Prof. Diego Gragnagniello

Prof. Eduardo Sommella Prof.ssa Vicky Caponigro

Report of Artificial Intelligence for omics data

Apicella Mario Bruno Salvatore

January 13, 2025

UNIVERSITÀ
DEGLI STUDI
DI SALERNO

Contents

1 Introduction

Congenital heart disease (CHD) represents the most common congenital anomaly, affecting approximately 0.63–0.8% of live births in Europe. CHD encompasses a diverse group of structural heart and vascular malformations that vary in their anatomical, clinical, and severity-based classification. Severe forms of CHD often require intervention during the first year of life, representing a significant burden from infancy into adulthood. Despite advancements in post-operative care and increased survival rates, the underlying etiology of CHD remains poorly understood, with over 60% of cases remaining unexplained. Known contributing factors include chromosomal abnormalities such as Down syndrome and environmental influences such as maternal health conditions, but much remains to be uncovered about the interplay between genetics, environment, and metabolism in its pathogenesis.
In recent years, metabolomics has emerged as a valuable tool for unraveling complex biological mechanisms. By providing a global representation of endogenous and exogenous metabolites in biological systems, metabolomics bridges the gap between genotype and phenotype. Metabolic profiles, or metabotypes, reflect the dynamic interaction between an individual's genetic makeup and environmental factors such as diet, lifestyle, and disease states. The maternal metabolome, for instance, offers a promising avenue for exploring CHD risk factors, as maternal metabolic states during pregnancy could influence fetal development and organ function, potentially altering metabolic signatures in maternal blood.

In this study, we leverage advanced metabolomic analysis to explore potential maternal metabolomic risk factors for CHD. The dataset employed integrates two complementary ionization modes— ESI+ (positive ion mode) and ESI- (negative ion mode)— to maximize metabolite detection. ESI+ is optimized for metabolites such as amino acids and peptides, while ESI- excels in detecting acidic molecules like fatty acids and nucleotides. This dual-mode approach provides a more comprehensive view of the metabolome, capturing a wider range of metabolites and their associated biological pathways.

The dataset is structured to include three primary classes of samples: CHD (Cardiac Heart Defects), CTRL (Controls), and QC (Quality Controls). QC samples are de-

rived from pooled plasma to ensure reproducibility, precision, and instrument stability throughout the analysis. The raw data consists of mass-to-charge (m/z) ratios and metabolite intensity values, which undergo preprocessing steps such as normalization, imputation, log transformation, and scaling. These steps ensure that data blocks from ESI+ and ESI- modes are independently harmonized before integration through multiblock chemometric approaches.

A significant aspect of this study involves using multiblock chemometric techniques to analyze and integrate data from the two ionization modes. Unlike simple concatenation, which can introduce biases due to differences in intensity and variance, multiblock methods treat the datasets as complementary sources of information. This approach preserves the unique contributions of each mode, enabling the identification of relevant biomarkers and improving model robustness.

By combining advanced statistical modeling with untargeted plasma metabolomic analysis, this project seeks to identify metabolic signatures associated with CHD. These signatures could provide insights into the biological pathways contributing to CHD risk and serve as a foundation for developing predictive models. In doing so, this study not only advances our understanding of CHD etiology but also paves the way for novel strategies in maternal-fetal health and disease prevention.

2

Materials and Architecture of Methods

2.1 Study Design and Dataset Description

This study employs an observational design to analyze untargeted metabolomics datasets for the identification of potential biomarkers associated with congenital heart disease (CHD). The datasets used in this project consist of metabolomic data acquired through **positive** ion mode (ESI+) and **negative** ion mode (ESI-), which were pre-processed independently to ensure data consistency and accuracy. The integration of these two complementary ionization modes allows for a more comprehensive exploration of the metabolome, capturing a broader range of metabolites across different chemical classes.

2.1.1 Sample Organization

The datasets are organized initially with samples in columns and variables (metabolites) in rows. Each dataset includes:

- A **mass-to-charge ratio (m/z)** column representing the mass spectrometry measurements.
- A **metabolite name** row corresponding to each m/z value.
- Additional metadata columns (e.g., **AF**, **AG**, **AH**) specific to metabolomics analysis, which were excluded from downstream analyses.

2.1.2 Description of the Classes

The datasets are divided into three main classes, each serving a specific purpose in the analysis:

- **CHD (Cardiac Heart Defects):** This class comprises plasma samples collected from individuals diagnosed with congenital heart defects. These samples represent the target group under investigation, with the aim of identifying metabolic alterations or potential biomarkers associated with CHD.
- **CTRL (Controls):** The control group includes plasma samples from healthy individuals without any diagnosed cardiac anomalies. These samples serve as a baseline for comparison, enabling the differentiation of metabolic profiles between affected and unaffected populations.

- **QC (Quality Controls):** QC samples consist of pooled plasma from multiple subjects, designed to represent a broad range of metabolic profiles. These samples were included in the experimental design to:
  - Monitor instrument stability and performance throughout the analysis.
  - Assess reproducibility and precision of the mass spectrometry data.
  - Detect potential drift or variation in the data during the experimental process.

  QC samples were analyzed periodically during the experiment to ensure consistent analytical performance, providing a critical benchmark for quality assurance.

2.2 Data Acquisition and Preprocessing

To ensure high-quality data and reliable analysis, several preprocessing steps were applied. While these steps are summarized here, they were implemented **following a specific pipeline** that will be described in detail in subsequent sections.

2.2.1 Preprocessing Workflow

To ensure data quality and compatibility for downstream analysis, the following preprocessing steps were applied independently to the ESI+ and ESI- datasets:

- **Normalization:** Intensity values were normalized using different robust methods such as median normalization, PQN (probabilistic quotient normalization), TIC (total ion current) or MEDIAN normalization to correct for systematic variability.
- **Missing Value Imputation:** Missing values were replaced with random values between one-fifth and one-fourth of the minimum non-zero value in the dataset.
- **Log Transformation:** A base-10 logarithmic transformation was applied to reduce skewness and improve data symmetry.
- **Autoscaling:** The data were standardized to have zero mean and unit variance, ensuring that all variables contributed equally to multivariate analyses.

2.3 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was performed to evaluate the structure of the dataset, identify underlying patterns, and detect potential anomalies or outliers. A combination of visualization techniques was employed to gain a comprehensive understanding of the data distribution and relationships between variables:

- **Principal Component Analysis (PCA):** PCA was applied to reduce the dimensionality of the dataset while preserving the maximum variance. The analysis utilized both **score plots** and **loadings plots**:
  - **Score Plots:** These were used to visualize class separability and clustering patterns among **CHD**, **CTRL**, and **QC** samples in the reduced principal component space. This allowed for the identification of potential overlaps or distinct groupings between classes.
  - **Loadings Plots:** These plots were examined to identify the variables (metabolites) that contributed most significantly to the separation observed in the score plots. This step was crucial for understanding the metabolic drivers behind the observed patterns.
  - **Explained Variance:** The percentage of variance explained by each principal component was calculated and explicitly shown for every PC. This provided a clear indication of the contribution of each component to the overall variance, aiding in the interpretation of the results.
- **Outlier Detection:** Mahalanobis distance plots were generated using robust covariance metrics, such as the Minimum Covariance Determinant (MCD) and Maximum likelihood estimation (MLE), to identify and exclude potential outliers that could skew the analysis.
- **Histograms:** These visualizations were used to assess the distribution of key features and variability across different classes. This step was critical for identifying skewness and informing subsequent normalization steps.
- **Scatter Plots:** Scatter plots were employed to explore relationships between metabolite intensities given the m/z values, highlighting differences across ionization modes (ESI+ and ESI-) and sample classes.

The combination of these techniques provided detailed insights into the dataset, ensuring a robust foundation for preprocessing and subsequent analytical steps.
These exploratory techniques provided critical insights into the dataset, guiding preprocessing decisions and ensuring robust downstream analyses.

2.4 Multi-Block Integration

Prior to applying the multi-block integration approach, all datasets underwent normalization using the Frobenius norm to ensure consistency and comparability across the data blocks. Additionally, **QC samples**, and **technical replicates** were systematically identified and removed to eliminate sources of variability that could bias the analysis.

The datasets obtained from ESI+ and ESI- modes were then integrated using a multi-block approach. This methodology preserved the unique contributions of each ionization mode while leveraging their complementary information. Following the integration, the datasets were furthemore analyzed and processed
The detailed steps and implementation of this workflow will be discussed in subsequent sections.

2.5 Objective of the Study

The primary objective of this project is to identify potential biomarkers associated with congenital heart disease (CHD) through a comprehensive metabolomic analysis. By leveraging untargeted metabolomics data, this study aims to explore metabolic alterations that could provide insights into the underlying mechanisms and potential risk factors contributing to CHD.

The study employs advanced multivariate statistical techniques and chemometric models to investigate differences in the metabolic profiles of three distinct classes of samples: **CHD (Cardiac Heart Defects)**, **CTRL (Controls)**, and **QC (Quality Controls)**,

The aim of this study was to gain insights into potential maternal metabolomic risk factors for childhood CHD.

2.6 Ethical Considerations

All analyses were conducted following ethical guidelines and approvals. QC samples were intentionally included to monitor data quality and reproducibility.

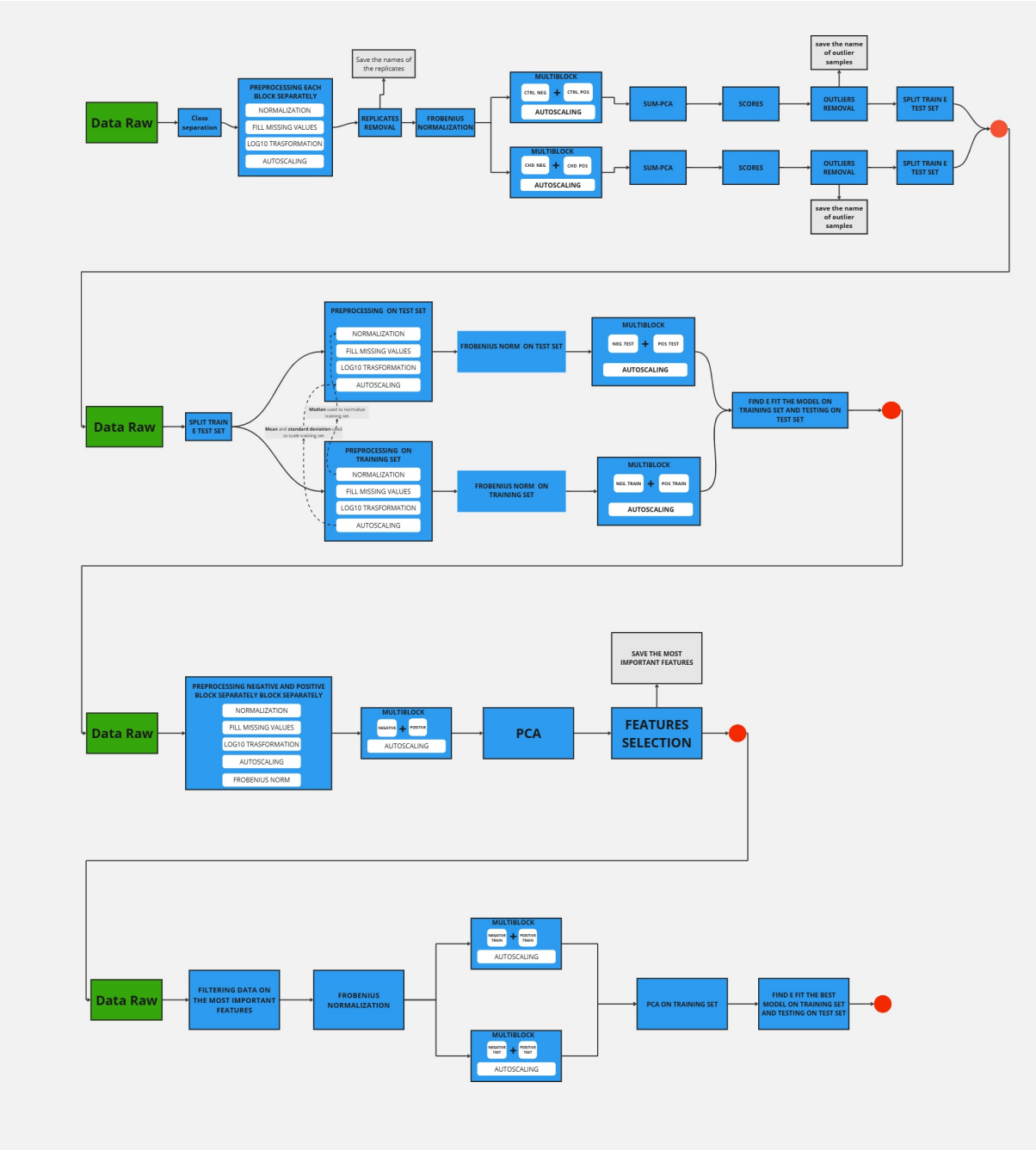
2.7 Next Steps

This section lays the foundation for the subsequent training of classification models and feature selection, which are detailed in the upcoming sections.

