since, as mentioned, the plateau was reached, trying different values would probably not yield a better score and so the iteration with GridsearchCV could be halted and the best-fitting model found at this iteration would more certainly be the best one.

Hence, as mentioned before, a lot of iterations were done in the final code to achieve the final model. With the help of the UMAP plots, small modifications were made to the wrapper feature selection methods to achieve the best dimensionality reduction without information loss. The different validation methods were also used to keep track of the model robustness, aiming at lower standard deviations and analyzing which features and algorithms were chosen each time. Finally, the validation curves were also plotted to help determine when the best fitting model was found.

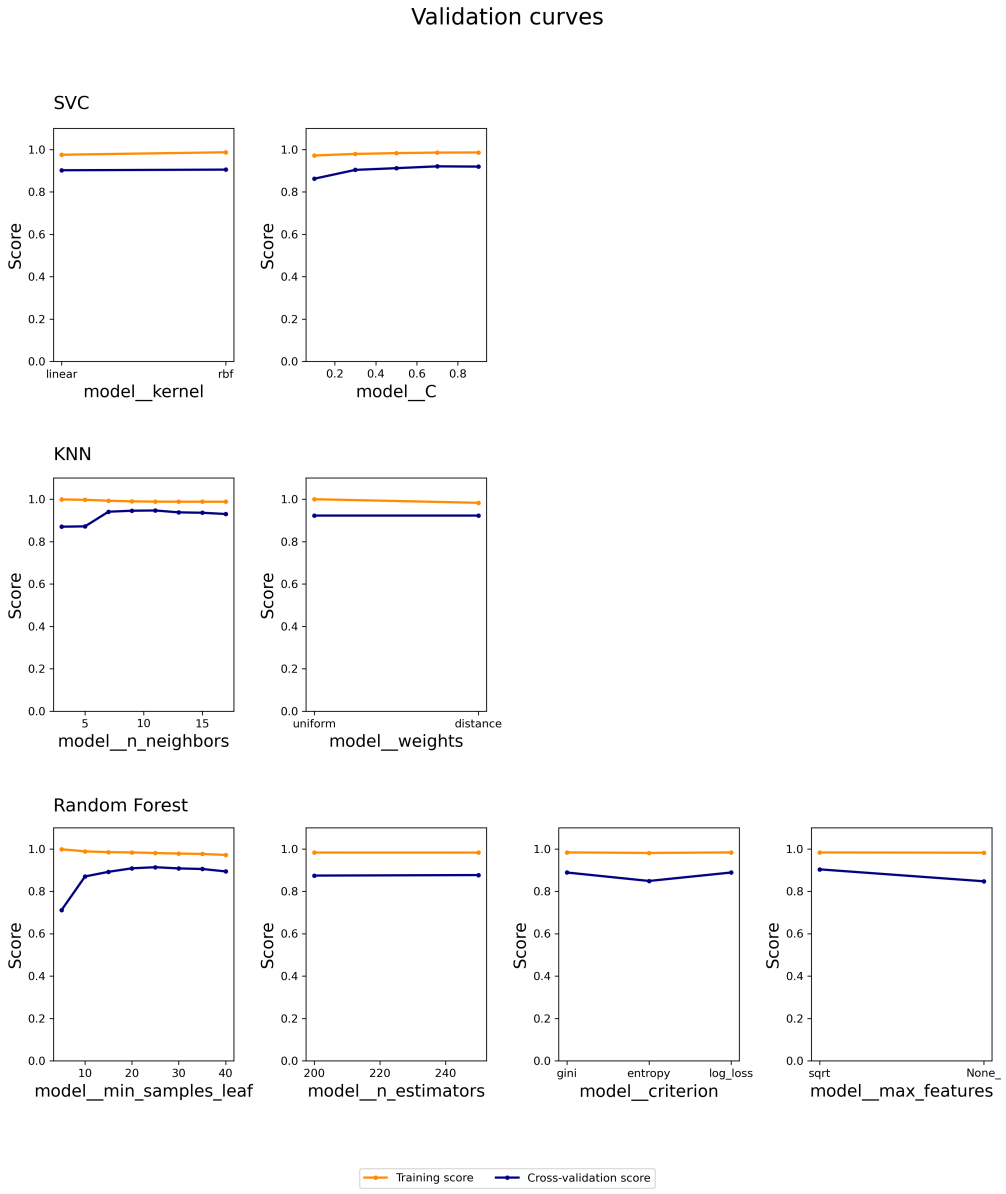

Figure 18: Validation Curves for *Maytenus* model

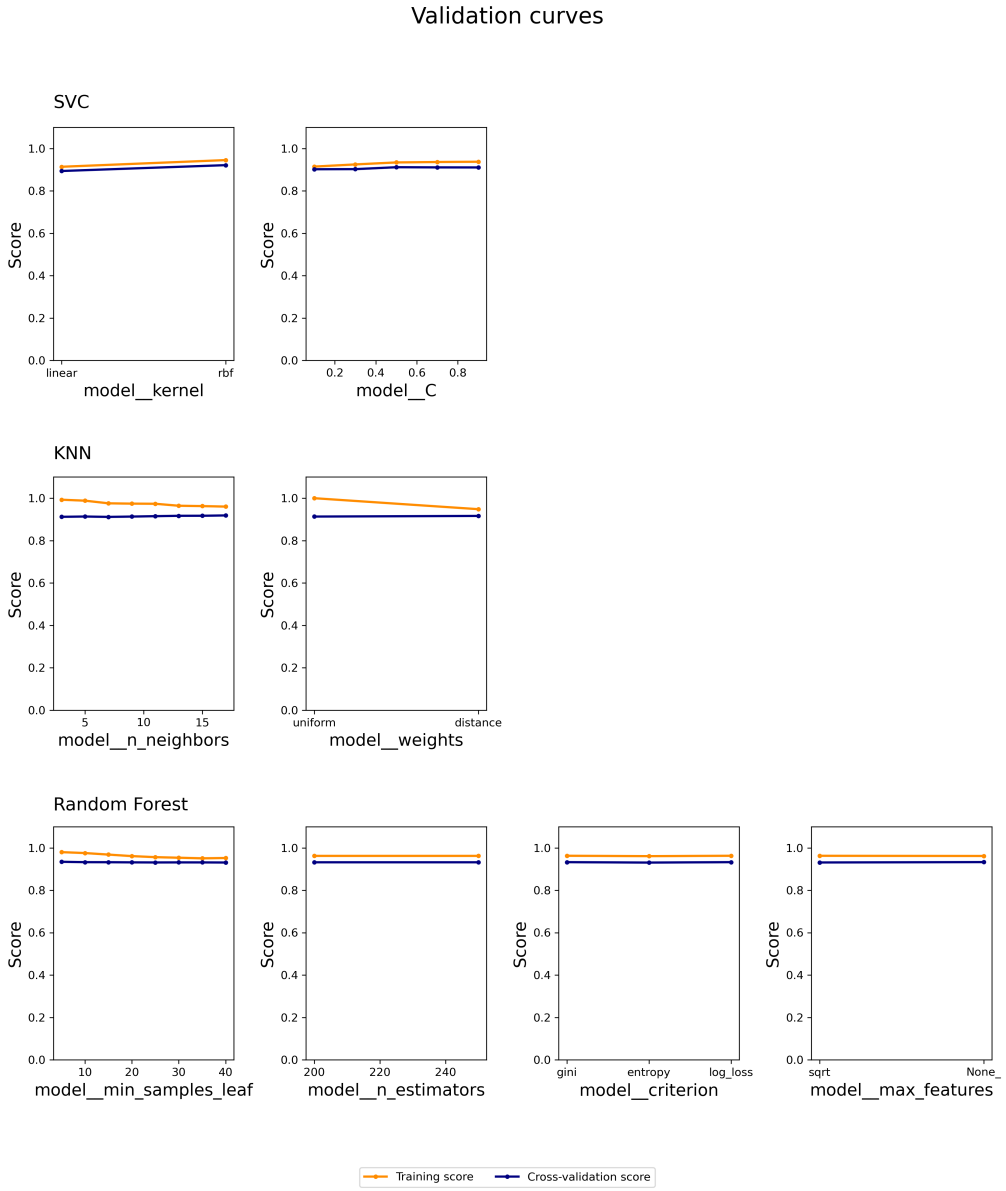

Figure 19: Validation Curves for *Mikania* model

Table 11 shows the mean scores for each model, at the last iteration of Gridsearch, with all the implementations mentioned herein. The final model for both target species still presented high scores but with all the experiments and improvements implemented, the confidence of the score obtained was also higher. For *Maytenus*, the best-fitting algorithm was KNN with a final score of 0.959. For *Mikania*, the best-fitting algorithm was the Random Forest with a final score of 0.942. For both models, the feature selection method that yielded these best-performing models was Mutual information.

Table 11: Mean scores returned by GridSearchCV for the final experiment

	Model 1: Identification of *M. ilicifolia*	**Model 2: Identification of *M. laevigata***
SVC		
MCC	0.904	0.906
F1	0.946	0.952
Random Forest		
MCC	0.874	0.932
F1	0.937	0.964
KNN		
MCC	0.922	0.915
F1	0.961	0.956

3.6 Final Model Analysis

According to Altmann et al. (2010), there are two possible goals or outocomes when applying machine learning to a research field: generating a model (possibly a black box model, with no interpretability) or generating insights into how the predictive features impact the variable of interest. The second task of feature discovery or ranking is, according to the authors "the essence of biomarker discovery in bioinformatics and life sciences".

Even though the focus of the present work is creating a model able to classify samples and aid the quality control of *M. ilicifolia* and *M. aquifolium*, the algorithms applied herein are not entirely considered black-box models, which means that interpreting their results is still possible. Therefore, the permutation and feature importance were extracted from both to understand which features were the most relevant in each model. The code to obtain the permutation score and feature importance are present in the Supplementary Material.

Since the best-performing model for *Maytenus* was the KNN, and this implementation on scikit-learn does not have a feature importance attribute, the permutation importance function was used to extract which features contributed the most to the final model.

To calculate the permutation importance, first, the model score is obtained on the features used on the estimator. Then, the feature values are permuted and the model's performance is recorded. This is done for every feature and can be repeated multiple times. The permutation score is the difference between the baseline metric and the metric after permutation. The higher this value, the higher the importance of that given feature on the estimator, as the permuted values had a higher effect on the model's score (Pedregosa et al., 2011).

The permutation score for the *Maytenus* model was calculated both on the train and test sets. Figures 20 and 21 show the top features for each set, and their permutation scores, with mean and standard deviation for each feature. All permutation importance values are present in the Supplementary Material (Table 16).

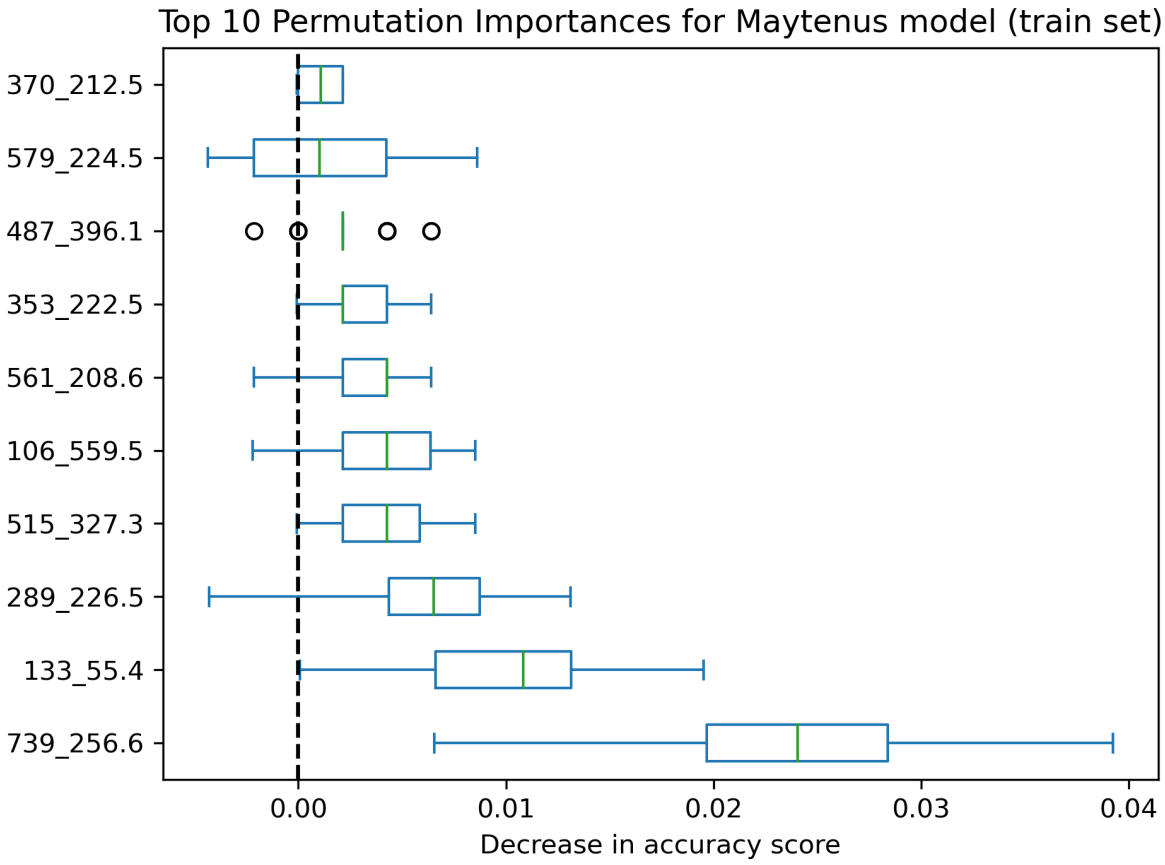

Figure 20: Permutation Importance values for the *Maytenus* model calculated on the training set. The box and whiskers are the distribution of the scores obtained in each iteration. On the y-axis, the top 10 most important features

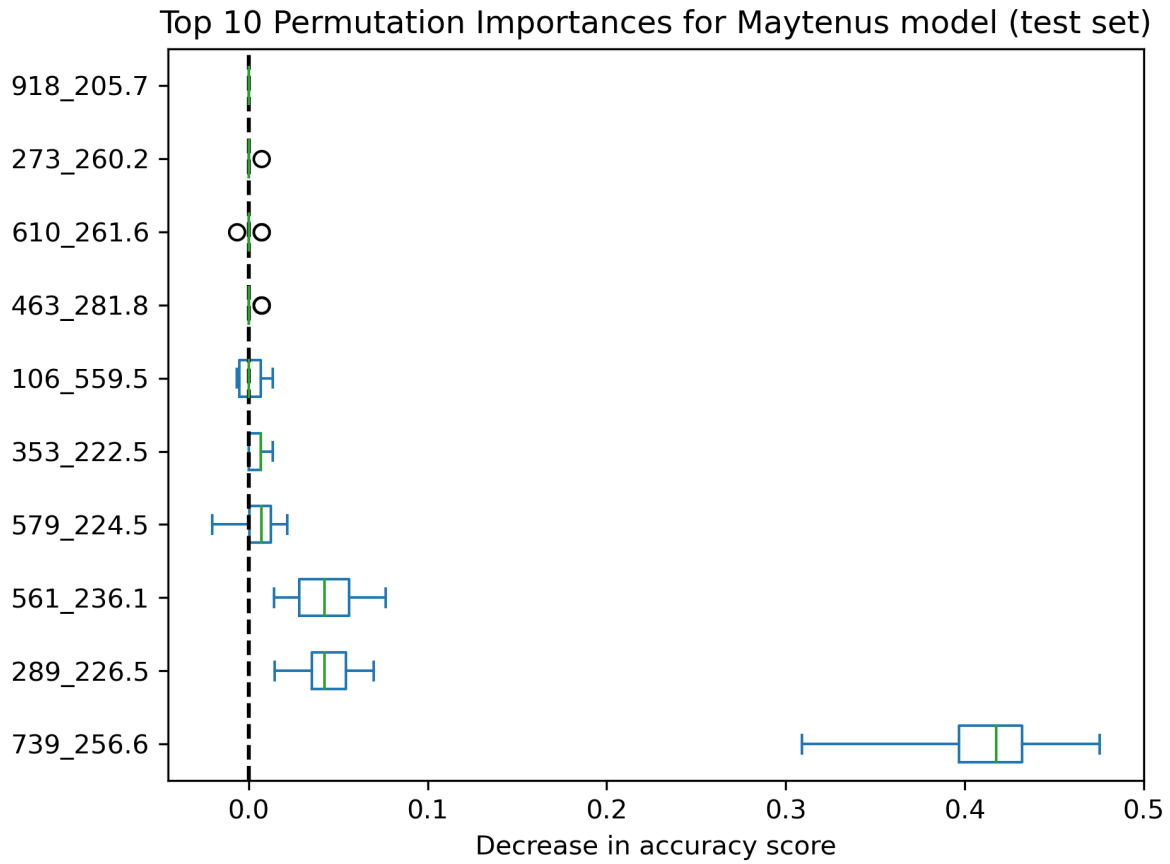

Figure 21: Permutation Importance for the *Maytenus* model calculated on the test set. The The box and whiskers are the distribution of the scores obtained in each iteration. On the y-axis, the top 10 most important features

By analyzing both the table and the plots it is interesting to notice that features which impacted most of the grouping, when analyzing the test set, were *m/z* 739, 289, 561, and 579 which were *m/z* values detected for *M. ilicifolia* both on this and previous studies. For the training set, the *m/z* 739 was also the most important with *m/z* 289 as the third on this ranking. Other features on the top of this ranking were also detected on *Mikania* species, also presenting higher importance for their model (*m/z* 133, 353, and 515, for instance)

Additionally, the permutation standard deviation on the training set was much higher than for the test set, but the decrease in the accuracy score was lower. This indicates that the result observed on the test set is much more consistent and the feature ranking for the test set could be more trustworthy.

When comparing these features with those detected in previous studies, as mentioned before, some of them are noteworthy Catechin and/or Epicatechin (m/z 289) and a triglycosylated flavonoid (m/z 739, formed by a Rhamnose- Rhamnose- Glucose-Kaempferol), for example, were considered some of the most important features for the *Maytenus* model, by the permutation importance on both datasets.

Catechin or Epicatechin, as mentioned before, are considered the chemical markers of *M.*

ilicifolia and usually serve as a standard for the quality control of medicinal herbs based on this species. This compound, however, was not the most important feature for the models, as the *m/z* 739 presented a much higher score.

This compound, which possibly represents a triglycosylated flavonoid, was also observed in high amounts in *M. ilicifolia* in previous studies. When analyzing the permutation importance calculated over the test set, this feature presented a higher score compared to others, followed by the *m/z* 289 and 561. This reinforces the need to use other compounds in addition to the chemical markers when testing the quality of a given herbal product. Such compounds could be extremely useful for herbal drug identification and quality control, as their isolated chemical compounds are not commercially available.

Both *m/z* 289 and 739, belonging to the flavonoids group, could contribute to the plant's adaptation to the environment and can also provide interesting pharmacological properties to the plant. According to a comprehensive review by Salam et al. (2023), various environmental factors can mediate the synthesis of flavonoids, which play a defensive role due to their antioxidant properties.

Flavonoids can be involved in the plant's response to water deficit and UV stress, by conversing the osmotic potential of the plant cells and accumulating on the leaf, creating a ticker epidermal layer that protects the plant from harmful radiation. Additionally, flavonoids have been associated with adaptations to heavy metal stress, low temperature, and nutrient stress as well (Salam et al., 2023).

Furthermore, flavonoids have demonstrated antifungal, antibacterial, antiviral, antimutagenic, antioxidant, and anti-inflammatory effects (Elshafie et al., 2023) . The antioxidant effect has even been associated with the gastroprotective properties of *Maytenus*, according to Suzuki et al. (2011).

Other features with high permutation scores obtained such as *m/z* 133, 515, and 353 were also observed to be important for the *Mikania* model and were detected in previous studies with *Mikania* samples. This indicates that the model used features from the negative class (*Maytenus* and *M. glomerata*) to aid the sample classification, which could cause the model to overfit. In this case, the model could perform better in a specific situation in which samples from *Mikania* and *Maytenus* were analyzed and tested together.

For the *Mikania* model, since the best-performing model was the Random Forest, it was possible to extract the feature importance via the feature_importances_ attribute. In this attribute, the importance of a feature is the total reduction of the chosen criterion measure brought by such a feature. If the criterion measure, for instance, was Gini impurity, the most important features are those that reduced this impurity the most. This calculation is done on the training set. Therefore, the higher the value, the more important the feature (Pedregosa et al., 2011).

Figures 22 and 23 show the top 10 feature and permutation importance scores for the *Mikania* model, respectively. Importance values obtained with both methods, for all features are present in the Supplementary Material (Table 17).

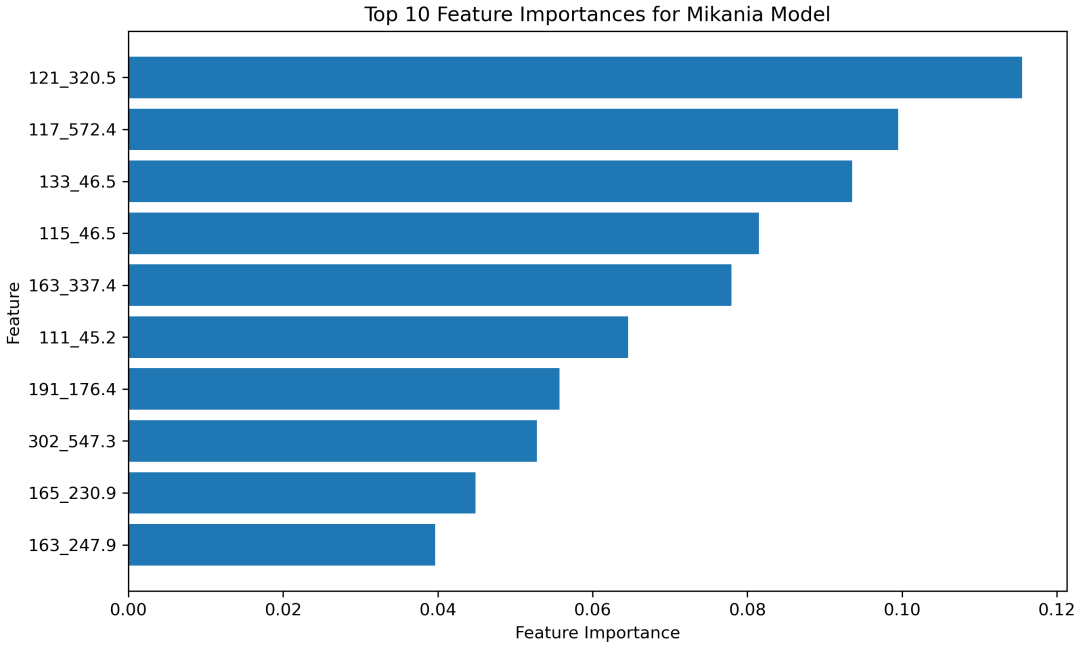

Figure 22: Top 10 most important features for *Mikania* model according to the feature importances attribute of RandomForestClassifier

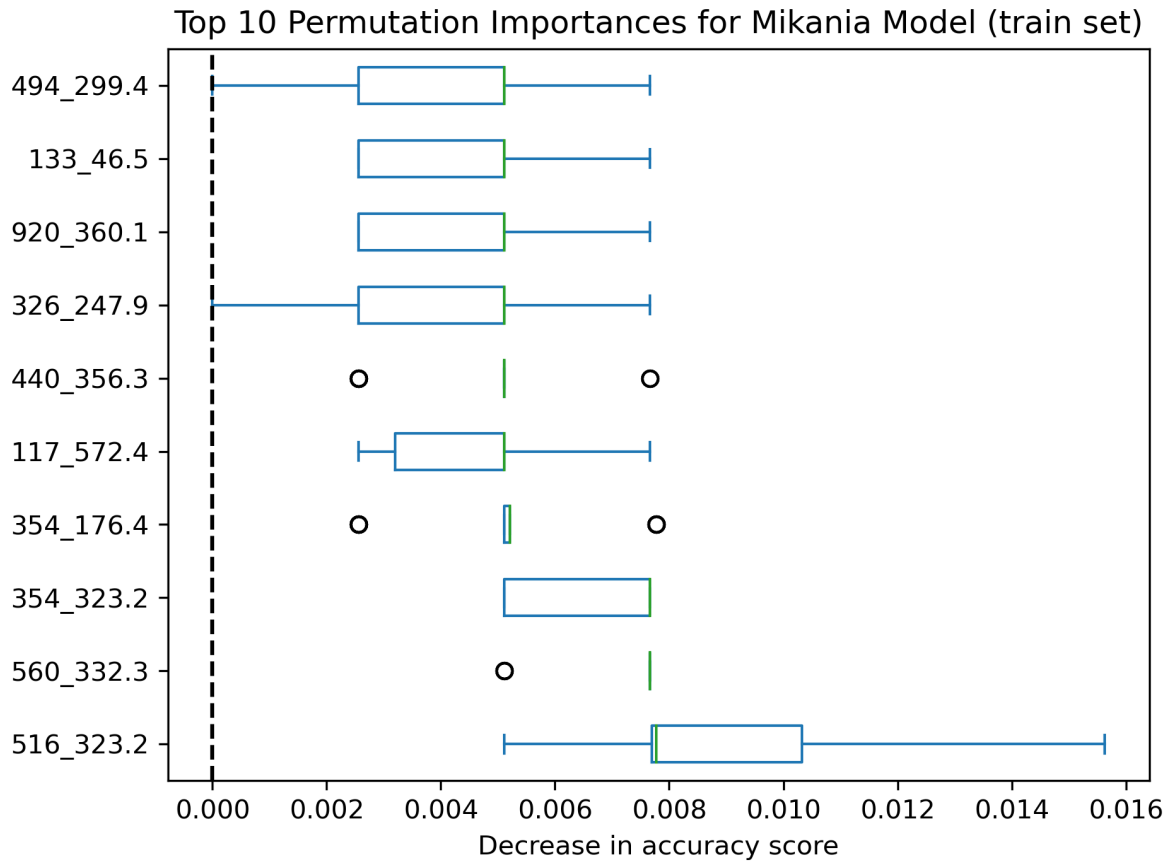

Figure 23: Top 10 permutation importances for *Mikania* model on the training set. The box and whiskers are the distribution of the scores obtained in each iteration. On the y-axis, the top 10 most important features

As mentioned before and similar to the *Maytenus* model, some of the most important features obtained by permutation and feature importance were also observed in other studies. The *m/z* values of 302, 163, and 516, for instance, could be tentatively identified as Kaurenoic acid, Coumaric acid, and Dicaffeoylquinic acid, respectively, and were also detected by Borghi et al. (2020) and Costa et al. (2018).