3

WorkFlow of statistical metabolomic data analysis

The metabolomics data analysis pipeline initiates with the segregation of raw data from two types of mass spectrometry, ESI+ (Electrospray Ionization Positive) and ESI- (Electrospray Ionization Negative). These datasets are classified into respective classes: CHD Positive (CHD Pos), CHD Negative (CHD Neg), Control Positive (CTRL Pos), Control Negative (CTRL Neg), Quality Control Negative (QC Neg), and Quality Control Positive (QC Pos).



Step-by-Step Process:

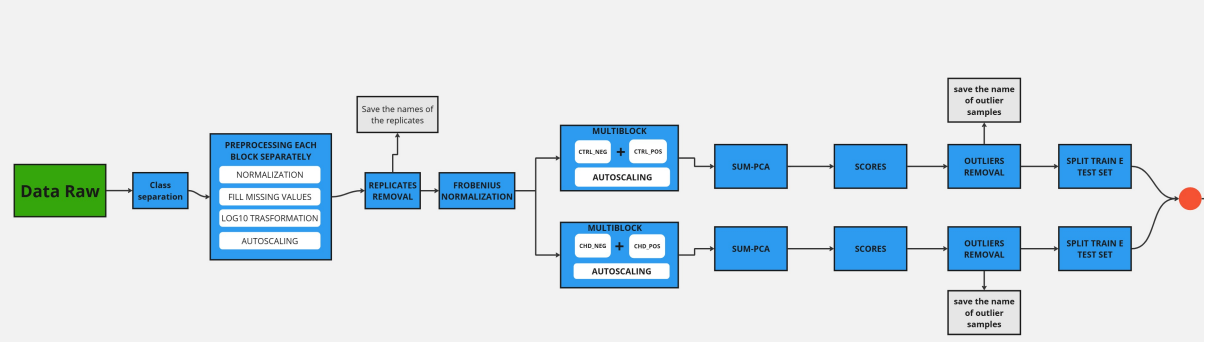


Figure 1: Train e test split

The process outlined in Figure 1 describes a comprehensive approach to handling metabolomics data, characterized by a series of meticulously designed steps to prepare and analyze the data effectively. Initially, data are segregated by class based on their ionization modes (either positive or negative) and physiological markers such as CHD, CTRL, or QC. This segregation ensures that subsequent analyses are specifically tailored to the unique characteristics of each class, allowing for more precise and meaningful insights.

Once segregated, each class undergoes a detailed preprocessing regimen:

1. **Normalization:** Intensity values of the data are normalized using methods like Median or Probabilistic Quotient Normalization (PQN), which are chosen to account for variability across samples effectively.
2. **Fill the Missing Values**Concurrently, missing values are strategically imputed by replacing them with random values that fall between one-fifth and one-fourth of the minimum non-zero value found within the dataset. This method maintains data completeness while preserving inherent variability.
3. **Log10 Trasformation**: To address issues of skewness and heteroscedasticity, a logarithmic transformation (base 10) is applied, significantly improving the data's compatibility with downstream statistical analyses.
4. **Autoscaling**: Furthermore, autoscaling is performed to ensure that each feature contributes equally to multivariate analyses by centering each feature to have zero mean and scaling it to unit variance.

Replicates removal Following preprocessing, replicated samples are identified and removed to eliminate data redundancy. The names of these replicated samples are meticulously recorded and saved in a separate file for future reference and analysis. This step

is critical as it ensures transparency and traceability in the data handling process, allowing for detailed scrutiny in subsequent analytical phases.

Low-Level Multiblock Approach First of all, the **Frobenius norm** is applied to each block, optimizing the numerical stability of the data. This preparation is crucial for effective integration using a low-level multiblock approach. In the **low-level multiblock approach**, data blocks such as CHD Pos with CHD Neg and CTRL Pos with CTRL Neg are concatenated, integrating data across different ionization modes. This concatenation is not merely a process of combining data; it is a strategic step designed to unify metabolic information from both ionization states under each physiological condition. The integration allows for a more holistic analysis and deeper insight into the metabolic interactions that may differ or coincide in positive and negative ion modes.

Purpose of Multiblock Concatenation The primary purpose of this approach is to prepare the data for subsequent advanced multivariate analysis techniques like Sum-PCA. By merging the blocks, we provide a richer and more complete dataset that captures a broader spectrum of metabolic activity, enabling a comprehensive assessment of the metabolic profiles under investigation.

Application of Sum-PCA After the multiblock concatenation and autoscaling, a Sum-PCA is computed on these concatenated blocks. The Sum-PCA plays a critical role by reducing the dimensionality of the data, which enhances data visualization and simplifies the identification of underlying patterns. More importantly, the scores obtained from the Sum-PCA are instrumental in identifying and differentiating the metabolic signatures of the various classes. These scores provide quantitative metrics that reflect the variance and relationships within the data, serving as a basis for further statistical analysis and interpretation.

Outlier Removal Outliers are meticulously identified and removed based on their Mahalanobis distance, employing methods such as the Minimum Covariance Determinant (MCD) and Maximum Likelihood Estimate (MLE). This removal is crucial as it minimizes the influence of anomalous data points that could distort the PCA results and ultimately affect the interpretation of metabolic profiles. To maintain the integrity of subsequent analyses, the names of the outlier samples are recorded and saved in a separate file. By ensuring that the data fed into the Sum-PCA is of the highest quality, the robustness and reliability of the analysis are significantly enhanced.

Training and Testing Set Division Following the removal of outliers, the remaining data is divided into training and testing sets. This division is executed using the Kennard-Stone algorithm, which selects samples to maximize the chemical space covered by the training set. Specifically, 30% of the samples are allocated to the test set, ensuring that both sets are representative of the overall dataset. The names of the samples included in the training and testing sets for each class are meticulously recorded and saved in separate files for future reference. This step is critical for maintaining the transparency and reproducibility of the model training and validation processes.

Through these steps, the pipeline not only prepares the data meticulously for analysis but also ensures that the findings are robust, reliable, and representative of the true biological signals, facilitating a deeper understanding of the metabolic profiles under investigation.

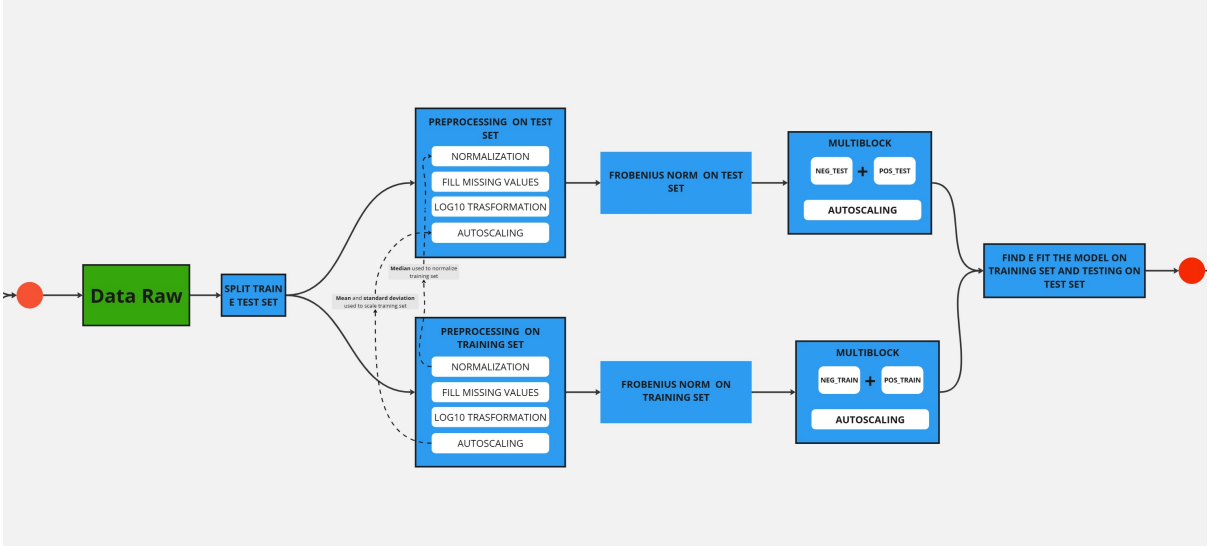


Figure 2: Find the model

With the refined dataset, we now proceed to the second phase of the analysis, illustrates in the figure 2 where we focus on more granular data handling and modeling. Starting with the raw datasets divided into negative and positive and previously transposed and cleaned, we focus on further refinement:

1. **Sample Filtering:** Using the names of samples saved from the initial phase of the pipeline, QC samples, replicates, and outliers are removed. This meticulous filtering, based on previously recorded names, ensures that only the most relevant and accurate data are forwarded for subsequent processing.
2. **Train and Test Division:** The refined samples are divided into training and

testing sets based on previously recorded sample names. The Kennard-Stone algorithm strategically allocates 30% of the samples to the test set, optimizing the representation of the dataset's chemical space. This approach is preferred over random sampling in omics data because it ensures a balanced distribution of key biochemical features between the sets. By systematically selecting samples to maximize chemical diversity, the Kennard-Stone method enhances the generalizability and robustness of the resulting models. It also improves the reproducibility of the study by providing a consistent and methodical framework for sample selection.

Data Preprocessing Both the training and test sets undergo critical preprocessing steps:

1. **Normalization:**
2. **Fill the Missing Values**
3. **Log10 Trasformation**
4. **Autoscaling**

In particular, for the PQN method is employed where the median from the training set is used to normalize both the training and test datasets for each block, ensuring consistency and minimizing bias. Subsequently, during the autoscaling is applied where the mean and standard deviation from the training set are used to scale the test set. This step is crucial to maintain uniform processing conditions across both datasets

Multiblock Approach and Frobenius Normalization First, the Frobenius norm is applied to enhance the numerical stability of the data:

1. **Frobenius Norm Application:** This norm is applied to both the test and training sets to prepare them for effective multiblock integration.
2. **Data Concatenation:** The blocks, such as Neg Test with Pos Test and Neg Train with Pos Train, are concatenated. This step integrates the data across different ionization modes, enabling a comprehensive analysis.

Model Training and Evaluation Following the preparation and integration of data blocks, a series of advanced predictive models are trained, each leveraging a different statistical learning technique:

1. **Models and Training Approaches:**
    - **Support Vector Machine (SVM):** Two SVM models are trained:
        - An SVM model optimized with Grid Search and Cross-Validation (CV) to fine-tune hyperparameters.
        - An SVM model using Leave-One-Out Cross-Validation (LOOCV) for robust evaluation and minimal bias.
    - **Random Forest (RF):** Similarly, two RF models are developed:
        - An RF model utilizing Grid Search and CV for hyperparameter selection.
        - An RF model employing LOOCV to ensure the model's stability and reliability.
    - **Logistic Regression:** Two models of logistic regression are prepared:
        - A model tuned with Grid Search and CV.
        - A model based on LOOCV, focusing on achieving high accuracy and generalizability.
2. **Training Data and PCA Scores:** All six models are trained on PCA scores derived from the training set. The PCA is calculated based on the principal components identified at the knee point of the elbow graph, ensuring that the most significant features are included in the model training process.
3. **Model Evaluation:** Each model is rigorously evaluated using the training set and then validated using the test set. This extensive testing confirms the models' effectiveness and reliability.