Even though the kaurenoic acid and dicaffeoylquinic acid have *m/z* values of 301 and 515, respectively, given the *m/z* rounding done by XCMS and all the preprocessing steps done previously to model training, a difference of around 2 *m/z* is acceptable to consider them the same values and compounds.

From these ions, the *m/z* 301, was tentatively identified as Kaurenoic acid, and *m/z* 163 was tentatively identified as Coumaric acid. Both were considered important features by the feature importance method, while Dicaffeoylquinic acid (*m/z* 515) was selected by the permutation importance test. Other features could not be tentatively identified.

As observed, both methods returned very different results and this could be due to the bias of the impurity-based feature importance methods, which can be misleading for high cardinality features, with multiple unique values, which is the case for this study (Pedregosa et al.,

2011).

According to Pedregosa et al. (2011) and Strobl et al. (2007), the impurity-based feature importance methods might not be reliable in situations where the variables vary in scale or present multiple categories. This, however, is often the case in genomics, bioinformatics, and related disciplines, such as metabolomics. In the present study, the variables were measured as relative abundances and therefore, they presented a high cardinality (unique values).

Taking this into account, the permutation importance method could present more reliable results, especially when looking at the permutation importance of the test set. While the feature importance measure is obtained in the training set, and might not reflect the ability of a feature to be useful for the model's predictions and generalize to the test set, the permutation importance can be performed on both sets.

Figure 24 shows the permutation importance calculated over the test set. For this test, it's possible to observe two features (*m/z* 516 and 326) were in common with the result of the previous permutation importance test, giving more confidence in the importance of both features for the *Mikania* model.

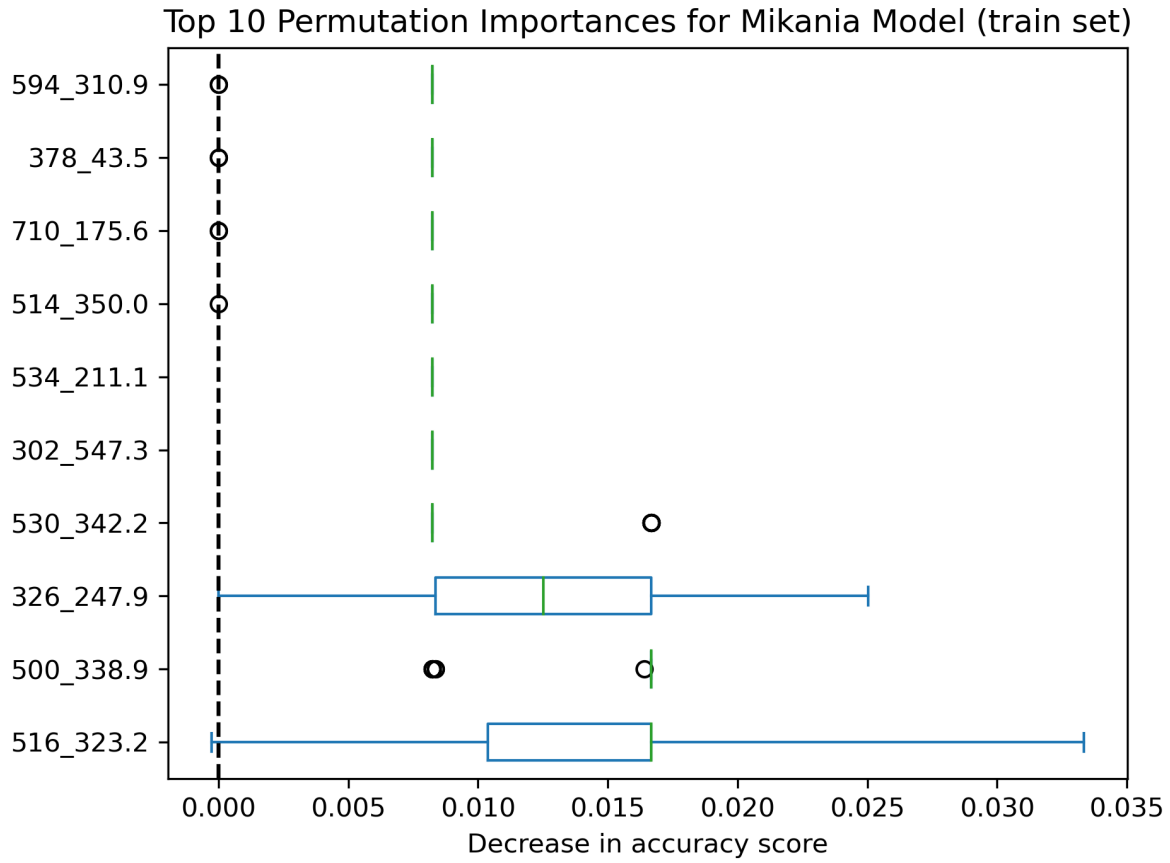

Figure 24: Top 10 permutation importances for *Mikania* model on test set. The box and whiskers are the distribution of the scores obtained in each iteration. On the y-axis, the top 10 most important features

Finally, by analyzing the most important features obtained by both methods it is possible to notice that for this model as well, no single feature was responsible for the model's entire performance. This also indicates the importance of analyzing multiple compounds when performing the quality control of herbal medicines, reducing the possibility of intentional or unintentional adulteration going undetected.

As mentioned before, if a chemical marker was the best parameter to attest to the quality control of herbal medicines, one single feature would present a much higher importance score by such methods. Since this was not the case, is possible to conclude that chemical marker analysis is not enough to guarantee the quality of herbal medicines, which is the main argument of the present study in favor of applying other quality control techniques. In other words, a general profile permits a more trustworthy plant identification method.

3.7 Final Model Testing

The last step to verify the model's performance is to test the model on completely unseen data called 'test set' (Géron, 2019; Walker, 2022). As mentioned before, the initial step taken within this study was splitting the data into three folders: train, validation, and test, for both models. To develop the final model only the train and validation folders were used.

To test the model the test set passed through the whole processing and machine learning pipeline, as mentioned before. The first step was to convert the data into mzXML using the MSConvert code present in the Supplementary Material section. Next, the data was processed using the relevant XCMS script obtained with IPO and the training set. The script used is also provided in the Supplementary Material section. Finally, the data passed through the machine learning Pipeline object, where it was normalized, the features were selected and the classification was applied.

The final score obtained for the *Maytenus* model was 0.938, while the final score for the *Mikania* model was 0.975. As mentioned before, the training scores obtained by both models on all experiments were still high, but the data processing steps taken before this final testing guaranteed a good performance of the model on unseen data. Tables 12 and 13 represent the confusion matrices for both models.

Table 12: Confusion matrix for *Maytenus* model. The number 1 represents the positive class, therefore, only samples from *M. ilicifolia*, 0 represents the negative class and any other species would fall into this class.

		Actual	
		1	0
Predicted	1	137	2
	0	7	142

As observed, there was a slightly bigger challenge to classify some samples from the positive class, for both models, in which seven samples from *M. ilicifolia* species and two samples from the *M. laevigata* species were classified into their respective negative groups. In terms of

Table 13: Confusion matrix for *Mikania* model. The number 1 represents the positive class, therefore, only samples from *M. laevigata*, 0 represents the negative class and any other species would fall into this class.

		Actual	
		1	0
Predicted	1	118	1
	0	2	119

quality control, classifying a 'good' sample as an adulterator is safer than classifying adulterator samples as belonging to the official and accepted species.

A similar misclassification issue was also observed with the models developed by Kharyuk et al. (2018). In their study, up to 30% of samples were misclassified on some algorithms, especially in cases where the differences between the species were subtle. Such issues might have occurred due to the objective of their model, which was to identify a species among a large group of options. In their case, to create such a model, the metabolomic method had to be generic enough to analyze different samples, and therefore, sutle differences between samples might not be captured. As a result, the data used to train their model would lack specific compounds that could differentiate two similar samples. In other words, the model developed by the authors was not as specialized as the one created herein.

In the present study, misclassification was much lower as the purpose of the model was to be efficient in distinguishing a target species from their counterparts, with similar composition. . These results reinforce the importance of creating specialized quality control methods for each species, instead of one universal method. Even though the latter would be much more practical, creating an LC-MS method applicable to a wide range of different samples is a challenging task. This might cause issues like those observed by Kharyuk et al. (2018), where the method might not be efficient in differentiating the target sample from their counterparts used as adulterators, which is the most important task in quality control.

However, even though the final models obtained herein had high scores, this study presents some limitations. The first limitation is that all samples were obtained from the same location and were analyzed in the same equipment, within the same laboratory. Additionally, for the *M. laevigata* model, all samples were also harvested on the same day.

For these reasons, it is possible that such high scores would only be obtained for samples in similar situations. One way to verify such an assumption would be to test the present models on samples harvested and analyzed in other locations and using different equipment.

Another limitation of this work is the analytical method itself. The LC-MS methods used herein were optimized for each target species specifically, to detect the highest possible number of compounds possible. Like the issue mentioned before, the score obtained could be the result of the model over-adjusting to specific analytical conditions. It's possible that if the samples were analyzed with different analytical methods, the final score would be much lower.

To improve the work in this regard, it could be possible to develop a method in which the Machine Learning model would be able to classify the samples correctly. However, if this study was expanded to all Brazilian herbal medicines and other models were created for each species, this unified method could become progressively harder to develop. Therefore it seems more applicable to develop one method for each species of medicinal plant.

Finally, it is also possible that both models need to be retrained from time to time. Since the objective of the models is to classify samples derived from living organisms, it is possible that the species' chemical composition changes over time, and metabolic patterns used by the models to classify the samples become less distinguishable.

As mentioned before, numerous studies have demonstrated how plants react to environmental changes (Eckardt et al., 2022). Since the overall global temperature has been gradually increasing and changes in the climate have been observed (Harvey et al., 2023; Eckardt et al., 2022), it is possible that the species will modify their chemical composition to better adapt to these conditions and, therefore, the models would need to be retrained if their efficiency dropped over time.

3.8 Machine Learning Application Proof-of-concept

As mentioned before, the present study aims to create a system for quality control of medicinal plants combining Metabolomics and Machine Learning techniques. To achieve this goal, a complex pipeline of data preprocessing and Machine Learning classification was created.

At the current state, if a user desired to test the solution presented herein on real samples, the necessary elements to achieve this objective would be the mzXML code to transform the raw data into mzXML format, the XCMS R script to process the data and generate the features table, the Python script to process the data and generate the feature names, normalize the data, select the relevant features and run the machine learning prediction. All these scripts would require the necessary software installations, the correct packages and library versions, and the expertise to use these programming languages.

To simplify and democratize this process, an application was developed to wrap around the whole metabolomics and machine learning pipeline. With a user-friendly application, any user would be able to test the system created in this study without much programming expertise or the need to install and configure their system.

Additionally, creating an application and a virtual environment with all necessary configurations would enable reproducibility. According to (Strangfeld, 2022), "a computational environment is a representation of everything that can influence computations done inside them [...] including computer hardware, operating systems, and software libraries". By providing access to an application with the computational environment kept constant, reproducibility is guaranteed.