These steps ensure that the analysis pipeline not only prepares the data with precision but also utilizes advanced modeling techniques to derive robust, reliable findings that accurately reflect the underlying biological processes. By integrating different types of models and using sophisticated validation techniques, the study establishes a strong foundation for predictive accuracy and model interpretability.
Finally, after models evaluation, all models are saved. This preservation facilitates future applications and further research without the need for retraining, thereby saving resources and ensuring consistency across future analyses.

Following the initial filtering and preprocessing phases, the pipeline progresses to a sophisticated multiblock approach that integrates the preprocessed negative and positive blocks. This stage is critical for combining complementary data from different ionization modes, enhancing the comprehensiveness of the analysis.

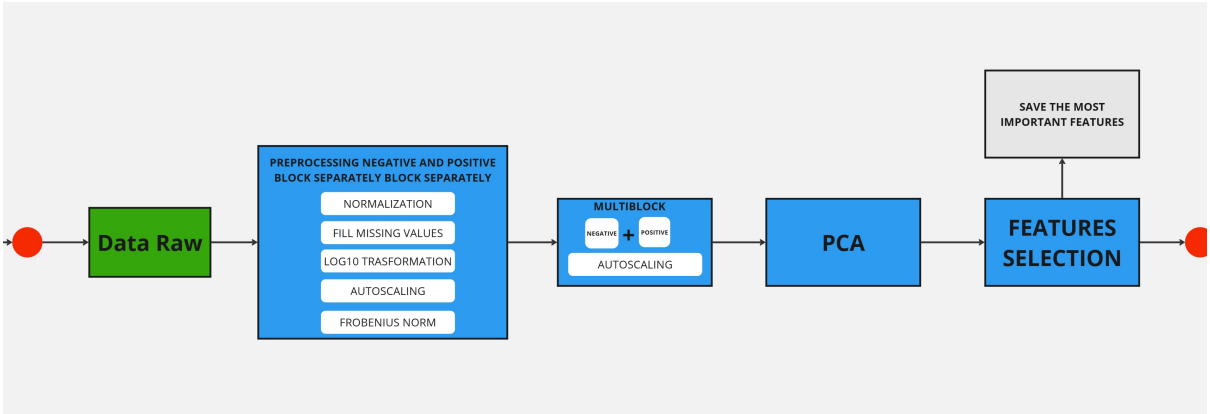


Figure 3: Find the most important features

Preprocessing and Integration

1. **Preprocessing of Filtered Blocks:** The positive and negative blocks, already refined to exclude outliers, replicates, and QC samples, undergo a series of preprocessing steps including normalization, missing value filling, log transformation, and autoscaling. These steps ensure that each dataset is optimally conditioned for high-dimensional integrative analysis.
2. **Frobenius Normalization:** Prior to concatenation, the Frobenius norm is applied to both the positive and negative datasets to enhance their numerical stability. This normalization process ensures that the concatenated data matrix has uniform scaling across all variables, facilitating more reliable multivariate analysis.
3. **Data Concatenation and Autoscaling:** Post-Frobenius normalization, the positive and negative blocks are concatenated, and autoscaling is applied. This step integrates the datasets thoroughly, ensuring that all features are treated equitably in subsequent analysis steps.

Feature Selection Using SHAP and Feature Importances After integrating the datasets, feature selection is conducted to identify the most influential variables affecting the predictive models. This process is crucial for focusing subsequent analyses on the most relevant biological signals:

1. **Feature Selection Methodology:** The top 50 most important features are identified from previously trained models (SVM, RF, and Logistic Regression). This selection is based on commonalities among the leading features across these models, ensuring that only the most predictive and robust features are retained.
2. **SHAP (SHapley Additive exPlanations):** For SVM and Logistic Regression, SHAP values are utilized to determine feature importance. SHAP, based on coop-

erative game theory, allocates an "importance value" to each feature with respect to how much each contributes to the prediction in each model.

3. **Feature Importances for RF:** The Random Forest models utilize their internal feature importance mechanism, which measures the increase in model prediction error after permuting the feature. This approach provides a straightforward metric of feature relevance based on how much the absence of a feature would confuse the model.

Documentation of Features The identified features, deemed most crucial across the models, are saved in a file for further analysis. This documentation facilitates a targeted exploration in subsequent research phases, focusing on those variables that most strongly influence the biological outcomes under study.

Univariate Analysis on Principal Components Following the feature selection, a univariate analysis is conducted on the principal components utilized during the model training. This analysis serves to examine each principal component individually to understand its contribution to the variance in the data and its relationship to the biological processes under study. Here's how the univariate analysis benefits the overall research process:

- **Understanding Variance:** By analyzing each principal component individually, researchers can discern which components capture the most significant biological variance, aiding in the interpretation of complex metabolic pathways.
- **Identifying Key Biomarkers:** This analysis helps to highlight biomarkers or metabolic features that are consistently influential across the principal components, providing insights into potential targets for further biochemical research or therapeutic intervention.

This approach not only leverages the data reduced by PCA for deeper insights but also ensures that the findings from the machine learning models are robust, reliable, and directly relevant to understanding the underlying metabolic mechanisms.

As the analysis progresses, the fourth branch of the pipeline engages with advanced modeling techniques, illustrates in figure 4, utilizing the refined datasets that have been previously filtered to remove QC samples, replicates, and outliers. This branch emphasizes the integration of data, application of multivariate analysis, and validation of predictive models based on selected features.

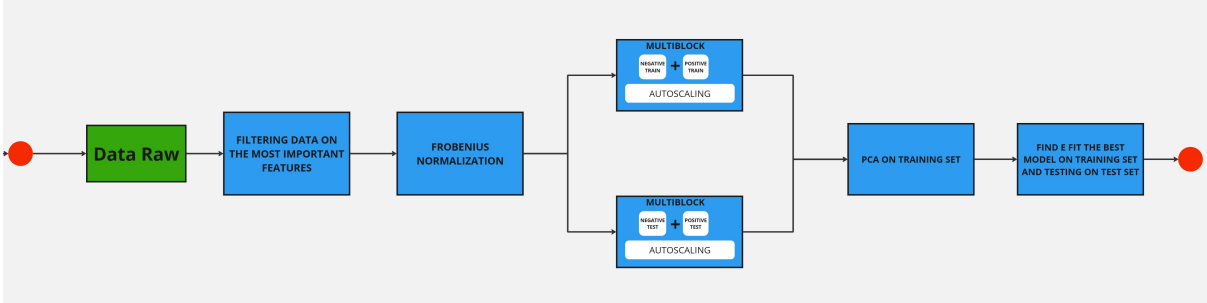


Figure 4: Fit the models on the most important features

Preparation and Integration of Data

1. **Normalization and Integration:** Initially, both the negative and positive datasets, now divided into training and testing sets, undergo Frobenius normalization. This step enhances numerical stability and ensures that each feature contributes appropriately to subsequent analyses.
2. **Multiblock Concatenation:** Post-normalization, the datasets for the training (Negative Train and Positive Train) and testing sets (Negative Test and Positive Test) are concatenated. This integration is crucial for maintaining consistency across ionization modes and ensuring that the data encompass the full spectrum of metabolic variations.
3. **Autoscaling of Concatenated Blocks:** Once concatenated, both the training and testing blocks are autoscaled. Autoscaling adjusts the data to have zero mean and unit variance, which is essential for effective multivariate analysis and for ensuring that no single feature dominates due to scale differences.

PCA and Model Training Based on Important Features After preprocessing and integration:

1. **Feature-Based Model Training:** Utilizing the most important features previously identified and saved from the second branch of the pipeline, a predictive model is trained exclusively on these features within the training set's principal components. This targeted approach ensures that the model focuses on the most biologically relevant and statistically significant features, enhancing predictive accuracy and relevance.
2. **Model Testing and Validation:** The trained model is then tested on the principal components of the test set. This validation step is crucial to assess the model's performance and generalizability to new data, confirming its effectiveness in predicting outcomes based on the selected features.

4 Metabolome Analysis

This chapter delves into the comprehensive analysis of the metabolomic datasets, focusing on the implementation of three key normalization techniques to ensure consistency and interpretability of the data. The research process involved a detailed exploration of the selected normalization methodologies, evaluation of their effectiveness in improving data quality, and identification of the most significant features contributing to the biological insights derived from the dataset.

The three normalization approaches employed in this study include:

- **Probabilistic Quotient Normalization (PQN)**
- **Total Ion Current (TIC) Normalization**
- **Median Normalization**

To facilitate efficient data analysis and management, the workflow was divided into multiple **Google Colab notebooks**, each designed to address a specific aspect of the pre-processing and analysis pipeline. These notebooks systematically guide the process of identifying the most relevant features, optimizing model parameters, and applying machine learning techniques for classification and prediction.

The division into separate notebooks enables a structured approach to managing results, providing a detailed and systematic exploration of the dataset. This organization ensures the reproducibility of the analysis and allows for a focused examination of the outcomes for each normalization technique.

Throughout this study, the three normalization approaches were evaluated for their ability to standardize the data, reduce variability, and enhance the interpretability of downstream analyses. By applying these methods, we aim to uncover meaningful patterns and insights, which will be further elaborated upon in the subsequent sections of this chapter.

4.1 Notebook Organization

To ensure a structured and efficient approach to data preprocessing, analysis, and model development, the workflow was divided into five notebooks. Each notebook is designed to address a specific phase of the project, from raw data handling to feature selection and model training. Below, we outline the purpose and processes performed in each notebook. Furthermore, it has been decided that every notebook and every step of the analysis, when necessary, saves an Excel file containing the dataset modified after preprocessing steps or specific analysis interventions. The dataset obviously follows a rigorous denomination.

N.B. **NormName** in notebooks' name changes considering the used normalization method.

- **Notebook 1: NormName_normalization_notebook**
  This notebook is dedicated to importing and organizing the raw datasets, performing initial preprocessing steps such as normalization, and conducting exploratory data analysis (EDA) to understand the structure and characteristics of the data.
- **Notebook 2: NormName_Multiblock_Train&TestSet**
  The focus of this notebook is on preparing the data for model training by integrating datasets using a multi-block approach, removing unwanted variability (e.g., QC samples, technical replicates, and outliers), and splitting the data into train and test sets.
- **Notebook 3: NormName_FindModel**
  This notebook is used for training predictive models on the processed train and test sets. It includes steps to optimize hyperparameters and evaluate model performance.
- **Notebook 4: NormName_FindMostImportantFeatures**
  This notebook focuses on identifying the most relevant features for classification by applying advanced feature selection techniques such as SHAP and Random Forest analysis on the full dataset.
- **Notebook 5: NormName_Fit_OnMostImportantFeatures**
  The final notebook applies the predictive models to a reduced dataset consisting of only the most important features identified in Notebook 4, ensuring a more focused and efficient analysis.

4.2 Class Extraction and Color Mapping

To facilitate the analysis and visualization of the dataset, a utility function was implemented to extract sample classes and map them into distinct categories. The primary goal of this step was to organize the samples into meaningful groups for subsequent processing and visualization.

The function signature used for class extraction is as follows:

```
    def extract_classes_to_dict(index):
    """
    Extracts sample classes from their names and stores them in a
       dictionary.

    Parameters:
    - index: List or index of sample names.

    Returns:
    - class_dict: Dictionary {class: list of samples}.
    """
```

This function identifies sample classes based on substrings in the sample names and categorizes them into the following groups:

- **QC:** Samples used for quality control purposes, extracted by identifying "QC" in the sample names.
- **CTRL:** Control samples, identified by the presence of "CTRL" in the sample names.
- **CHD:** Samples representing cardiac heart defects, identified by the substring "CHD" in the sample names.
- **m/z meas.:** Although not a class, rows containing "m/z meas." are also identified. This ensures that this row, critical for downstream processing, can be easily distinguished from the sample data.

Importantly, the classification information was not directly added to the dataset as a new column (e.g., *Sample Class*) to avoid altering the structure or dimensionality of the original dataset. This decision ensures:

- Compatibility with downstream analyses that rely on the dataset's original structure.
- Avoidance of introducing unnecessary dependencies or modifications that might interfere with multivariate analyses or pre-processing pipelines.
- Preservation of the raw dataset's integrity, ensuring that any processing steps remain non-destructive and reproducible.

Additionally, a color mapping scheme was applied to enhance the visualization of the dataset:

- **QC Samples:** Represented in red.
- **CTRL Samples:** Represented in orange.
- **CHD Samples:** Represented in green.

This approach ensures that each sample class is clearly identifiable during exploratory data analysis (EDA) and visualization, aiding in the interpretation of the dataset's structure and characteristics.

4.3 Dataset Import and Transposition

The first step in the data preprocessing pipeline involves importing and organizing the metabolomics datasets. This process ensures that the data is structured correctly for subsequent analyses. Specifically, the datasets were transposed to ensure that samples are represented as rows and features (metabolites) as columns, following the standard format for multivariate analysis.

The workflow for dataset import and transposition was as follows:

- **Dataset Loading:** The raw datasets for both **ESI+** and **ESI-** modes were loaded from their respective Excel files. The datasets were initially organized with metabolites as rows and samples as columns.
- **Setting Index:** The "Name" column, which contains metabolite identifiers, was set as the index to ensure the uniqueness and organization of the data.
- **Transposition:** The datasets were transposed so that samples are represented as rows and metabolites as columns. This step aligns the data structure with the requirements of downstream analysis.
- **Cleaning Metadata:** Rows containing metadata, such as summary statistics (**e.g., MEDIA QC, DEV.ST QC, CV% QC**), were removed to focus exclusively on the actual sample data.
- **Saving Transposed Datasets:** The cleaned and transposed datasets were saved in separate files for **ESI+** and **ESI-** modes. This ensures reusability and consistency across different stages of the pipeline.

This step was essential for ensuring that the datasets were correctly formatted and cleaned before proceeding to normalization and exploratory data analysis. By organizing the data into a standardized structure, the pipeline ensures compatibility with advanced statistical and machine learning methods used in later stages.

4.4 Variability Analysis of Metabolomics Datasets

Variability analysis is a critical step in metabolomics to evaluate the consistency and reliability of the data. It allows us to understand the dispersion of metabolite measurements and identify potential inconsistencies that could affect downstream analyses. For this study, variability was analyzed for both the positive and negative ionization mode datasets using two complementary metrics:

- **Standard Deviation (SD):** A measure of absolute dispersion that quantifies the spread of metabolite values around their mean. Higher SD values indicate greater variability.
- **Coefficient of Variation (CV):** A dimensionless relative measure of variability, calculated as the ratio of the standard deviation to the mean. This metric normalizes variability, making it comparable across metabolites with different ranges or magnitudes. It has been decided to make this check at this point in order to have an initial vision on raw data.

4.4.1 Preprocessing for Variability Analysis

Before calculating variability metrics, the datasets underwent preprocessing to ensure that the results were meaningful and representative:

- Metadata rows and columns (e.g., *m/z meas.*) were excluded to focus solely on metabolite intensity values.
- The first row of each dataset, containing non-numerical or metadata values, was skipped.
- All calculations were performed separately for the positive and negative datasets to account for differences in their distributions and ionization modes.

4.4.2 Positive Ionization Mode Dataset

The variability analysis for the positive dataset revealed several metabolites with high CV values, suggesting significant dispersion in their measurements. These metabolites might be subject to biological variability, experimental noise, or outlier effects.

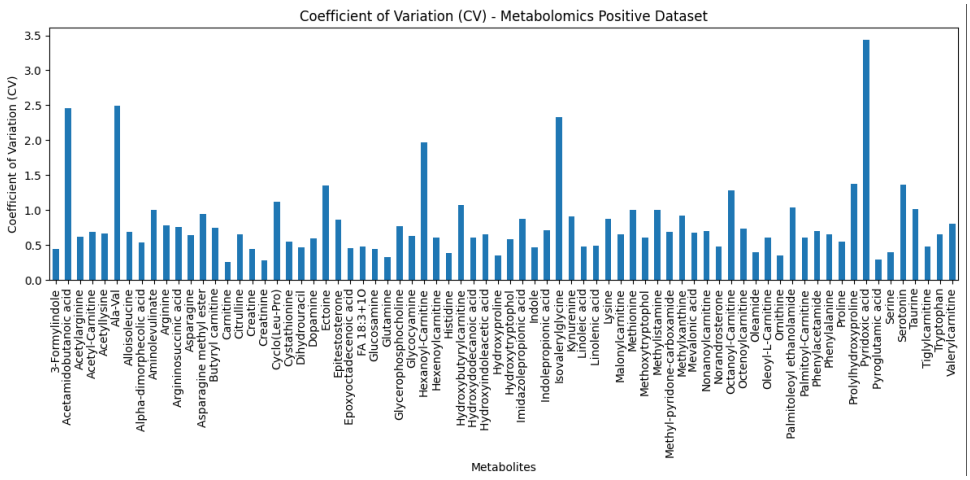


Figure 5: Coefficient of Variation (CV) for the Metabolomics Positive Dataset. Metabolites such as *Prolylhydroxyproline* and *Acetamidobutanoic acid* exhibit elevated CV values, indicating high variability.

Figure 5 demonstrates the distribution of CV values across metabolites, with some metabolites standing out due to their high variability. These findings highlight the need to evaluate these metabolites further for their potential biological significance or experimental inconsistencies.

4.4.3 Negative Ionization Mode Dataset

Similarly, the negative dataset exhibited a range of variability, with certain metabolites displaying notably high CV values. These metabolites may represent outliers or biological markers of interest.

As seen in Figure 6, metabolites with elevated CV values could either represent true biological variability or arise due to technical inconsistencies during sample preparation or measurement.

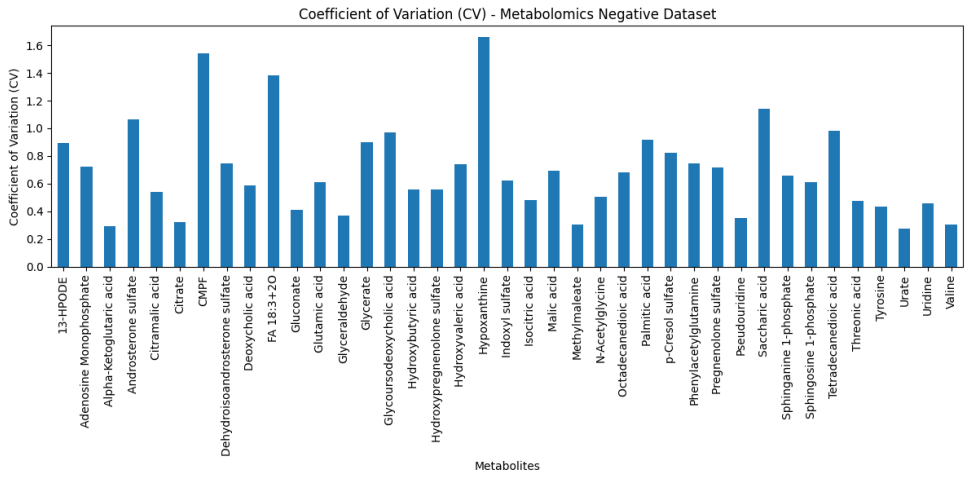


Figure 6: Coefficient of Variation (CV) for the Metabolomics Negative Dataset. Notable metabolites with high CV include *Hypoxanthine* and *Saccharic acid*, which may warrant further investigation.

4.4.4 Key Observations and Implications

The variability analysis provided several critical insights:

- **High CV metabolites:** These may indicate potential outliers, measurement errors, or metabolites with genuine biological variability. They will require additional scrutiny in downstream analyses.
- **Low CV metabolites:** These are associated with high consistency and reliability, making them ideal candidates for model training and biomarker discovery.
- **Differences between ionization modes:** The variability patterns observed in the positive and negative datasets suggest that these modes capture complementary information about the metabolome.

4.4.5 Contextual Importance of Variability Analysis

Understanding variability is crucial for the following reasons:

- **Data Quality Assessment:** High variability in certain metabolites may signal the presence of technical or experimental inconsistencies, such as batch effects or instrument drift.
- **Biological Insights:** Metabolites with high CV values might reflect underlying biological processes or heterogeneity within the sample population.
- **Outlier Detection:** Identifying metabolites with unusually high variability can guide the exclusion of outliers, improving the robustness of downstream statistical and machine learning models.

- **Feature Selection:** Low-variability metabolites, being more stable and reliable, are better suited for feature selection and model building.

4.4.6 Future Steps

The insights gained from variability analysis will inform subsequent preprocessing and feature selection steps. In particular:

- Metabolites with excessively high CV values may be flagged for potential removal or normalization.
- The results of variability analysis will be combined with other exploratory data analysis (EDA) techniques, such as PCA and clustering, to identify patterns and relationships within the data.

This foundational analysis ensures that the datasets are well-characterized, setting the stage for robust and reliable downstream analyses, as will be detailed in subsequent sections of this report. So in next steps other observation will be done on variables

4.5 Data Visualization

Data visualization is an essential step in exploratory data analysis, offering insights into the structure, distribution, and variability of the data. For this study, pre-normalization visualization was performed using scatter plots and histograms to identify trends, variability, and potential outliers.

4.5.1 Scatter Plots

Scatter plots were generated to explore the relationship between m/z measurements and intensity levels across different classes (**QC**, **CTRL**, and **CHD**). This visualization allowed us to assess the intensity distribution of each class in the **Metabolomics Positive** and **Metabolomics Negative** datasets. For a detailed examination:

- The first set of scatter plots showcases all samples from each class, highlighting the variability and clustering within the QC, CTRL, and CHD groups.
- The second set of scatter plots focuses on a randomly selected sample from each class, enabling a closer examination of the m/z and intensity patterns specific to individual samples.

These scatter plots reveal differences in intensity levels across classes and provide insights into the consistency and uniqueness of each class.

4.6 Scatter Plot Analysis for Selected Samples

The presented scatter plots show the distributions of metabolite intensities as a function of m/z values for each class: **QC** (Quality Controls), **CHD** (Congenital Heart Disease), and **CTRL** (Controls). The data represents one selected sample per class from both the **positive** and **negative** datasets.

4.6.1 Purpose of the Visualization

The same visualization approach was applied to each individual sample in the dataset. However, to enhance interpretability and reduce visual complexity, only one representative sample per class is shown. This choice allows us to highlight general patterns and potential outliers associated with each class without overwhelming the reader with excessive information.