To create this proof-of-concept the Python library Streamlit was used. Streamlit is a frame-

work that allows building data applications using Python and fewer lines of code than traditional full-stack methods. With Streamlit there is no need to code the backend of the application, manage HTTP requests, or write HTML and CSS code for the frontend (Shukla et al., 2021).

Figure 25 illustrates the application´s interface, which can also be accessed via the link `https://medplant-ai.streamlit.app/`.

The first step in the application is to select the target species. On the 'backend' of the application, selecting a species will change the code that will be applied in the uploaded data, as both XCMS and machine learning model codes are different for both species.

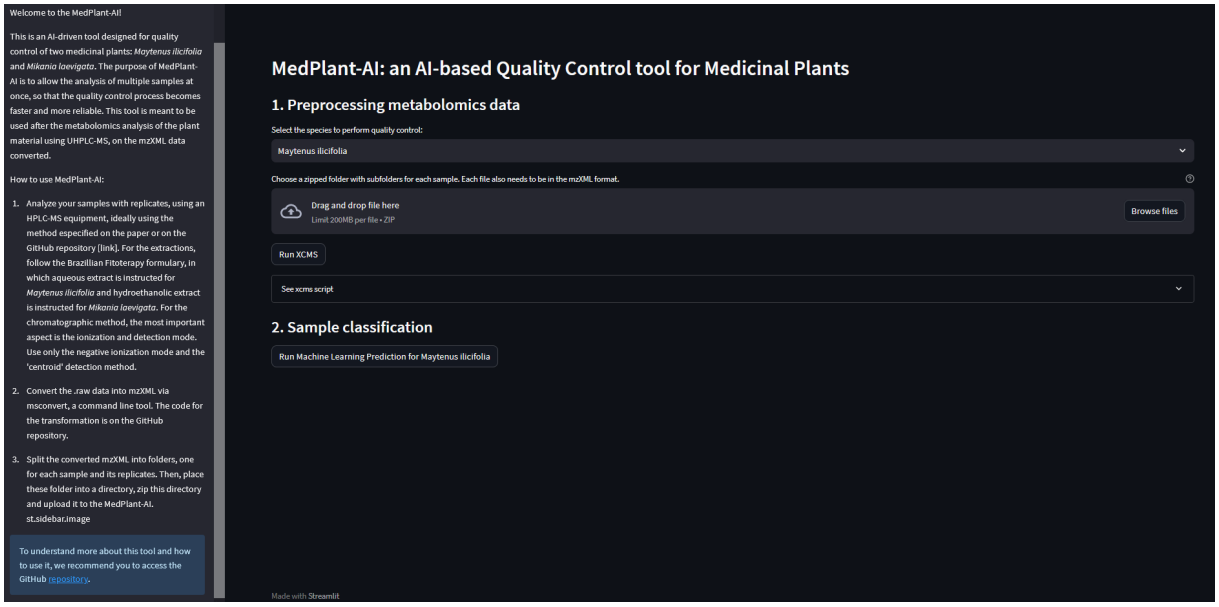

Figure 25: Machine Learning application's interface. On the left side, an instructions panel containing all the necessary steps the user needs to take to use the application properly. At the center, the application's interface is where the user will load the relevant data to be analyzed.

The next step is to upload the data into a zipped folder. Within the zipped folder, other sub-folders have to be present, for each sample that will be tested. These folders are necessary for XCMS preprocessing, which uses the folders to determine the groups.

As mentioned before, the groups within XCMS are important to determine which features are 'real' and which ones will be trimmed. The data within the sub-folders must be in mzXML format and the application and code needed to achieve this are provided in a left panel. This step was not added to the application due to technical limitations.

Within the 'back-end' of the application, the uploaded folder will be unzipped and the XCMS preprocessing will start, applying the relevant code depending on the selected species. Next, the data will pass to the machine learning model to be tested and after the prediction is obtained, it will be displayed on the screen, with the percentage of confidence for each sample to belong to the target species.

4 Conclusion

Currently, the Brazilian Pharmacopoeia's guidelines for quality control of regulated medicinal herbs rely on limited methods that focus on specific chemical markers, requiring expensive analytical standards. The alternative presented herein, which associates Untargeted Metabolomics with Machine Learning offers a more efficient, trustworthy, and cheaper way for the quality control of Brazilian medicinal species, specifically *Maytenus ilicifolia* and *Maytenus aquifolium*.

By combining these methodologies, we have developed two robust models that offer several advantages over traditional approaches, including efficiency, reliability, and cost-effectiveness. Our results showcase the power of machine learning algorithms in accurately classifying samples, achieving impressive MCC scores of 94% and 97% for *M. ilicifolia* and *M. aquifolium*, respectively.

These high scores highlight the efficacy of our approach in distinguishing between target samples, even among closely related species. Moreover, the utilization of machine learning enables rapid sample testing and data analysis, contributing to enhanced efficiency in the quality control process.

Furthermore, the synergy between machine learning and metabolomics enhances the consistency and reliability of results. By detecting abnormal patterns in sample composition, our approach ensures more trustworthy outcomes, particularly in scenarios with variations in growing conditions or the presence of adulterants. This aspect highlights the robustness of our method and its potential for application in diverse real-world settings.

An additional advantage of applying machine learning to metabolomics data is the elimination of the need for compound identification. Instead, the models can discern differences in data patterns without relying on specific analytical standards, significantly reducing costs compared to traditional methods. This cost-effectiveness makes our approach highly appealing for widespread adoption in the field of herbal medicine quality control. To the best of our knowledge, no similar strategy has been proposed in this field yet, especially for Brazilian medicinal species

Moreover, during the model training and validation, we employed both intra-dataset and inter-dataset validation techniques to ensure the robustness of our models. While intra-dataset validation is more commonly used, our adoption of inter-dataset validation allowed for a more realistic assessment of model performance across different datasets. This comparative analysis provided valuable insights into the generalization capabilities of our models and strengthened their reliability in real-world scenarios.

However, the present work presents limitations that can be addressed in the future. The next steps are to verify the model's performance on samples harvested and analyzed at different locations, equipment, and analytical methods. If the models continue to perform well, the next step is to improve/recreate the web application to allow the method to be used in actual

scenarios of quality control testing.

Finally, should these methods gain widespread adoption, further models can be developed for each Brazilian medicinal species described in the Brazilian Pharmacopoeia. This aims to improve the quality control of regulated medicinal herbs and, hopefully, establish the association between Untargeted Metabolomics and Machine Learning as the standard quality control method.

References

A. Alonso, S. Marsal e A. Julià. Analytical methods in untargeted metabolomics: state of the art in 2015. *Frontiers in bioengineering and biotechnology*, 3:23, 2015.

A. Altmann, L. Toloşi, O. Sander e T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 04 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq134. URL https://doi.org/10.1093/bioinformatics/btq134.

E. R. M. Antunes, R. S. Duarte, T. Moritz e A. C. H. F. Sawaya. Differentiation of two maytenus species and their hybrid via untargeted metabolomics. *INDUSTRIAL CROPS AND PRODUCTS*, 158, 2020.

E. R. M. Antunes et al. *Variabilidade sazonal da composição química e atividade antioxidante de extratos de folhas de Maytenus ilicifolia Mart. ex Reiss, Maytenus aquifolium Mart.(Celastraceae) e seus híbridos: Seasonal variation of the chemical composition and antioxidant activity of leaf extracts of Maytenus ilicifolia Mart. ex Reiss, Maytenus aquifolium Mart.(Celastraceae) and their hybrid*. PhD thesis, Universidade Estadual de Campinas (UNICAMP). Instituto de Biologia, 2019.

A. N. d. V. S. ANVISA. *Farmacopeia brasileira: volume 2: 6ª edição*, 2019.

A. N. d. V. S. ANVISA. *Formulário de fitoterápicos da farmacopeia brasileira 2ª Edição*, 2021.

C. Arnold, L. Biedebach, A. Küpfer e M. Neunhoeffer. The role of hyperparameters in machine learning models and how to tune them. *Political Science Research and Methods*, 2023.

A. M. Ascensión, O. Ibáñez-Solé, I. Inza, A. Izeta e M. J. Araúzo-Bravo. Triku: a feature selection method based on nearest neighbors for single-cell data. *GigaScience*, 11:giac017, 03 2022. ISSN 2047-217X. doi: 10.1093/gigascience/giac017. URL https://doi.org/10.1093/gigascience/giac017.

J. Bergstra e Y. Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

A. A. Borghi, C. d. L. Almeida e A. C. Sawaya. Damage and drying modify the composition of mikania glomerata and mikania laevigata leaves. *Revista Brasileira de Farmacognosia*, 29: 793–797, 2020.

A. A. Borghi, E. Minatel, D. S. Mizobuti, C. C. de Lourenço, F. F. de Araújo, G. M. Pastore, P. Hewitson, S. Ignatova e A. C. Sawaya. Antioxidant and anti-inflammatory activity of mikania glomerata and mikania laevigata extracts. *Pharmacognosy Research*, 15(1), 2023.

M. C. Chambers, B. Maclean, R. Burke, D. Amodei, D. L. Ruderman, S. Neumann, L. Gatto, B. Fischer, B. Pratt, J. Egertson et al. A cross-platform toolkit for mass spectrometry and proteomics. *Nature biotechnology*, 30(10):918–920, 2012.

M. Commisso, P. Strazzer, K. Toffali, M. Stocchero e F. Guzzo. Untargeted metabolomics: an emerging approach to determine the composition of herbal products. *Computational and structural biotechnology journal*, 4(5):e201301007, 2013.

V. Costa, A. Borghi, J. Mayer, J. Niehues, P. Bonetti, M. Souza, A. Maia, A. Piovezan, R. Peters, F. Oliveira et al. Comparison of the morphology, anatomy, and chemical profile of mikania glomerata and mikania laevigata. *Planta Medica*, 84(03):191–200, 2018.

A. Cutler, D. R. Cutler e J. R. Stevens. Random forests. In *Ensemble machine learning*, pages 157–175. Springer, 2012.

M. da Saúde. Portaria nº. 2.982 de 26 de novembro de 2009: Aprova as normas de execução e de financiamento da assistência farmacêutica na atenção básica, 2009.

C. de Lazzari Almeida, R. M. Xavier, A. A. Borghi, V. F. dos Santos e A. C. H. F. Sawaya. Effect of seasonality and growth conditions on the content of coumarin, chlorogenic acid and dicaffeoylquinic acids in mikania laevigata schultz and mikania glomerata sprengel (asteraceae) by uhplc–ms/ms. *International Journal of Mass Spectrometry*, 418:162–172, 2017a.

C. de Lazzari Almeida, R. M. Xavier, A. A. Borghi, V. F. dos Santos e A. C. H. F. Sawaya. Effect of seasonality and growth conditions on the content of coumarin, chlorogenic acid and dicaffeoylquinic acids in mikania laevigata schultz and mikania glomerata sprengel (asteraceae) by uhplc–ms/ms. *International Journal of Mass Spectrometry*, 418:162–172, 2017b.

C. de Lazzari Almeida, R. M. Xavier, A. A. Borghi, V. F. dos Santos e A. C. H. F. Sawaya. Effect of seasonality and growth conditions on the content of coumarin, chlorogenic acid and dicaffeoylquinic acids in mikania laevigata schultz and mikania glomerata sprengel (asteraceae) by uhplc–ms/ms. *International Journal of Mass Spectrometry*, 418:162–172, 2017c.

R. Di Guida, J. Engel, J. W. Allwood, R. J. M. Weber, M. R. Jones, U. Sommer, M. R. Viant e W. B. Dunn. Non-targeted UHPLC-MS metabolomic data processing methods: a comparative investigation of normalisation, missing value imputation, transformation and scaling. *Metabolomics*, 12(5):93, Apr. 2016.

R. S. Duarte, E. R. M. Antunes e A. C. H. F. Sawaya. Simultaneous uhplc-ms quantification of catechins and untargeted metabolomic profiling for proof-of-concept authenticity determination of maytenus ssp. samples. *Molecules*, 27(17), 2022. ISSN 1420-3049. doi: 10.3390/molecules27175520. URL https://www.mdpi.com/1420-3049/27/17/5520.

W. B. Dunn e D. I. Ellis. Metabolomics: current analytical platforms and methodologies. *TrAC Trends in Analytical Chemistry*, 24(4):285–294, 2005.