The two sets of scatter plots illustrate the distribution of metabolite intensities against m/z values across three classes (**QC**, **CTRL**, and **CHD**) for selected samples. The

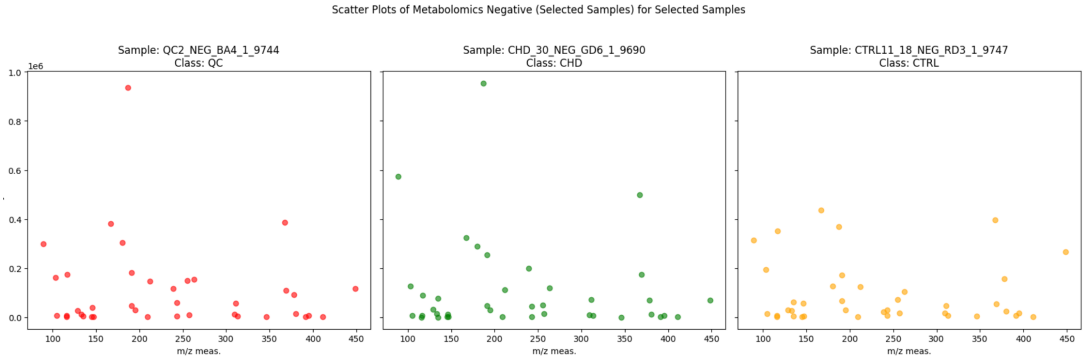


Figure 7: Pre-Norm Visualization Esi- sample

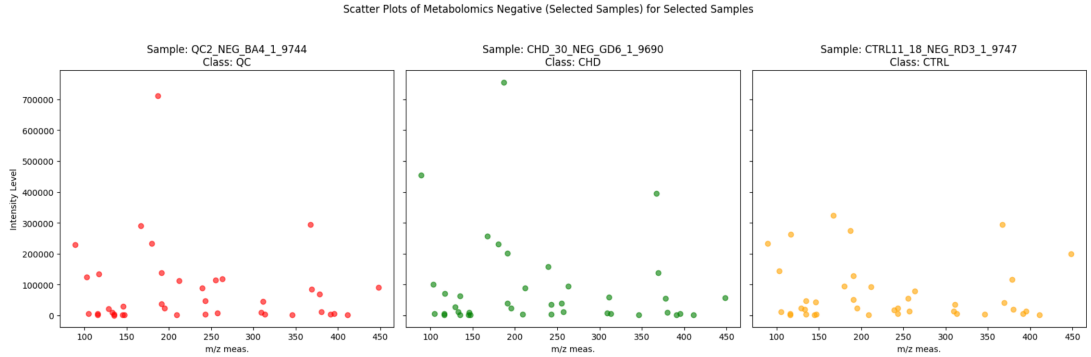


Figure 8: Post-Norm Visualization Esi- sample

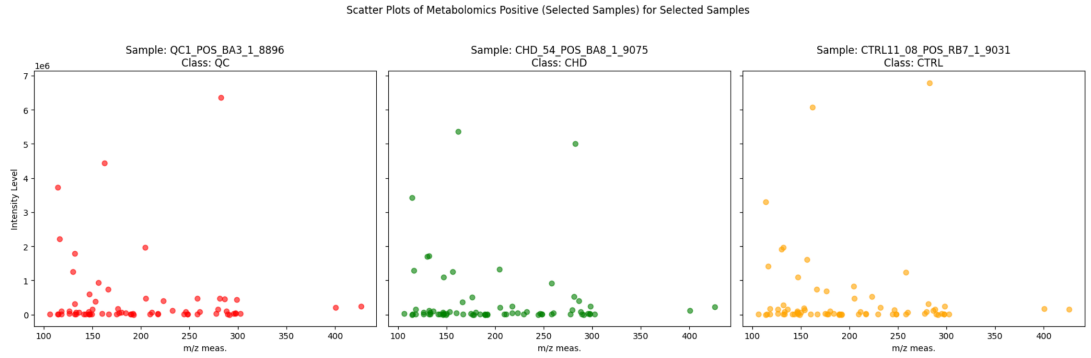


Figure 9: Pre-Norm Visualization Esi+ sample

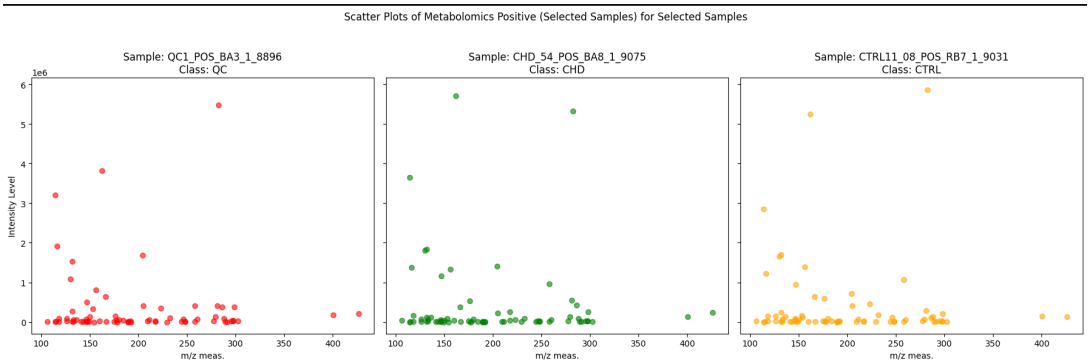


Figure 10: Post-Norm Visualization Esi+ sample

first plot represents data pre-normalization, while the second plot corresponds to post-normalization.

In the pre-normalized scatter plot:

- **Intensity Variability:** The intensity levels exhibit significant variability between samples, with higher peaks observed in specific metabolites, particularly in the **CTRL** and **CHD** classes.
- **Scaling Issues:** The large range of intensity values complicates direct comparison across metabolites and classes, as certain metabolites dominate the distribution.

In the post-normalized scatter plot:

- **Reduced Variability:** Normalization has reduced the overall variability in intensity levels, particularly among **QC** samples, ensuring better comparability across classes.
- **Improved Scaling:** The scaling of the intensity values is more uniform, facilitating direct comparisons between metabolites and reducing the influence of highly abundant metabolites.

4.6.2 Overall Significance

These plots serve as a preliminary tool to understand data distribution and identify general patterns or anomalies.

This visualization is an essential initial step in interpreting metabolomic data and identifying the key characteristics of each class. Subsequent analyses will include statistical and multivariate methods to validate these preliminary observations.

4.6.3 Histograms

Histograms provide an essential tool for visualizing the distribution of metabolite intensities within each class and dataset. They allow us to observe the frequency of different intensity values and identify underlying patterns, such as skewness, spread, and potential outliers. In this analysis, histograms were generated for both the negative and positive datasets, dividing the data by class (**QC**, **CTRL**, **CHD**).

4.6.4 Purpose of Histograms

Histograms serve several important purposes in data analysis:

- **Distribution Analysis:** They provide insights into the shape of the distribution, helping to determine whether the data is normally distributed, skewed, or contains multiple modes.
- **Comparison Across Classes:** By generating separate histograms for each class, we can compare the frequency distributions and detect variations among the classes (**QC**, **CTRL**, **CHD**).

4.6.5 Methodology

The histograms were created using the following steps:

1. **Histogram Generation:** For each class, side-by-side histograms were plotted, including a Kernel Density Estimate (KDE) curve to visualize the smoothed distribution of the intensity values. The plots exclude the row corresponding to "m/z meas." to focus solely on intensity data.
2. **Comparison Between Datasets:** To compare the distributions between the negative and positive datasets, combined histograms were generated, showing both distributions within the same subplot for each class. In this case we'll show histograms separately pre and post normalization, but in the notebook there are also overlapped histograms

Histograms are an essential tool for visualizing the distribution of data within different classes. They provide insights into the frequency and spread of intensity values, allowing for an understanding of the underlying patterns or anomalies in the dataset. In this analysis, histograms were generated separately for the pre-normalized data and the autoscaled data (which underwent a $\log_{10}$ transformation prior to autoscaling).

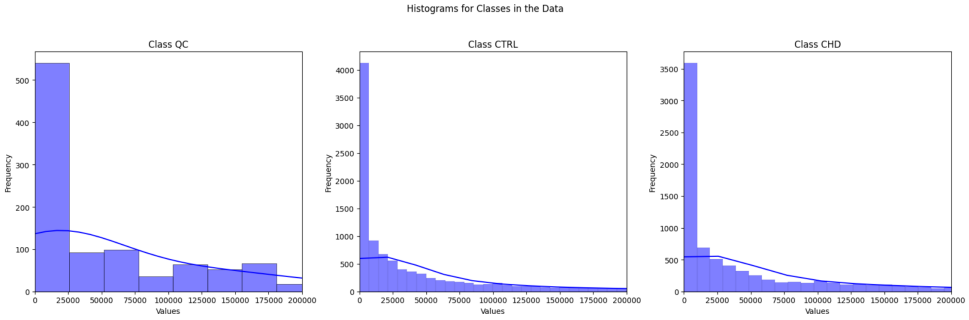


Figure 11: Histogram ESI- Pre Norm

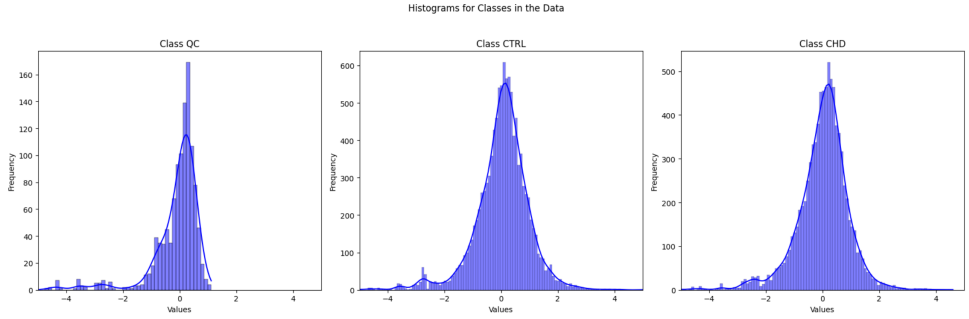


Figure 12: EHistogram ESI- Post Norm

Figure 13: Histogram ESI+ Pre Norm

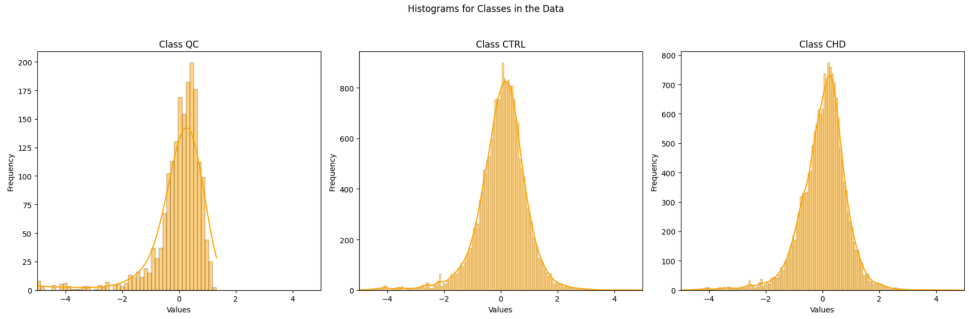


Figure 14: Histogram ESI+ Post Norm

The histograms for the pre-normalized data show a highly skewed distribution. Specifically, intensity values are concentrated in the lower range, with a sharp decline as the values increase. This distribution reflects the original scale and variance of the metabolomic data, which can hinder downstream analyses due to differences in magnitude between features.

Conversely, the second figure displays the distributions after the $\log_{10}$ transformation followed by autoscaling. The autoscaled data exhibit a nearly Gaussian-like shape for all three classes (QC, CTRL, and CHD). This transformation reduces the skewness observed in the pre-normalized data, leading to a more symmetric and standardized distribution centered around zero. Such standardization ensures that all features contribute equally to the analysis, minimizing the influence of outliers and large variances.

Interpretation: The comparison between these histograms highlights the importance of preprocessing steps like $\log_{10}$ transformation and autoscaling in metabolomics data. The transformation ensures that the data follow a distribution that is better suited for multivariate analysis and statistical modeling. For illustrative purposes, we talked about the Negative Dataset; similar trends were observed in the Positive Dataset.

4.7 Normalization with PQN

Probabilistic Quotient Normalization (PQN) is a widely used method in metabolomics for chemical normalization. It ensures that the intensity values across samples are comparable by accounting for differences in sample dilution or other technical variations. In this study, PQN was implemented following the defined pipeline and associated notebooks, ensuring consistency in the preprocessing workflow.
The PQN method works as follows:

- **Reference Sample:** A reference sample is selected, which represents the median profile of the dataset. This is calculated by determining the median intensity for each feature across all samples.
- **Quotient Calculation:** For each sample, the intensity of each feature is divided by the corresponding feature intensity in the reference sample, resulting in a vector of quotients.
- **Normalization Factor:** The normalization factor for each sample is determined as the median of the quotients calculated in the previous step.
- **Scaling:** Each feature intensity in a sample is then divided by the sample-specific normalization factor, effectively scaling the data and mitigating dilution effects.