N. A. Eckardt, E. A. Ainsworth, R. N. Bahuguna, M. R. Broadley, W. Busch, N. C. Carpita, G. Castrillo, J. Chory, L. R. DeHaan, C. M. Duarte, A. Henry, S. V. K. Jagadish, J. A. Langdale, A. D. B. Leakey, J. C. Liao, K.-J. Lu, M. C. McCann, J. K. McKay, D. A. Odeny, E. Jorge de Oliveira, J. D. Platten, I. Rabbi, E. Y. Rim, P. C. Ronald, D. E. Salt, A. M. Shigenaga, E. Wang, M. Wolfe e X. Zhang. Climate change challenges, plant science solutions. *The Plant Cell*, 35(1):24–66,

10 2022. ISSN 1040-4651. doi: 10.1093/plcell/koac303. URL https://doi.org/10.1093/plcell/koac303.

H. S. Elshafie, I. Camele e A. A. Mohamed. A comprehensive review on the biological, agricultural and pharmaceutical properties of secondary metabolites based-plant origin. *International Journal of Molecular Sciences*, 24(4):3266, 2023.

P. M. Ferreira, C. N. de Oliveira, A. B. de Oliveira, M. J. Lopes, F. Alzamora e M. A. R. Vieira. A lyophilized aqueous extract of maytenus ilicifolia leaves inhibits histamine-mediated acid secretion in isolated frog gastric mucosa. *Planta*, 219:319–324, 2004.

H. Flück. The influence of climate on the active principles in medicinal plants. *Journal of Pharmacy and Pharmacology*, 7(1):361–383, 1955.

M. I. Galbiatti, G. P. Pinheiro, E. R. M. Antunes, V. V. Hernandes e A. C. H. F. Sawaya. Effect of environmental factors on plectranthus neochilus volatile composition: A gc-ms-based metabolomics approach. *Planta Medica International Open*, 8(03):e153–e160, 2021.

R. A. T. Games. *Contribuição ao controle de qualidade da espinheira santa (Maytenus ilicifolia mart. ex reiss. – celastraceae)*. PhD thesis, Universidade Bandeirante Brasil, Uniban, 2010.

A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2019.

V. Gilard, S. Balayssac, M. Malet-Martino e R. Martino. Quality control of herbal medicines assessed by nmr. *Current Pharmaceutical Analysis*, 6(4):234–245, 2010.

B. Gilbert e L. Alves. Synergy in plant medicines. *Current medicinal chemistry*, 10(1):13–20, 2003.

F. Gonzalez, T. Portela, E. Stipp e L. Di Stasi. Antiulcerogenic and analgesic effects of maytenus aquifolium, sorocea bomplandii and zolernia ilicifolia. *Journal of Ethnopharmacology*, 77(1): 41–47, 2001.

J. A. Harvey, K. Tougeron, R. Gols, R. Heinen, M. Abarca, P. K. Abram, Y. Basset, M. Berg, C. Boggs, J. Brodeur, P. Cardoso, J. G. de Boer, G. R. De Snoo, C. Deacon, J. E. Dell, N. Desneux, M. E. Dillon, G. A. Duffy, L. A. Dyer, J. Ellers, A. Espíndola, J. Fordyce, M. L. Forister, C. Fukushima, M. J. G. Gage, C. García-Robledo, C. Gely, M. Gobbi, C. Hallmann, T. Hance, J. Harte, A. Hochkirch, C. Hof, A. A. Hoffmann, J. G. Kingsolver, G. P. A. Lamarre, W. F. Laurance, B. Lavandero, S. R. Leather, P. Lehmann, C. Le Lann, M. M. López-Uribe, C.-S. Ma, G. Ma, J. Moiroux, L. Monticelli, C. Nice, P. J. Ode, S. Pincebourde, W. J. Ripple, M. Rowe, M. J. Samways, A. Sentis, A. A. Shah, N. Stork, J. S. Terblanche, M. P. Thakur, M. B. Thomas, J. M. Tylianakis, J. Van Baaren, M. Van de Pol, W. H. Van der Putten, H. Van Dyck, W. C. E. P. Verberk, D. L. Wagner, W. W. Weisser, W. C. Wetzel, H. A. Woods, K. A. G. Wyckhuys e S. L. Chown. Scientists' warning on climate change and insects. *Ecological Monographs*, 93(1):e1553, 2023. doi: https://doi.org/10.1002/ecm.1553. URL https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1002/ecm.1553.

K. E. Hoque e H. Aljamaan. Impact of hyperparameter tuning on machine learning models in stock price forecasting. *IEEE Access*, 9:163815–163830, 2021.

Y. Huang e P. Zhang. Evaluation of machine learning approaches for cell-type identification from single-cell transcriptomics data. *Briefings in bioinformatics*, 22(5):bbab035, 2021.

insightSLICE. Herbal medicine market global sales are expected to reach us$ 550 billion by 2030, as stated by insightslice. *GlobeNewswire*, 2021. URL https://www.globenewswire.com/en/news-release/2021/02/16/2176036/0/en/Herbal-Medicine-Market-Global-Sales-Are-Expected-To-Reach-US-550-Billion-by-2030.html.

G. James, D. Witten, T. Hastie, R. Tibshirani e J. Taylor. *An Introduction to Statistical Learning: with Applications in Python*. Springer Texts in Statistics. Springer New York, 2023. ISBN 9783031387470. URL https://link.springer.com/book/10.1007/978-3-031-38747-0.

F. Jiang, Q. Liu, Q. Li, S. Zhang, X. Qu, J. Zhu, G. Zhong e M. Huang. Signal drift in liquid chromatography tandem mass spectrometry and its internal standard calibration strategy for quantitative analysis. *Analytical Chemistry*, 92(11):7690–7698, 2020.

M. Katajamaa e M. Orešič. Data processing for mass spectrometry-based metabolomics. *Journal of chromatography A*, 1158(1-2):318–328, 2007.

L. C. Kenny, W. B. Dunn, D. I. Ellis, J. Myers, P. N. Baker e D. B. Kell. Novel biomarkers for pre-eclampsia detected using metabolomics and machine learning. *Metabolomics*, 1(3):227–234, 2005.

P. Kharyuk, D. Nazarenko, I. Oseledets, I. Rodin, O. Shpigun, A. Tsitsilin e M. Lavrentyev. Employing fingerprinting of medicinal plants by means of lc-ms and machine learning for species identification task. *Scientific reports*, 8(1):1–12, 2018.

C. Kuhl, R. Tautenhahn, C. Bottcher, T. R. Larson e S. Neumann. Camera: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets. *Analytical chemistry*, 84(1):283–289, 2012.

M. B. Kursa e W. R. Rudnicki. Feature selection with the boruta package. *Journal of Statistical Software*, 36(11):1–13, 2010. doi: 10.18637/jss.v036.i11. URL https://www.jstatsoft.org/index.php/jss/article/view/v036i11.

K.-M. Lee, J.-Y. Jeon, B.-J. Lee, H. Lee e H.-K. Choi. Application of metabolomics to quality control of natural product derived medicines. *Biomolecules & therapeutics*, 25(6):559, 2017.

Y. Li, X. Wang, C. Li, W. Huang, K. Gu, Y. Wang, B. Yang e Y. Li. Exploration of chemical markers using a metabolomics strategy and machine learning to study the different origins of ixeris denticulata (houtt.) stebb. *Food Chemistry*, 330:127232, 2020.

Y.-Z. Liang, P. Xie e K. Chan. Quality control of herbal medicines. *Journal of chromatography B*, 812(1-2):53–70, 2004.

P. Liashchynskyi e P. Liashchynskyi. Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*, 2019.

G. Libiseller, M. Dvorzak, U. Kleb, E. Gander, T. Eisenberg, F. Madeo, S. Neumann, G. Trausinger, F. Sinner, T. Pieber et al. Ipo: a tool for automated optimization of xcms parameters. *BMC bioinformatics*, 16(1):1–10, 2015.

U. W. Liebal, A. N. Phan, M. Sudhakar, K. Raman e L. M. Blank. Machine learning applications for mass spectrometry-based metabolomics. *Metabolites*, 10(6):243, 2020.

C. M. Loescher, D. W. Morton, S. Razic e S. Agatonovic-Kustrin. High performance thin layer chromatography (hptlc) and high performance liquid chromatography (hplc) for the qualitative and quantitative analysis of calendula officinalis—advantages and limitations. *Journal of Pharmaceutical and Biomedical Analysis*, 98:52–59, 2014.

L. McInnes, J. Healy e J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020.

L. V. d. Melo e A. C. Sawaya. Uhplc–ms quantification of coumarin and chlorogenic acid in extracts of the medicinal plants known as guaco (mikania glomerata and mikania laevigata). *Revista Brasileira de Farmacognosia*, 25:105–110, 2015.

A. Mohammadi, S. Bhattacharjee e S. Marcel. Improving cross-dataset performance of face presentation attack detection systems using face recognition datasets. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2947–2951, 2020. doi: 10.1109/ICASSP40776.2020.9053922.

J. B. Mokochinski, P. Mazzafera, A. C. H. F. Sawaya, R. Mumm, R. C. H. de Vos e R. D. Hall. Metabolic responses of eucalyptus species to different temperature regimes. *Journal of integrative plant biology*, 60(5):397–411, 2018.

M. Y. Morad, H. El-Sayed, M. F. El-Khadragy, A. Abdelsalam, E. Z. Ahmed e A. M. Ibrahim. Metabolomic profiling, antibacterial, and molluscicidal properties of the medicinal plants calotropis procera and atriplex halimus: In silico molecular docking study. *Plants*, 12(3), 2023. ISSN 2223-7747. doi: 10.3390/plants12030477. URL https://www.mdpi.com/2223-7747/12/3/477.

A. V. Nadimpalli e A. Rattani. On improving cross-dataset generalization of deepfake detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 91–99, June 2022.

D. Nazarenko, P. Kharyuk, I. Oseledets, I. Rodin e O. Shpigun. Machine learning for lc–ms medicinal plants identification. *Chemometrics and Intelligent Laboratory Systems*, 156:174–180, 2016.

D. Nazarenko, I. Rodin e O. Shpigun. The use of machine learning in the analytical control of the preparations of medicinal plants. *Inorganic Materials*, 55(14):1428–1438, 2019.

J. Y. Ng, S. Anant e N. D. Parakh. Characteristics of the research literature on herbal medicines corresponding with herbal supplements yielding the highest total sales: A bibliometric analysis. *Advances in Integrative Medicine*, 10(2):64–79, 2023. ISSN 2212-9588. doi: https://doi.org/10.1016/j.aimed.2023.05.004. URL https://www.sciencedirect.com/science/article/pii/S2212958823000514.

T. Okada, F. Mochamad Afendi, M. Altaf-Ul-Amin, H. Takahashi, K. Nakamura e S. Kanaya. Metabolomics of medicinal plants: the importance of multivariate analysis of analytical chemistry data. *Current Computer-Aided Drug Design*, 6(3):179–196, 2010.

G. J. Patti. Separation strategies for untargeted metabolomics. *J. Sep. Sci.*, 34(24):3460–3469, Dec. 2011.

G. J. Patti, O. Yanes e G. Siuzdak. Metabolomics: the apogee of the omics trilogy. *Nature reviews Molecular cell biology*, 13(4):263–269, 2012.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot e E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

P. G. Pedrioli, J. K. Eng, R. Hubley, M. Vogelzang, E. W. Deutsch, B. Raught, B. Pratt, E. Nilsson, R. H. Angeletti, R. Apweiler et al. A common open representation of mass spectrometry data and its application to proteomics research. *Nature biotechnology*, 22(11):1459–1466, 2004.

D. A. Pisner e D. M. Schnyer. Support vector machine. In *Machine learning*, pages 101–121. Elsevier, 2020.

M. M. Qaderi, A. B. Martel e C. A. Strugnell. Environmental factors regulate plant secondary metabolites. *Plants*, 12(3), 2023. ISSN 2223-7747. doi: 10.3390/plants12030447. URL https://www.mdpi.com/2223-7747/12/3/447.

M. Ramírez-Meraz, R. Méndez-Aguilar, D. Hidalgo-Martínez, N. Villa-Ruano, L. G. Zepeda-Vallejo, F. Vallejo-Contreras, C. J. Hernández-Guerrero e E. Becerra-Martínez. Experimental races of capsicum annuum cv. jalapeño: Chemical characterization and classification by 1h nmr/machine learning. *Food Research International*, 138:109763, 2020.

P. Rasoanaivo, C. W. Wright, M. L. Willcox e B. Gilbert. Whole plant extracts versus single compounds for the treatment of malaria: synergy and positive interactions. *Malaria journal*, 10(1):1–12, 2011.

M. S. d. Reis. *Conservação e uso sustentável de plantas medicinais e aromáticas - Maytenus spp., espinheira-santa*. Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais Renováveis, 1 edition, 2004. ISBN 9788573001.

J. Rodríguez-Coira, M. I. Delgado-Dolset, D. Obeso, M. Dolores-Hernández, G. Quintás, S. Angulo, D. Barber, T. Carrillo, M. M. Escribese e A. Villaseñor. Troubleshooting in large-scale lc-tof-ms metabolomics analysis: Solving complex issues in big cohorts. *Metabolites*, 9(11), 2019.

ISSN 2218-1989. doi: 10.3390/metabo9110247. URL https://www.mdpi.com/2218-1989/9/11/247.

U. Salam, S. Ullah, Z.-H. Tang, A. A. Elateeq, Y. Khan, J. Khan, A. Khan e S. Ali. Plant metabolomics: An overview of the role of primary and secondary metabolites against different environmental stress factors. *Life*, 13(3):706, 2023.

J. C. Sánchez-Rangel, J. Benavides, J. B. Heredia, L. Cisneros-Zevallos e D. A. Jacobo-Velázquez. The folin–ciocalteu assay revisited: improvement of its specificity for total phenolic content determination. *Analytical Methods*, 5(21):5990–5999, 2013.

R. Santos-Oliveira, S. Coulaud-Cunha e W. Colaço. Revisão da maytenus ilicifolia mart. ex reissek, celastraceae. contribuição ao estudo das propriedades farmacológicas. *Revista Brasileira de Farmacognosia*, 19:650–659, 2009.

S. Shukla, A. Maheshwari e P. Johri. Comparative analysis of ml algorithms & stream lit web application. In *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, pages 175–180, 2021. doi: 10.1109/ICAC3N53548.2021.9725496.

E. L. Singsaas. Terpenes and the thermotolerance of photosynthesis. *The New Phytologist*, 146 (1):1–4, 2000.

C. A. Smith, E. J. Want, G. O'Maille, R. Abagyan e G. Siuzdak. Xcms: processing mass spectrometry data for metabolite profiling using nonlinear peak alignment, matching, and identification. *Analytical chemistry*, 78(3):779–787, 2006.

M. L. O. Souza-Formigoni, M. G. M. Oliveira, M. G. Monteiro, N. G. da Silveira-Filho, S. Braz e E. Carlini. Antiulcerogenic effects of two maytenus species in laboratory animals. *Journal of Ethnopharmacology*, 34(1):21–27, 1991.

R. Srirama, J. Santhosh Kumar, G. Seethapathy, S. G. Newmaster, S. Ragupathy, K. Ganeshaiah, R. Uma Shaanker e G. Ravikanth. Species adulteration in the herbal trade: causes, consequences and mitigation. *Drug safety*, 40:651–661, 2017.

M. Steinbach e P.-N. Tan. knn: k-nearest neighbors. In *The top ten algorithms in data mining*, pages 165–176. Chapman and Hall/CRC, 2009.

M. Strangfeld. *Reproducibility of Computational Environments for Software Development*. PhD thesis, Rheinisch-Westfälische Technische Hochschule Aachen, 2022.

C. Strobl, A.-L. Boulesteix, A. Zeileis e T. Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8(1):1–21, 2007.

H. Suzuki, T. Nishizawa, H. Tsugawa, S. Mogami e T. Hibi. Roles of oxidative stress in stomach disorders. *Journal of clinical biochemistry and nutrition*, 50(1):35–39, 2011.

X. Tian, G. Zhang, Y. Shao e Z. Yang. Towards enhanced metabolomic data analysis of mass spectrometry image: Multivariate curve resolution and machine learning. *Analytica chimica acta*, 1037:211–219, 2018.

V. A. Ueno e A. C. H. F. Sawaya. Influence of environmental factors on the volatile composition of two brazilian medicinal plants: Mikania laevigata and mikania glomerata. *Metabolomics*, 15:1–11, 2019.

N. M. van Dam e E. van der Meijden. A role for metabolomics in plant ecology. *Annual Plant Reviews Volume 43: Biology of Plant Metabolomics*, 43:87–107, 2011.

A. Wahid. Physiological implications of metabolite biosynthesis for net assimilation and heat-stress tolerance of sugarcane (saccharum officinarum) sprouts. *Journal of plant Research*, 120: 219–228, 2007.

M. Walker. *Data Cleaning and Exploration with Machine Learning*. Packt Publishing Ltd., 2022. ISBN 9781803241678. URL https://www.oreilly.com/library/view/data-cleaning-and/9781803241678/.

W. World Health Organization. *WHO guidelines on safety monitoring of herbal medicines in pharmacovigilance systems*, 2004.

J. H. Yariwake, F. M. Lanças, E. A. Cappelaro, E. C. d. Vasconcelos, L. A. Tiberti, A. Pereira e S. d. C. Franca. Variabilidade sazonal de constituintes químicos (triterpenos, flavonóides e polifenóis) das folhas de maytenus aquifolium mart.(celastraceae). *Revista Brasileira de Farmacognosia*, 15:162–168, 2005.

A. C. Zanatta, N. C. Vieira, R. Dantas-Medeiros, W. Vilegas e R. Edrada-Ebel. Understanding the seasonal effect of metabolite production in terminalia catappa l. leaves through a concatenated ms- and nmr-based metabolomics approach. *Metabolites*, 13(3), 2023. ISSN 2218-1989. doi: 10.3390/metabo13030349. URL https://www.mdpi.com/2218-1989/13/3/349.

X.-D. Zhang. *A Matrix Algebra Approach to Artificial Intelligence*. Springer Nature, 2020.

A Supplementary Material

```
msconvert *RAW --mzXML --zlib -o mzXML --filter peakPicking cwt snr=0.1 peakSpace
    =0.1 msLevel=1-
```

Code Listing 1: MSConvert

```
library(xcms)
library(CAMERA)
library(IPO)

datafiles <- list.files("./[...]/mzXML",
                        recursive = TRUE, full.names = TRUE)

files_list <- list(datafiles)

# Peak picking parameters optimization
peakpickingParameters <- getDefaultXcmsSetStartingParams('matchedFilter')

time.xcmsSet <- system.time({
                resultPeakpicking <-
                            optimizeXcmsSet(files = datafiles[1205:1280],
                            params = peakpickingParameters,
                            nSlaves = 1,
                            subdir = "./IPO",
                            plot = TRUE)
                            })

resultPeakpicking$best_settings$result

optimizedXcmsSetObject <- resultPeakpicking$best_settings$xset


# Retention time correction optimization
retcorGroupParameters <- getDefaultRetGroupStartingParams()
time.RetGroup <- system.time({
            resultRetcorGroup <-
                            optimizeRetGroup(xset = optimizedXcmsSetObject,
                            params = retcorGroupParameters,
                            nSlaves = 1,
                            subdir =  "./IPO",
                            plot = TRUE)
                            })

writeRScript(resultPeakpicking$best_settings$parameters,
             resultRetcorGroup$best_settings)

time.xcmsSet
```

```
time.RetGroup
```

Code Listing 2: IPO

```
splitfolders . --ratio 0.8 0.2 --group_prefix 3 --seed 2187
```

Code Listing 3: train test split

```
xset <- xcmsSet(
        method   = "matchedFilter",
        fwhm     = 18,
        snthresh = 3,
        step     = 1,
        steps    = 5,
        sigma    = 12.9522677085103,
        max      = 5,
        mzdiff   = 1,
        index    = FALSE)

xset2 <- retcor(
        xset,
        method         = "obiwarp",
        plottype       = "none",
        distFunc       = "cor_opt",
        profStep       = 1,
        center         = 13,
        response       = 1,
        gapInit        = 0.4,
        gapExtend      = 2.4,
        factorDiag     = 2,
        factorGap      = 1,
        localAlignment = 1)

xset3 <- group(
        xset2,
        method  = "density",
        bw      = 25,
        mzwid   = 0.1,
        minfrac = 0.1,
        minsamp = 10,
        max     = 100)

xset4 <- fillPeaks(xset3)

an <- xsAnnotate(xset4)
anF <- groupFWHM(an, perfwhm = 0.6)
anI <- findIsotopes(anF, mzabs=0.01)
anIC <- groupCorr(anI, cor_eic_th=0.75)
anFA <- findAdducts(anIC, polarity="negative")

write.csv(getPeaklist(anIC), file="maytenus.csv")
```

Code Listing 4: xcms experiment for Maytenus

```
xset <- xcmsSet(
        method   = "matchedFilter",
        fwhm     = 28,
        snthresh = 3,
        step     = 0.15,
        steps    = 3,
        sigma    = 11.8906064209275,
        max      = 5,
        mzdiff   = 0.35,
        index    = FALSE)

xset2 <- retcor(
        xset,
        method         = "obiwarp",
        plottype       = "none",
        distFunc       = "cor_opt",
        profStep       = 1,
        center         = 68,
        response       = 1,
        gapInit        = 0.2,
        gapExtend      = 2.4,
        factorDiag     = 2,
        factorGap      = 1,
        localAlignment = 0)

xset3 <- group(
        xset2,
        method  = "density",
        bw      = 29.2,
        mzwid   = 0.015,
        minfrac = 0.2,
        minsamp = 1,
        max     = 50)

xset4 <- fillPeaks(xset3)

an <- xsAnnotate(xset4)
anF <- groupFWHM(an, perfwhm = 0.6)
anI <- findIsotopes(anF, mzabs=0.01)
anIC <- groupCorr(anI, cor_eic_th=0.75)
anFA <- findAdducts(anIC, polarity="negative")

write.csv(getPeaklist(anIC), file='mikania.csv')
```

Code Listing 5: xcms experiment for Mikania

```

library(ggfortify)
library(cluster)
library(patchwork)

# Maytenus plots

```

```
data_posneg <- read.csv('./maytenus_qc_posneglabel.csv', sep = ",")
data_species <- read.csv('./maytenus_qc_specieslabel.csv', sep = ",")

pca_data_posneg = data_posneg[3:268]
pca_data_species = data_species[3:268]

pca_res_posneg <- prcomp(pca_data_posneg, scale. = TRUE, center = TRUE)
pca_res_species <- prcomp(pca_data_species, scale. = TRUE, center = TRUE)

png("PCA_maytenus.png", height = 480, width = 1080 )
posneg <- autoplot(pca_res_posneg, data = data_posneg,
                   colour = 'Label', size=2)
species <- autoplot(pca_res_species, data = data_species,
                    colour = 'Label', size=2)
posneg + species
dev.off()


# Mikania plots

data_posneg <- read.csv('./mikania_qc_posneg.csv', sep = ",")
data_species <- read.csv('./mikania_ipoqc_species.csv', sep = ",")

pca_data_posneg = data_posneg[3:166]
pca_data_species = data_species[3:166]

pca_res_posneg <- prcomp(pca_data_posneg, scale. = TRUE, center = TRUE)
pca_res_species <- prcomp(pca_data_species, scale. = TRUE, center = TRUE)

png("PCA_mikania.png", height = 480, width = 1080 )
posneg <- autoplot(pca_res_posneg, data = data_posneg,
                   colour = 'Label', size=2)
species <- autoplot(pca_res_species, data = data_species,
                    colour = 'Label', size=2)
posneg + species
dev.off()
```