This process was applied to both **ESI+** and **ESI-** datasets independently, as outlined in the first notebook. After normalization, the data were subjected to further exploratory data analysis (EDA) to assess the impact of normalization and ensure the integrity of the processed dataset.
The PQN approach was chosen due to its robustness in handling variations specific to metabolomics data. By normalizing against a representative reference sample, PQN minimizes systematic biases while preserving the biological variability of interest. This normalization step is crucial for subsequent multivariate analyses, as it ensures that the features are comparable across all samples, improving the reliability and interpretability of downstream models.
The results and impact of PQN normalization will be further discussed in the exploratory data analysis and modeling sections.

Implementation in This Study: The PQN methodology was applied to both the positive (ESI+) and negative (ESI-) datasets. Initially, the mass-to-charge ratio (`m/z meas.`) row was separated from the dataset to prevent interference with the normalization process. After normalization, the `m/z meas.` row was reinserted to maintain

dataset structure.
Key outcomes of this process included the identification of the best reference samples for both datasets, which were:

- **Negative dataset reference sample:** *[QC4_NEG_BA3_1_9858]*
- **Positive dataset reference sample:** *[CHD_74_POS_GB8_1_9146]*

The normalized datasets were saved for subsequent steps in the workflow. The use of PQN at this stage ensures that downstream analyses, such as feature selection and multivariate modeling, are not influenced by systematic dilution effects.

Advantages of PQN: This normalization approach is particularly advantageous in metabolomics studies due to its robustness against extreme values and its ability to account for both systematic and random variability. By leveraging the mode of dilution quotients, PQN minimizes the impact of outliers on the normalization process. The detailed implementation of PQN was carried out following the pipeline defined in the initial stages of this study and is seamlessly integrated into the computational notebooks designed for preprocessing tasks.

4.8 Handling of Missing Values

Missing values are a common challenge in metabolomics datasets and, if not properly addressed, can introduce bias and affect the reliability of downstream analyses. To address this, a tailored imputation strategy was adopted. Missing values were replaced with random values uniformly distributed between one-fifth and one-fourth of the column's minimum value. This approach was designed to preserve the variability and distribution of the data while minimizing the risk of introducing artificial bias.
The row containing "m/z meas." values was excluded from the imputation process to maintain the integrity of the dataset and avoid altering its structure. The rationale behind this choice was to ensure that only sample-related data underwent modification, preserving the core mass-to-charge information critical for metabolite identification.
Imputation was applied both before and after normalization to evaluate its impact on the coefficient of variation (CV). While the standard practice is to handle missing values after normalization, this additional analysis was conducted to illustrate how missing value imputation interacts with data normalization in terms of variability. The results of this comparison are shown in Figures 15 and 16.
The procedure was designed to:

- **Preserve Data Variability:** By generating random values relative to the column's minimum, the imputation aimed to mimic the natural variation of the data.
- **Minimize Bias:** Uniform random imputation reduces the likelihood of over- or under-estimating missing values compared to methods like mean or median imputation.
- **Evaluate Impact:** Conducting imputation both pre- and post-normalization allowed for a detailed assessment of how missing value handling affects data variability.

The comparison of CV across four scenarios—pre-normalization with missing values, pre-normalization with imputed values, post-normalization with missing values, and post-normalization with imputed values—provides insights into the stabilization of variability achieved through imputation.

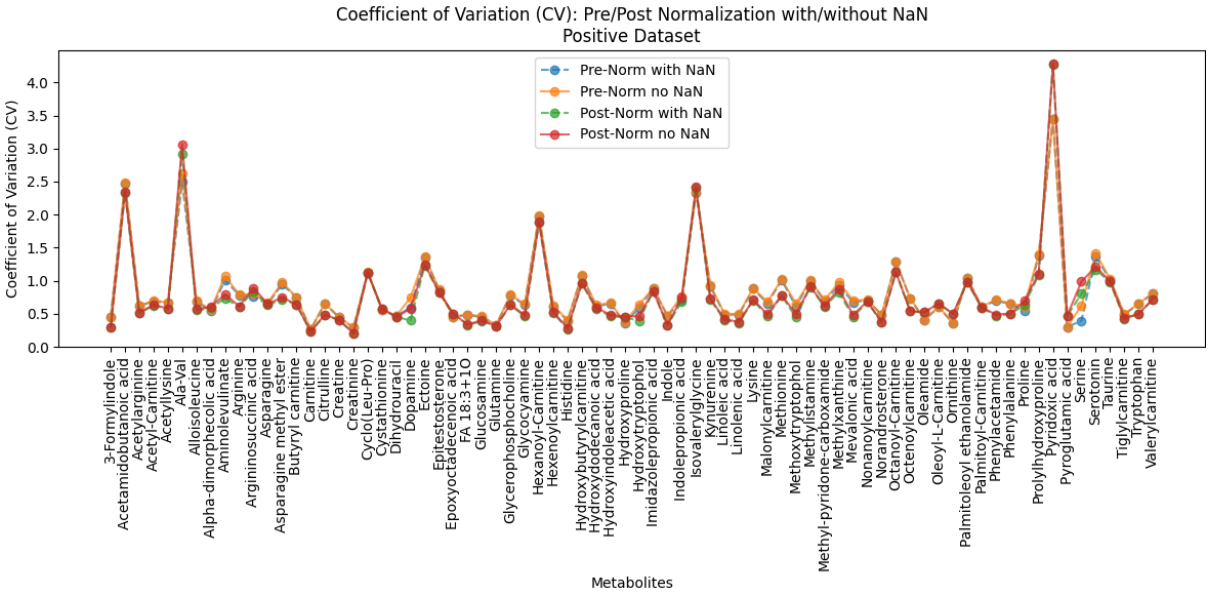


Figure 15: Comparison of Coefficient of Variation (CV) for the positive dataset across four scenarios: pre-normalization with NaN, pre-normalization without NaN, post-normalization with NaN, and post-normalization without NaN.

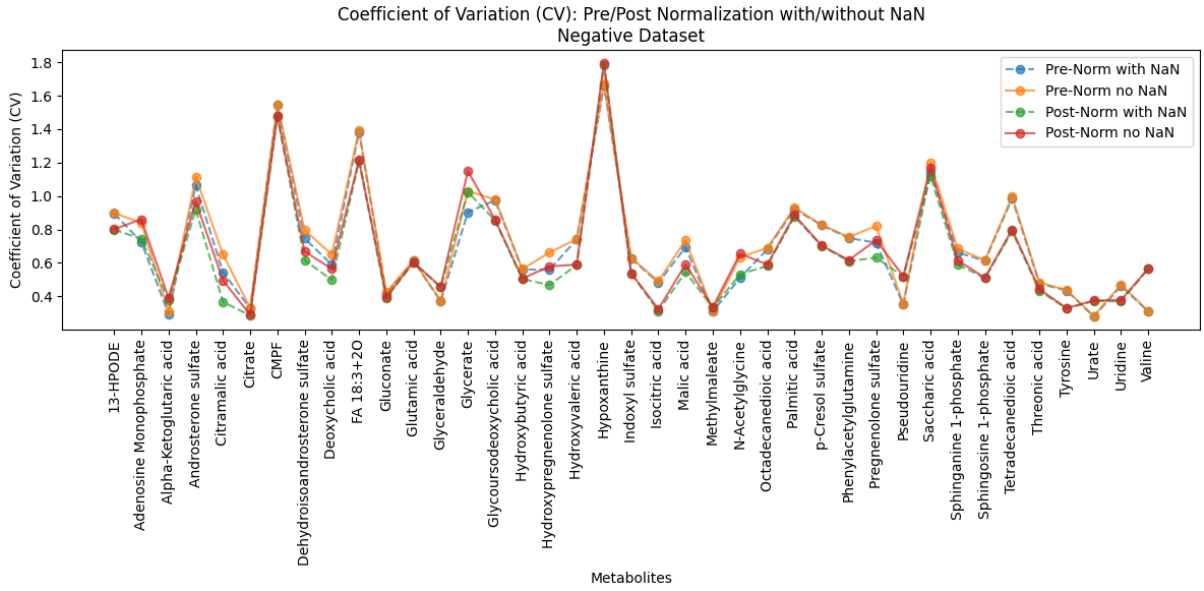


Figure 16: Comparison of Coefficient of Variation (CV) for the negative dataset across four scenarios: pre-normalization with NaN, pre-normalization without NaN, post-normalization with NaN, and post-normalization without NaN.

The results indicate that the imputation method slightly reduces the coefficient of variation (CV), particularly in the post-normalized datasets. However, the overall impact on the dataset remains minimal. This suggests that the chosen imputation strategy effectively stabilizes the data without introducing significant alterations, preserving the original data structure and variability..
It is important to note that the comparison between pre and post-normalization imputation was conducted for illustrative purposes only, as the standard pipeline involves handling missing values after normalization. So we can continue the analysis considering datas normalized and then imputed.

4.9 Logarithmic Transformation

Logarithmic transformation is a fundamental preprocessing step in metabolomics data analysis. It is used to stabilize variance and compress the scale of feature intensities, which often span several orders of magnitude. This transformation makes the data more amenable to statistical analysis by reducing the influence of extreme values.
In this study, a log10 transformation with a constant addition was applied to ensure no negative values in the logarithmic calculation. The transformation process is delineated as follows:

- **Exclusion of Metadata:** The "m/z meas." row, which contains mass-to-charge ratios crucial for metabolite identification, is initially excluded to prevent its alteration during the logarithmic transformation.
- **Transformation Application:** For each numeric value in the dataset, the log10 transformation is applied after adding a constant (typically set to 1) to each value. This adjustment, log10(x + c), ensures that all values are positive, avoiding undefined logarithmic values.
- **Handling Different Classes:** The dataset is segmented according to predefined classes (e.g., positive, negative, QC samples), and the transformation is applied separately to each class to maintain the integrity of their specific distributions.
- **Reintegration of Metadata:** After the transformation, the "m/z meas." row is reintegrated at the top of the dataset, preserving the original structure and ensuring that subsequent analyses can correctly associate transformed intensities with their corresponding m/z values.

This transformation was essential for reducing skewness and heteroscedasticity in the data, which are typical in metabolomics due to the wide range of concentration levels and the presence of highly abundant compounds. By transforming the data logarithmically, we enhance the normality of the distribution, which is beneficial for the linear models and multivariate analysis techniques used later in the pipeline.

Advantages of Logarithmic Transformation:

- **Variance Stabilization:** Log transformation reduces the relative impact of very high values, leading to a more uniform spread of data points across the scale.
- **Enhanced Data Normality:** Many statistical analyses assume normality of the data. Log transformation helps in approximating this assumption, especially when dealing with exponential or multiplicative data behaviors.

The effectiveness of this normalization technique was validated through extensive exploratory data analysis (EDA), examining the transformed data's distribution and variance. These analyses demonstrated significant improvements in the homogeneity and analytical tractability of the data, setting a strong foundation for robust downstream analyses such as clustering and Principal Component Analysis (PCA).

4.10 Autoscaling

Autoscaling, commonly referred to as Z-score normalization, is a critical preprocessing step in metabolomics data analysis that standardizes the features of the dataset to have

a mean of zero and a standard deviation of one. This standardization is essential for many multivariate analyses that assume data features are on the same scale.

The procedure for applying autoscaling in this study, using the provided Python function, is structured as follows:

- **Exclusion of Metadata:** Initially, any metadata rows such as "m/z meas." are removed from the dataset. This exclusion is crucial to ensure that only feature intensities are normalized without distorting critical instrumental data.
- **Z-Score Calculation:** The mean and standard deviation for each feature across all samples are computed. Each feature's intensity value is then transformed by subtracting the mean and dividing by the standard deviation, thereby scaling the data to have unit variance and zero mean.
- **Reintegration of Metadata:** After the scaling process, the "m/z meas." row is reintegrated into the dataset, ensuring the structure remains intact and the mass-to-charge ratios are preserved for subsequent analyses.