Code Listing 6: PCA plot functions applied to create PCAs for both species

```

def Tus_normalized(injection_order, xcms_table, class_column, color_class = True,
    color = 'dark_red', plot_title = 'Total Useful Signal', save_plot = True,
    figure_name = 'TUS.tif',figure_format = 'tif', figure_resolution = 300,
    directory = './'):

        """
    injection_order: Path to the injection order table - Must contain   ONE column
    named 'samples' with the samples ordered according to the injection order
    xcms_table: Path to the xcms table. The first column should be named 'samples'
    and contain the sample names in the same format as the injection order table.
    The following columns should each contain one feature (m/z and retention time).
    plot_title: Title of the plot. Must be written between quotes (" ").
    color: Color of the plot. Supported values are: aquamarine, green, light_blue,
```

```
    dark_blue, orange, red, dark_red, magenta, pink, purple, brown and grey
    save_plot: If True, save the plot as a file
    figure_name: File name. The name MUST contain the file extension
    figure_format: File format such as 'png', 'tif', 'jpeg' etc.
    figure_resolution: File resolution in dpi. If 'figure', the saved figure will
    have the same resolution as the one presented
    directory: Folder in which the plot will be saved

    """

    colors = {'aquamarine': 'Dark2',
              'green': 'YlGn_r',
              'light_blue': 'Spectral_r',
              'dark_blue':'Blues_r' ,
              'orange': 'Wistia_r',
              'red': 'Set1',
              'dark_red': 'Reds_r',
              'magenta': 'PuRd_r',
              'pink': 'PiYG' ,
              'purple': 'Purples_r' ,
              'brown': 'Oranges_r',
              'grey': 'Greys_r',
              'inferno': 'inferno',
              'red_blue': 'RdBu'}

    if color in colors.keys():
        # loading the datasets
        order = pd.read_csv(injection_order)
        intensity_table = pd.read_csv(xcms_table)

        # reordering the intensity table
        ordered_table = pd.merge(order, intensity_table, how='outer')
        ordered_table['sum'] = ordered_table.sum(numeric_only=True, axis=1)

        # indexes to list
        index = ordered_table.index.tolist()

        # PLOT

        # figsize
        plt.figure(figsize=(15,8))

        plt.ylim(min(ordered_table['sum']), max(ordered_table['sum']))

        # theme
        custom_params = {"axes.spines.right": False, "axes.spines.top": False}
        sns.set_theme(style = 'ticks',rc=custom_params, palette=colors[color])

        if color_class:

            ax = sns.scatterplot(x=index , y = ordered_table['sum'], hue =
    ordered_table[class_column], legend = 'full', alpha=0.7, s=80)
```

```
58
59      else:
60          ax = sns.scatterplot(x=index , y = ordered_table['sum'], alpha=0.7, s
    =80)
61
62      # titles and axis labels
63      ax.set_xlabel("Injection Order", fontsize = 13)
64      ax.set_ylabel("Intensity", fontsize = 13)
65      ax.set_title(plot_title, fontsize = 15);
66
67      # mean and std lines
68      mean = [np.mean(ordered_table['sum'])]*len(ordered_table['sum'])
69      std = [np.std(ordered_table['sum'])][0]
70
71      mean_line = ax.plot(index,mean, label='Mean', linestyle='--', color = 'k')
72      mean_line_1 = ax.plot(index,([x+std for x in mean]), label='Mean + 1 Std',
    linestyle='--', color = 'darkgrey')
73      mean_line_2 = ax.plot(index,([x+2*std for x in mean]), label='Mean + 2 Std',
     linestyle='--', color = 'darkgrey')
74      mean_line_3 = ax.plot(index,([x+3*std for x in mean]), label='Mean + 3 Std',
     linestyle='--', color = 'darkgrey')
75      mean_line_m1 = ax.plot(index,([x-std for x in mean]), label='Mean - 1 Std',
    linestyle='--', color = 'darkgrey')
76      mean_line_m2 = ax.plot(index,([x-2*std for x in mean]), label='Mean - 2 Std'
    , linestyle='--', color = 'darkgrey')
77      mean_line_m3 = ax.plot(index,([x-3*std for x in mean]), label='Mean - 3 Std'
    , linestyle='--', color = 'darkgrey')
78
79      #     save plot
80      if save_plot:
81          plt.savefig(directory + figure_name, format = figure_format, dpi =
    figure_resolution)
82
83      return plt.show()
84
85  else:
86      print(f"Please enter a valid color. \nSupported values are: \n  {', '.join
    ([*colors])}")
```

Code Listing 7: MSTUS plot function

Table 14: Features selected by the three methods for the Maytenus experiment

Mutual Information	**RFE**	**Boruta**
-	114_53.0	114_53.0
-	-	116_55.7
118_109.9	118_109.9	118_109.9
-	132_560.4	-
133_55.4	133_55.4	133_55.4
-	143_560.1	143_560.1

-	144_560.2	144_560.2
164_211.0	164_211.0	164_211.0
178_42.8	178_42.8	178_42.8
181_42.1	181_42.1	181_42.1
-	185_331.6	185_331.6
191_222.3	-	191_222.3
-	191_90.0	191_90.0
192_47.4	192_47.4	192_47.4
193_46.8	193_46.8	193_46.8
194_45.3	194_45.3	194_45.3
195_44.1	195_44.1	195_44.1
203_91.5	203_91.5	203_91.5
-	206_333.5	-
207_328.6	207_328.6	-
209_46.9	209_46.9	209_46.9
210_47.9	210_47.9	210_47.9
217_42.8	217_42.8	217_42.8
219_44.1	219_44.1	219_44.1
220_44.4	220_44.4	220_44.4
221_44.6	221_44.6	221_44.6
222_45.2	222_45.2	222_45.2
233_49.0	233_49.0	233_49.0
234_48.4	234_48.4	234_48.4
245_247.7	245_247.7	-
257_328.2	-	257_328.2
264_47.6	264_47.6	264_47.6
-	264_569.4	264_569.4
-	265_569.9	265_569.9
-	266_568.7	266_568.7
271_41.4	271_41.4	271_41.4
272_256.7	272_256.7	272_256.7
-	272_38.4	272_38.4
273_260.2	273_260.2	273_260.2
274_261.2	-	274_261.2
276_41.6	276_41.6	276_41.6
277_43.1	277_43.1	277_43.1
278_43.9	278_43.9	278_43.9
-	279_296.8	279_296.8
-	281_44.8	-
-	-	283_108.4

287_72.7	287_72.7	287_72.7
289_226.5	289_226.5	289_226.5
290_195.7	290_195.7	290_195.7
293_312.1	293_312.1	293_312.1
293_443.9	293_443.9	293_443.9
294_443.5	294_443.5	294_443.5
295_441.3	295_441.3	295_441.3
303_47.8	303_47.8	303_47.8
305_248.6	-	-
309_127.7	-	-
310_426.4	310_426.4	310_426.4
311_151.2	311_151.2	-
311_431.5	-	311_431.5
312_164.2	312_164.2	312_164.2
-	318_46.5	-
-	319_46.0	319_46.0
-	320_46.4	-
-	321_46.6	-
326_238.5	326_238.5	-
327_193.6	327_193.6	327_193.6
328_382.3	328_382.3	328_382.3
329_192.2	329_192.2	329_192.2
329_390.6	329_390.6	329_390.6
-	-	330_239.4
-	332_228.4	-
335_229.4	-	-
336_227.5	-	336_227.5
-	339_46.3	339_46.3
344_329.6	344_329.6	344_329.6
348_203.0	348_203.0	348_203.0
353_197.1	353_197.1	353_197.1
353_222.5	353_222.5	353_222.5
355_198.2	355_198.2	355_198.2
368_264.2	368_264.2	368_264.2
369_228.8	369_228.8	369_228.8
370_212.5	370_212.5	370_212.5
371_210.2	371_210.2	371_210.2
378_43.7	-	-
379_43.1	-	-
380_43.3	-	-

381_43.3	-	-
382_43.7	-	-
-	388_225.9	-
393_165.1	393_165.1	393_165.1
393_174.8	-	393_174.8
396_178.4	-	-
410_366.5	410_366.5	410_366.5
-	-	414_42.4
423_409.8	-	423_409.8
428_314.9	-	428_314.9
429_44.1	429_44.1	429_44.1
432_262.4	432_262.4	432_262.4
433_265.2	-	-
439_374.3	439_374.3	439_374.3
-	440_48.3	-
-	451_169.5	-
452_222.3	452_222.3	452_222.3
453_221.5	453_221.5	453_221.5
456_168.3	456_168.3	456_168.3
-	457_167.9	457_167.9
463_281.8	463_281.8	463_281.8
464_252.1	464_252.1	464_252.1
-	469_132.8	469_132.8
474_46.0	-	-
475_46.4	-	-
476_47.3	-	-
479_229.8	-	479_229.8
483_366.5	483_366.5	483_366.5
487_396.1	-	-
515_327.3	515_327.3	515_327.3
-	516_44.6	-
517_326.3	517_326.3	517_326.3
-	517_43.9	-
525_290.2	525_290.2	525_290.2
526_360.5	526_360.5	526_360.5
533_250.6	533_250.6	533_250.6
535_44.2	-	-
536_43.8	-	-
538_43.7	-	-
545_257.3	545_257.3	545_257.3

547_256.5	547_256.5	547_256.5
548_256.2	548_256.2	548_256.2
560_299.9	560_299.9	560_299.9
561_208.6	561_208.6	561_208.6
561_236.1	561_236.1	561_236.1
562_259.5	562_259.5	562_259.5
564_239.2	564_239.2	564_239.2
-	566_134.7	566_134.7
-	569_280.4	569_280.4
574_146.2	574_146.2	574_146.2
577_177.9	577_177.9	577_177.9
578_221.3	578_221.3	578_221.3
579_224.5	579_224.5	579_224.5
579_225.8	-	579_225.8
593_276.4	593_276.4	593_276.4
594_271.9	594_271.9	594_271.9
597_205.6	597_205.6	597_205.6
-	598_207.0	598_207.0
607_43.4	-	-
610_259.2	-	610_259.2
610_261.6	610_261.6	610_261.6
612_308.9	612_308.9	612_308.9
615_44.8	-	615_44.8
-	624_82.2	624_82.2
626_130.0	626_130.0	626_130.0
-	627_148.0	627_148.0
652_283.6	-	-
669_437.4	669_437.4	669_437.4
688_300.8	688_300.8	688_300.8
690_299.8	690_299.8	690_299.8
690_432.3	690_432.3	690_432.3
691_299.4	691_299.4	691_299.4
691_300.9	691_300.9	691_300.9
691_433.7	691_433.7	691_433.7
692_432.8	692_432.8	692_432.8
704_436.3	704_436.3	704_436.3
705_436.4	705_436.4	705_436.4
707_222.0	707_222.0	707_222.0
723_301.1	723_301.1	723_301.1
725_314.3	725_314.3	725_314.3

727_361.7	727_361.7	727_361.7
739_256.6	739_256.6	739_256.6
740_259.2	740_259.2	740_259.2
-	746_437.1	746_437.1
-	747_433.5	747_433.5
-	747_436.3	747_436.3
756_241.1	756_241.1	756_241.1
777_258.4	-	-
778_255.7	-	-
-	-	818_320.2
-	827_42.9	-
830_291.0	830_291.0	830_291.0
831_278.4	831_278.4	831_278.4
832_275.7	832_275.7	832_275.7
833_166.8	833_166.8	833_166.8
833_291.0	833_291.0	833_291.0
834_230.7	834_230.7	834_230.7
835_202.6	835_202.6	835_202.6
836_273.8	836_273.8	836_273.8
848_182.7	848_182.7	848_182.7
849_190.0	849_190.0	849_190.0
850_216.6	-	850_216.6
851_228.7	851_228.7	851_228.7
866_204.8	-	-
868_226.2	-	-
-	893_42.7	893_42.7
902_221.9	902_221.9	902_221.9
918_205.7	918_205.7	918_205.7

Table 15: Features selected by the three methods for the Mikania experiment

Mutual Information	**RFE**	**Boruta**
-	1000_338.1	1000_338.1
-	101_572.8	-
111_45.2	111_45.2	111_45.2
115_46.5	115_46.5	115_46.5
117_572.4	117_572.4	117_572.4
119_337.1	119_337.1	119_337.1
-	119_41.3	-
121_320.5	121_320.5	121_320.5

133_46.5	133_46.5	133_46.5
-	136_572.4	-
-	159_37.5	159_37.5
-	161_323.4	161_323.4
-	161_572.4	-
163_247.9	163_247.9	163_247.9
163_337.4	163_337.4	163_337.4
165_230.9	165_230.9	165_230.9
165_321.0	165_321.0	165_321.0
165_43.4	-	165_43.4
-	179_41.6	179_41.6
181_42.3	181_42.3	181_42.3
-	187_334.8	187_334.8
191_176.4	191_176.4	191_176.4
-	191_597.4	-
204_46.9	204_46.9	204_46.9
-	205_572.8	-
210_44.6	210_44.6	210_44.6
216_43.0	-	216_43.0
217_43.2	217_43.2	217_43.2
-	217_572.8	217_572.8
-	241_572.8	-
-	274_37.0	274_37.0
278_530.2	278_530.2	278_530.2
-	-	290_229.7
302_547.3	302_547.3	-
306_209.5	306_209.5	-
318_433.7	318_433.7	318_433.7
326_181.2	326_181.2	326_181.2
326_247.9	326_247.9	326_247.9
328_231.5	328_231.5	328_231.5
-	338_337.0	338_337.0
-	342_40.8	-
-	345_578.0	345_578.0
-	350_172.5	350_172.5
354_176.4	354_176.4	354_176.4
354_323.2	354_323.2	354_323.2
362_247.4	362_247.4	362_247.4
372_250.1	372_250.1	372_250.1
-	-	378_43.5