This normalization was meticulously applied to both the **ESI+** and **ESI-** datasets, following the detailed steps in the computational workflow. Autoscaling is particularly beneficial in this context because it allows different features (e.g., metabolites) of vastly different concentrations to contribute equally to the analysis, avoiding bias towards high variance features.

Advantages of Autoscaling:

- **Enhanced Analytical Accuracy:** By ensuring that all features have the same scale, autoscaling eliminates the skewness towards features with larger ranges and higher variability.
- **Compatibility with Multivariate Techniques:** Many advanced statistical and machine learning techniques, such as PCA and clustering, assume that all data points contribute equally. Autoscaling facilitates this by normalizing feature scales.

The implementation of autoscaling was crucial for the uniform processing of data, ensuring that no single metabolite disproportionately influences the outcome of multivariate analyses. This step fosters more reliable comparisons and interpretations across samples and conditions.

Following autoscaling, exploratory data analysis was conducted to verify the uniformity and normalization effect across the datasets. These analyses demonstrated significant improvements in data homogeneity, confirming the effectiveness of autoscaling in preparing the data for subsequent analytical steps outlined in the pipeline.

4.11 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique widely used in metabolomics and other high-dimensional data analysis fields to reduce the dimensionality of a dataset while retaining most of the variation in the dataset. This reduction is achieved by transforming the original variables into a new set of variables called principal components (PCs), which are linear combinations of the original variables. The principal components are ordered so that the first few retain most of the variation present in all of the original variables.

The effectiveness of PCA is highly dependent on the normalization of the data, as PCA is sensitive to the relative scaling of the original variables. Different normalization methods can significantly affect the outcome and interpretability of the PCA results.

The PCA plots include three essential components: the **score plot**, the **explained variance**, and the **loadings**.

- The **score plot** visualizes the samples in the reduced-dimensional space defined by the principal components (PCs), allowing the identification of clusters, trends, or outliers. The axes, typically PC1 and PC2, represent the directions of maximum variance.
- The **explained variance** quantifies how much of the total variability in the dataset is captured by each PC. This measure, expressed as a percentage, helps determine the importance of each component and ensures that the reduced-dimensional space retains most of the dataset's information.
- The **loadings** highlight the contribution of each variable (e.g., metabolites or m/z values) to the PCs. Variables with high absolute loading values are the most influential in driving the variance captured by a particular PC, offering insights into the features underlying the observed sample distributions.

These three components collectively provide a comprehensive view of the dataset's structure, making PCA an effective tool for exploratory data analysis.

Pre-Normalization PCA: Before applying any normalization, the PCA captures the inherent variability in the data but might also reflect noise and technical artifacts. The provided figures (not displayed here) would typically show wider spreads in the scores plots, indicating a higher apparent variability that might not necessarily correspond to biologically relevant differences. This initial PCA is crucial for assessing the distribution of samples in the reduction space, highlighting the influence of scale differences, outliers, and technical variations inherent in the raw data. This step is instrumental in identifying the need for normalization to ensure that subsequent analyses are not skewed by these non-biological factors. By examining the PCA results before normalization, researchers can better understand the baseline characteristics of the dataset, which helps in selecting the most effective normalization method to enhance data quality and interpretability.

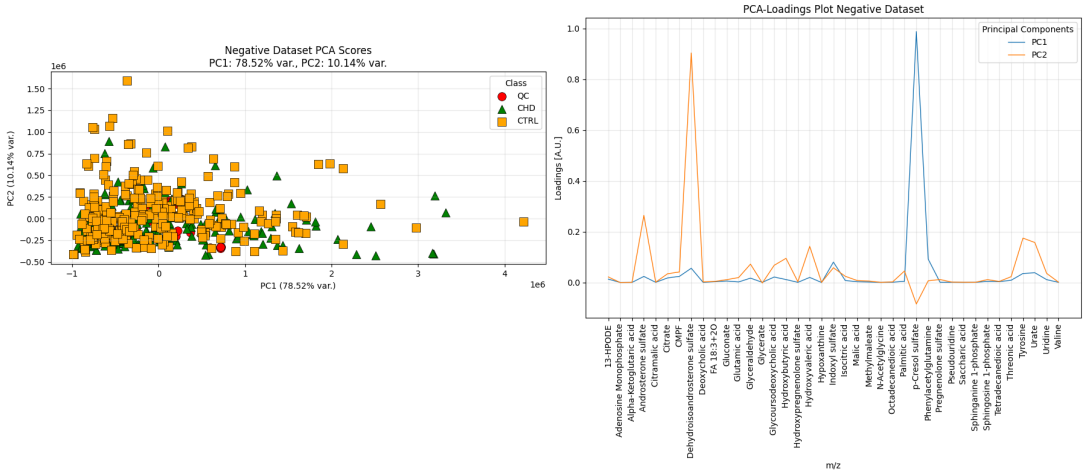


Figure 17: PCA Pre Normalization ESI- along PC1 and PC2

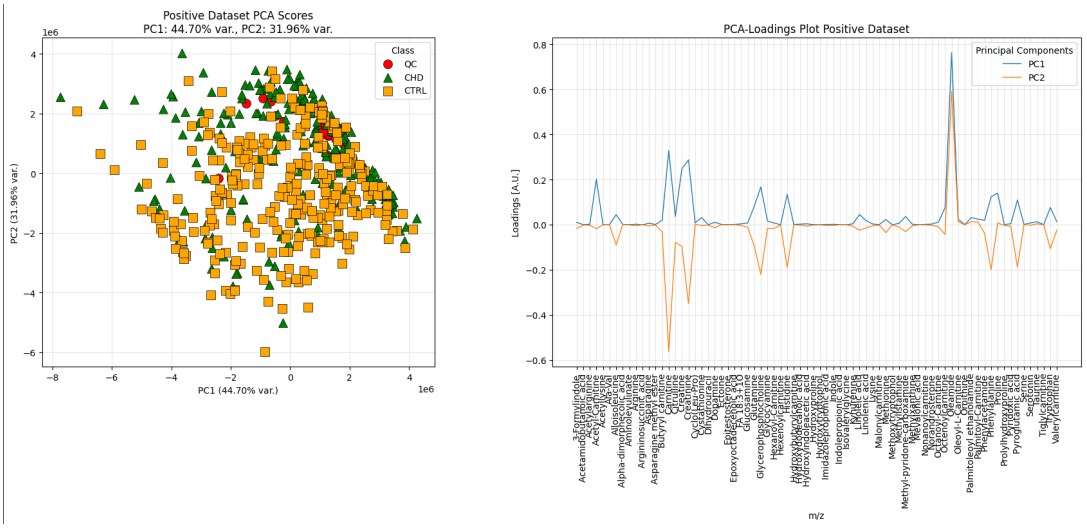


Figure 18: PCA Pre Normalization ESI+ along PC1 and PC2

Post-Normalization PCA with PQN: After applying Probabilistic Quotient Normalization (PQN), PCA tends to show a more compact clustering of samples, reflecting the reduction in technical variability. The loadings plot will illustrate which metabolites (m/z values) contribute most to the variance observed in the PCA, indicating the features that are most stable and potentially biologically relevant after normalization.

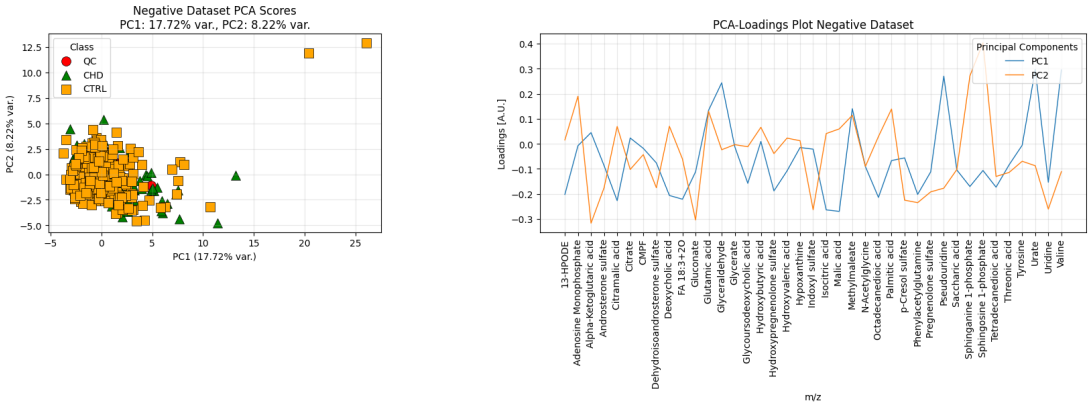


Figure 19: PCA Post Normalization ESI- along PC1 and PC2

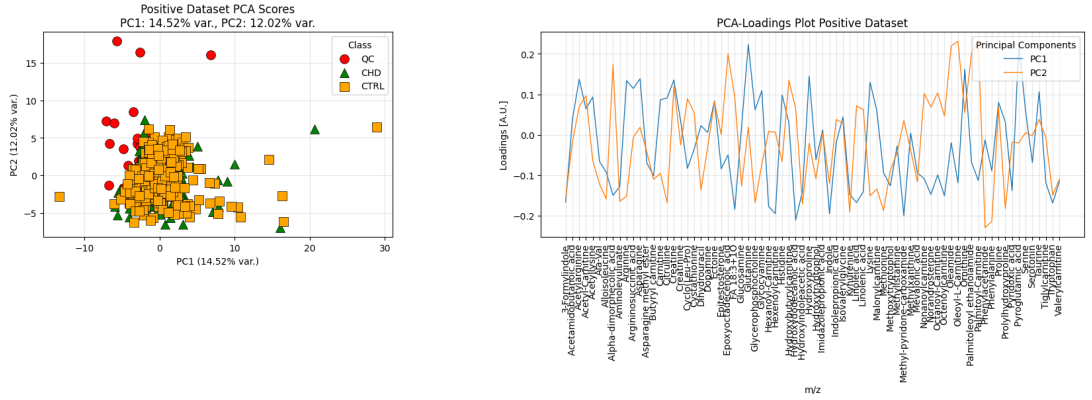


Figure 20: PCA Post Normalization ESI+ along PC1 and PC2

The PCA Scores Plot demonstrates increased dispersion along PC1, with a reduced explained variance, indicating that the normalization distributed variability more uniformly across the principal components.

The PCA Loadings Plot highlights changes in the contribution of features to the components, implying that normalization altered the relative importance of individual features.

This analysis suggests that the new normalization method introduced greater complexity into the dataset by redistributing variability across dimensions. However, the actual

impact on class discrimination (CHD vs. CTRL) needs further evaluation, such as analyzing model accuracy or the relevance of selected features.

Elbow Graph in PCA: The elbow graph is a crucial tool in principal component analysis (PCA) used to determine the optimal number of principal components to retain for further analysis. This graph plots the percentage of explained variance against the number of components. The goal is to identify the point where the addition of further components does not result in a significant increase in explained variance—this point is referred to as the "elbow."
The significance of the elbow graph lies in its ability to visually represent the trade-off between complexity and information gain. By focusing on the components before the elbow, researchers can reduce the dimensionality of the dataset while preserving the majority of the information. This reduction not only simplifies the dataset, making it more manageable but also minimizes noise and the potential for overfitting in predictive models.

In practical terms, the elbow graph helps in making informed decisions about how many principal components should be used to adequately capture the underlying structure of the data without unnecessary complexity. This is particularly important in metabolomics and other high-dimensional datasets, where interpreting and visualizing data can be challenging due to the large number of measured variables.