-	380_43.1	-
-	388_39.7	-
388_515.4	388_515.4	388_515.4
390_515.1	390_515.1	390_515.1
-	400_522.1	400_522.1
406_328.8	406_328.8	406_328.8
-	410_348.7	410_348.7
424_380.9	424_380.9	424_380.9
434_279.6	434_279.6	434_279.6
440_356.3	440_356.3	440_356.3
442_356.4	-	442_356.4
-	448_317.3	-
448_530.3	448_530.3	448_530.3
-	450_298.3	-
-	472_40.0	-
478_319.0	478_319.0	478_319.0
482_269.5	482_269.5	482_269.5
-	484_349.2	484_349.2
488_348.4	488_348.4	488_348.4
490_371.4	490_371.4	490_371.4
492_344.6	492_344.6	492_344.6
494_299.4	-	494_299.4
-	500_338.9	500_338.9
-	508_338.3	-
-	514_350.0	514_350.0
516_323.2	516_323.2	516_323.2
518_327.1	518_327.1	518_327.1
526_269.6	526_269.6	526_269.6
-	530_342.2	530_342.2
534_211.1	534_211.1	534_211.1
-	534_43.5	534_43.5
548_530.5	548_530.5	548_530.5
-	550_343.8	550_343.8
560_332.3	560_332.3	560_332.3
562_243.2	562_243.2	562_243.2
562_321.3	562_321.3	562_321.3
572_353.1	572_353.1	572_353.1
574_377.6	574_377.6	574_377.6
-	578_210.6	578_210.6
580_230.9	580_230.9	580_230.9

594_310.9	594_310.9	594_310.9
-	-	610_280.1
626_333.6	626_333.6	626_333.6
640_284.1	640_284.1	640_284.1
652_181.2	652_181.2	652_181.2
652_247.9	652_247.9	652_247.9
652_319.3	652_319.3	652_319.3
654_247.9	654_247.9	654_247.9
655_331.2	655_331.2	655_331.2
656_231.0	-	656_231.0
674_247.9	674_247.9	674_247.9
680_179.4	680_179.4	680_179.4
708_176.2	708_176.2	708_176.2
-	726_341.4	726_341.4
740_274.3	-	740_274.3
756_258.9	756_258.9	756_258.9
834_287.1	834_287.1	834_287.1
876_360.2	876_360.2	876_360.2
904_381.6	904_381.6	904_381.6
918_353.8	918_353.8	918_353.8
920_360.1	920_360.1	920_360.1
-	923_330.0	923_330.0
-	940_376.4	940_376.4
946_391.7	-	946_391.7
962_367.5	-	962_367.5
-	988_298.7	988_298.7

Table 16: Permutation Importance for Maytenus model on train and test set

Feature	Permutation Importance (train)	Permutation Importance (test)
739_256.6	0.023640016	0.412407509
133_55.4	0.009833473	-0.001245575
289_226.5	0.005931235	0.044483909
515_327.3	0.004041797	0
561_208.6	0.002915579	0
353_222.5	0.002907578	0.004018015
487_396.1	0.002276546	0
579_224.5	0.001240925	0.005918971
370_212.5	0.001065229	0
578_221.3	0.000683383	0

432_262.4	0.00063564	0
326_238.5	0.000569425	0
834_230.7	0.000426886	-0.001798233
329_390.6	0.000355981	0
310_426.4	0.000284785	0
192_47.4	0.000276801	0
368_264.2	0.000213589	0
433_265.2	0.000212281	0
193_46.8	0.000211026	0
194_45.3	0.000211026	0
423_409.8	0.000142392	0
439_374.3	0.000142392	0
652_283.6	0.000142392	0
203_91.5	7.12E-05	-0.000224779
328_382.3	7.12E-05	0
525_290.2	7.12E-05	0
533_250.6	7.12E-05	0
118_109.9	0	0
164_211.0	0	0
191_222.3	0	0
207_328.6	0	0
209_46.9	0	0
210_47.9	0	0
221_44.6	0	0
222_45.2	0	0
233_49.0	0	0
234_48.4	0	0
245_247.7	0	0
257_328.2	0	0
264_47.6	0	0
271_41.4	0	0
272_256.7	0	0
273_260.2	0	0.000238363
274_261.2	0	0
276_41.6	0	0
277_43.1	0	0
287_72.7	0	0
293_312.1	0	0
293_443.9	0	0
294_443.5	0	0

295_441.3	0	0
303_47.8	0	0
305_248.6	0	0
309_127.7	0	0
311_431.5	0	0
329_192.2	0	0
336_227.5	0	0
344_329.6	0	0
348_203.0	0	0
371_210.2	0	0
378_43.7	0	0
379_43.1	0	0
380_43.3	0	0
381_43.3	0	0
382_43.7	0	0
393_165.1	0	0
393_174.8	0	0
396_178.4	0	0
410_366.5	0	0
428_314.9	0	0
429_44.1	0	0
456_168.3	0	0
463_281.8	0	0.000715088
464_252.1	0	0
474_46.0	0	0
475_46.4	0	0
476_47.3	0	0
479_229.8	0	0
483_366.5	0	0
517_326.3	0	0
526_360.5	0	0
535_44.2	0	0
536_43.8	0	0
538_43.7	0	0
545_257.3	0	0
547_256.5	0	0
548_256.2	0	0
560_299.9	0	0
564_239.2	0	0
574_146.2	0	0

577_177.9	0	0
593_276.4	0	-0.002922128
594_271.9	0	0
597_205.6	0	0
607_43.4	0	0
610_259.2	0	0
610_261.6	0	0.000251946
612_308.9	0	0
615_44.8	0	0
626_130.0	0	0
669_437.4	0	0
690_432.3	0	0
691_433.7	0	0
692_432.8	0	0
704_436.3	0	0
705_436.4	0	0
723_301.1	0	0
725_314.3	0	0
727_361.7	0	0
777_258.4	0	0
778_255.7	0	0
833_166.8	0	0
835_202.6	0	0
848_182.7	0	0
849_190.0	0	0
850_216.6	0	0
866_204.8	0	0
868_226.2	0	0
195_44.1	-7.84E-06	0
327_193.6	-7.13E-05	0
335_229.4	-7.13E-05	0
278_43.9	-7.38E-05	0
311_151.2	-7.38E-05	0
355_198.2	-7.38E-05	0
707_222.0	-7.90E-05	0
353_197.1	-0.000147518	0
452_222.3	-0.000155207	0
851_228.7	-0.000214025	-0.003596465
691_300.9	-0.000285512	0
836_273.8	-0.000287784	0

690_299.8	-0.000356708	0
831_278.4	-0.000374787	0
217_42.8	-0.000499391	0
220_44.4	-0.000499391	0
688_300.8	-0.000499391	0
830_291.0	-0.000499537	0
691_299.4	-0.000570733	0
312_164.2	-0.000590073	0
832_275.7	-0.000665715	0
178_42.8	-0.000934693	0
181_42.1	-0.000934693	-0.00537478
833_291.0	-0.001015251	0
756_241.1	-0.001455388	-0.000324231
290_195.7	-0.001506942	0
369_228.8	-0.001514776	-0.002249194
579_225.8	-0.001653519	0
562_259.5	-0.001713057	0
453_221.5	-0.001854882	0
561_236.1	-0.002102275	0.043213462
219_44.1	-0.0034285	-0.001348675
918_205.7	-0.003570154	0
902_221.9	-0.003746026	0
740_259.2	-0.006506251	0

Table 17: Feature Importance and Permutation importance of all Mikania's features

Feature	Feature Importance	Permutation Importance (train set)	Permutation Importance (test set)
111_45.2	0.064567327	0.002475816	0.00027443
115_46.5	0.081479406	0.003412564	0.004665311
117_572.4	0.099477111	0.004944364	0.003018731
119_337.1	0.025399968	0.000681905	0
121_320.5	0.115523472	0.001963578	0.006388028
133_46.5	0.093537334	0.004178571	0.003293161
163_247.9	0.039676307	0.002901396	0.003304352
163_337.4	0.077966985	0.003412353	0
165_230.9	0.044877252	0.00213411	0
165_321.0	0.017898948	0.001792832	0
165_43.4	0.032383974	0.002219697	0.007135182
181_42.3	0.005270267	0.003497082	0.003284225
191_176.4	0.055699771	0.002731506	0

204_46.9	0.008988624	0.002048951	0.00192101
210_44.6	0.009694742	0.002219697	0.001639923
216_43.0	0.003172587	0.002561189	0.005214171
217_43.2	0.034463292	0.002390443	0
278_530.2	0.005646886	0.001878205	0.004937522
302_547.3	0.052815566	0.002561189	0.008232902
306_209.5	0.013979367	0.00230507	0.001084175
318_433.7	0.004973705	0.002816665	0.005223124
326_181.2	0.016297704	0.003667827	0
326_247.9	0.007370505	0.004348887	0.012233796
328_231.5	0.002219869	0.003412778	0.005760868
354_176.4	0.005380272	0.005008755	0.005927319
354_323.2	0.000962123	0.006899834	0
362_247.4	0.000362467	0.00145134	0.000294706
372_250.1	0.000619971	0.002390443	-0.000634658
388_515.4	0.001982654	0.003157301	-0.001975191
390_515.1	0.001573323	0.002134324	0.00192101
406_328.8	0.002194823	0.002560974	-0.00753783
424_380.9	0.001141995	0.003327405	0.000238307
434_279.6	0.000872619	0.002304856	0
440_356.3	0.002080603	0.004859842	0.001349669
442_356.4	0.002934182	0.002390443	0.00082329
448_530.3	0.000395911	0.002561189	-0.00479808
478_319.0	0.003327269	0.002475816	0.006037461
482_269.5	0.002705165	0.001109633	-0.002489843
488_348.4	0.000405449	0.002731506	0.001678028
490_371.4	0.000301747	0.002390443	0.001214085
492_344.6	0.004693607	0.002475816	0.003293161
494_299.4	0.00115183	0.003923305	-0.002249621
516_323.2	0.002053488	0.008191088	0.016071432
518_327.1	0.004382878	0.002561189	0.005488601
526_269.6	0.000133544	0.002730222	-0.004249094
534_211.1	0.001447268	0.002560974	0.008232902
548_530.5	0.001250009	0.000426865	-0.00479808
560_332.3	0.00201752	0.007324567	0
562_243.2	0.006543497	0.00238916	0
562_321.3	0.001128029	0.002475816	0
572_353.1	0.00184087	0.002390443	-0.003873089
574_377.6	0.000271956	0.000510943	-0.002528361
580_230.9	0.001577509	0.002475816	-0.000562405

594_310.9	0.00386742	0.003242032	0.007684042
626_333.6	0.001351263	0.002390443	0.002737643
640_284.1	0.000978763	0.002560546	-0.003376254
652_181.2	0.000725123	0.003072142	0
652_247.9	0.000864124	0.00255969	0
652_319.3	0.001162718	0.002475816	-0.003873089
654_247.9	0.000295869	0.002048951	-0.003873089
655_331.2	0.002237538	0.002646347	0
656_231.0	0.001044332	0.002730222	0.00192101
674_247.9	0.003302076	0.001878205	0
680_179.4	0.000397582	0.001792618	0.002739862
708_176.2	0.000752331	0.003412565	0.004001385
740_274.3	0.001297653	0.002304642	0.002157058
756_258.9	0.003014177	0.001707245	0
834_287.1	0.000854902	0.001707459	0.003842021
876_360.2	0.003156157	0.002390443	-0.001936545
904_381.6	0.000692907	0.00213411	0.001363274
918_353.8	0.000403918	0.001963578	-0.003873089
920_360.1	0.000509524	0.004178996	0.00054886
946_391.7	0.002580331	0.002731292	0.002469871
962_367.5	0.001397738	0.001963578	0.006037461

Declaração

As cópias de artigos de minha autoria ou de minha co-autoria, já publicados ou submetidos para publicação em revistas científicas ou anais de congressos sujeitos a arbitragem, que constam da minha Dissertação/Tese de Mestrado/Doutorado, intitulada **Metabolomics and Machine Learning for Quality Control of Medicinal Plants**, não infringem os dispositivos da Lei n.° 9.610/98, nem o direito autoral de qualquer editora.

Campinas, 29/03/2025

Assinatura : _____

Nome do(a) autor(a): **Elisa Ribeiro Miranda Antunes Vedovatti**
RG n.° 39000881-3

Assinatura : _____

Nome do(a) orientador(a): **Alexandra Christine Helena Frakland Sawaya**
RG n.° 7530826-5

Declaração

As cópias de artigos de minha autoria ou de minha co-autoria, já publicados ou submetidos para publicação em revistas científicas ou anais de congressos sujeitos a arbitragem, que constam da minha Dissertação/Tese de Mestrado/Doutorado, intitulada **Metabolomics and Machine Learning for Quality Control of Medicinal Plants**, não infringem os dispositivos da Lei n.° 9.610/98, nem o direito autoral de qualquer editora.

Campinas, 29/03/2025

Assinatura : _____

Nome do(a) autor(a): **Elisa Ribeiro Miranda Antunes Vedovatti**
RG n.° 39000881-3

Assinatura : _____

Nome do(a) orientador(a): **Alexandra Christine Helena Frakland Sawaya**
RG n.° 7530826-5