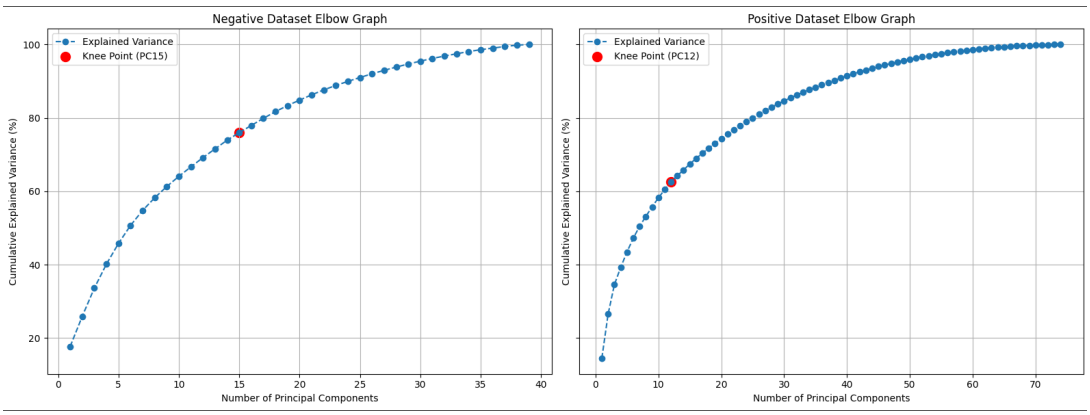


Figure 21: Elbow Graph Post Normalization Both Datasets

From the graph we can understand we could consider 15 PCs for ESi- and 12 PCs for ESi+

4.12 Pairplot Analysis of Positive Dataset Principal Components

The figure illustrates the pairplot of the first five principal components (PCs) for the positive dataset, obtained after applying PCA. Each subplot represents the relationships between two PCs, while the diagonal plots showcase the distributions of individual PCs.

4.12.1 Key Observations

- **Class Separation**: *QC* samples (red) are visually distinct from the CTRL (blue) and CHD (green) classes, indicating the successful isolation of quality control samples. However, significant overlap between the CTRL and CHD classes is evident across most PC pair combinations, reflecting the intrinsic complexity of distinguishing these classes in the positive dataset.
- **Principal Component Variability**: The variance captured by PC1 dominates, as shown by its broader spread in the pairwise plots and its density distribution. Subsequent PCs (PC2, PC3, etc.) exhibit reduced variability, indicating their smaller contribution to the overall dataset variance.
- **Cluster Distribution**: Both CTRL and CHD classes form loose clusters with notable overlap. Some subtle differences can be observed in PC pairings such as (PC2, PC3) and (PC4, PC5), but the separation remains insufficient for clear class discrimination.
- **QC Outliers**: A few QC samples appear as outliers in higher-order PCs, indicating potential noise or unique characteristics within the QC dataset.
- **Density Plots**: The diagonal plots reveal broader distributions for PC1 compared to higher-order PCs, which exhibit tighter distributions. This supports the interpretation that PC1 carries the most significant portion of the dataset variance.

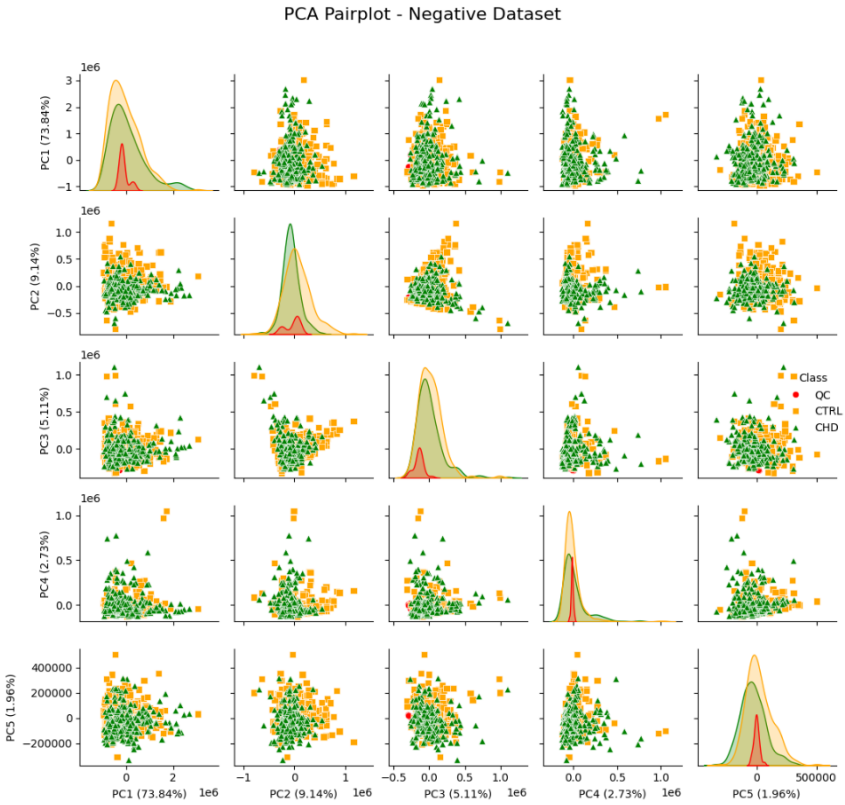


Figure 22: Pair Plot Esi-

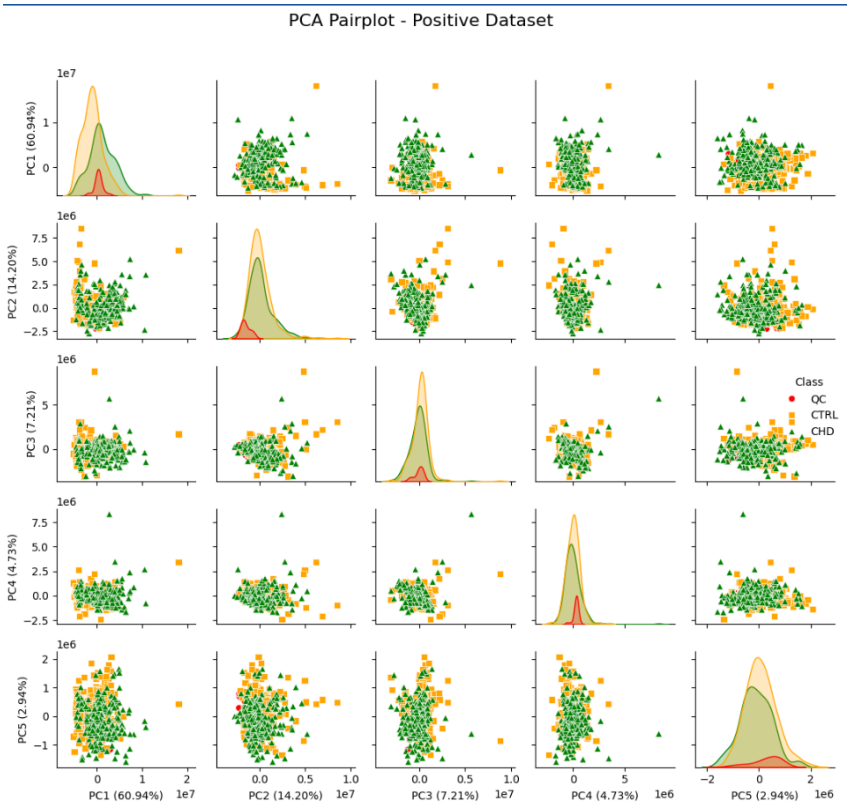


Figure 23: pair Plot Esi +

4.12.2 Conclusion

The pairplot reveals that, while QC samples are well-isolated, the CTRL and CHD classes remain significantly overlapping, making their separation challenging. This overlap necessitates further feature selection or advanced modeling to achieve better class discrimination in the positive dataset. The dominant role of PC1 is evident

4.13 Notebook 2: Multiblock Analysis

In this notebook, we continue the data analysis by applying previously explained steps, avoiding unnecessary repetition of detailed explanations for clarity. While these steps are revisited frequently, they serve to maintain consistency in the preprocessing and analytical workflow.

4.13.1 Class Separation and Preprocessing

For both the negative and positive datasets, the samples were separated by their respective classes (CHD, CTRL). After this separation:

- **Quality Control (QC) Removal:** Samples belonging to the QC class were removed to focus exclusively on the biological variability between CHD and CTRL classes.
- **Technical Replicates:** Technical replicates were identified within each class. These are repeated measurements of the same biological sample, typically used to evaluate technical reproducibility and detect potential experimental noise. All replicates have been saved in a file in order to keep track of their name.

4.13.2 Frobenius Norm for Technical Replicates

To consolidate technical replicates into a single representative measurement, the *Frobenius norm* was applied. This method calculates a representative value by considering the overall magnitude of replicate measurements, ensuring that the combined data reflects the true biological signal while minimizing noise. The Frobenius norm is particularly effective in combining replicates for multivariate datasets, as it considers the entire data structure.

The Frobenius norm of a matrix X is defined as:

$$\|X\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |x_{ij}|^2}$$

where x_{ij} represents the elements of the matrix X. This norm effectively measures the overall magnitude or energy of the matrix.
To scale a matrix using the Frobenius norm, each element of the matrix is divided by the norm itself:

$$X_{\text{scaled}} = \frac{X}{\|X\|_F}$$

This normalization ensures that all matrices are brought to a similar range, preserving their relative patterns while eliminating absolute magnitude differences. By applying the Frobenius norm, the alignment of data blocks becomes more robust, facilitating effective data fusion and ensuring that no block disproportionately influences the analysis.

4.13.3 Multiblock Low-Level Integration

After preprocessing, the datasets were prepared for **multiblock low-level integration**, a technique that combines data from multiple sources. For this analysis:

- The **CHD** samples from the negative and positive datasets were combined.
- Similarly, the **CTRL** samples from the negative and positive datasets were integrated.

After the muiltiblock combination datas have been again autoscaled

4.13.4 Sum-PCA for Combined Blocks

To analyze the combined blocks, a *Sum-PCA* approach was applied. **Sum-PCA** is a variation of principal component analysis designed for multiblock data, where blocks are concatenated by summing their respective contributions. This approach allows the integration of information from different datasets (e.g., negative and positive) while retaining their unique contributions. The resulting PCA scores and loadings provide insights into the shared and individual variability between the blocks, facilitating interpretation across multiple data sources.

In the following sections, we will proceed with the steps outlined above, maintaining a focus on streamlined analysis without redundant explanations. This approach enables us to efficiently evaluate the data while ensuring methodological consistency.

4.13.5 PCA on Multiblocks and Outlier Detection

After combining the multiblocks for each class (**CHD Neg + CHD Pos** and **CTRL Neg + CTRL Pos**), Principal Component Analysis (**PCA**) was performed to reduce the dimensionality and explore the data structure. For each computed PCA, the following were generated:

- **Score plot**: displays the distribution of observations in the space defined by the first two principal components (PC1 and PC2).
- **Loadings plot**: highlights the metabolites contributing most to the variance explained by PC1 and PC2.
- **Explained variance**: represents the percentage of variance explained by each principal component, indicating the quality of the dimensionality reduction.

4.13.6 Outlier Detection Using Mahalanobis Distance

Outliers were identified using the **Mahalanobis distance**, leveraging two main approaches:

- **MCD (Minimum Covariance Determinant)**: provides a robust estimate of the covariance, less sensitive to outliers.
- **MLE (Maximum Likelihood Estimation)**: uses a traditional covariance estimation approach.

Threshold for Outlier Identification To distinguish between inliers and outliers, a threshold was defined based on a 99% confidence level using the chi-square distribution

Outlier Visualization For each class, the results were visualized through:

- **Score plot with Mahalanobis contours**:
  - *Inliers* (observations within the threshold) are shown as black points.
  - *Outliers* (observations exceeding the threshold) are shown in red.
  - Contours for both MCD and MLE illustrate the estimated distribution of Mahalanobis distances.
- **Boxplots of Mahalanobis distances**:
  - Distances are represented using their cubic root for improved visualization.
  - Separate boxplots for inliers and outliers are displayed for both MCD and MLE estimates, highlighting differences in the distribution.

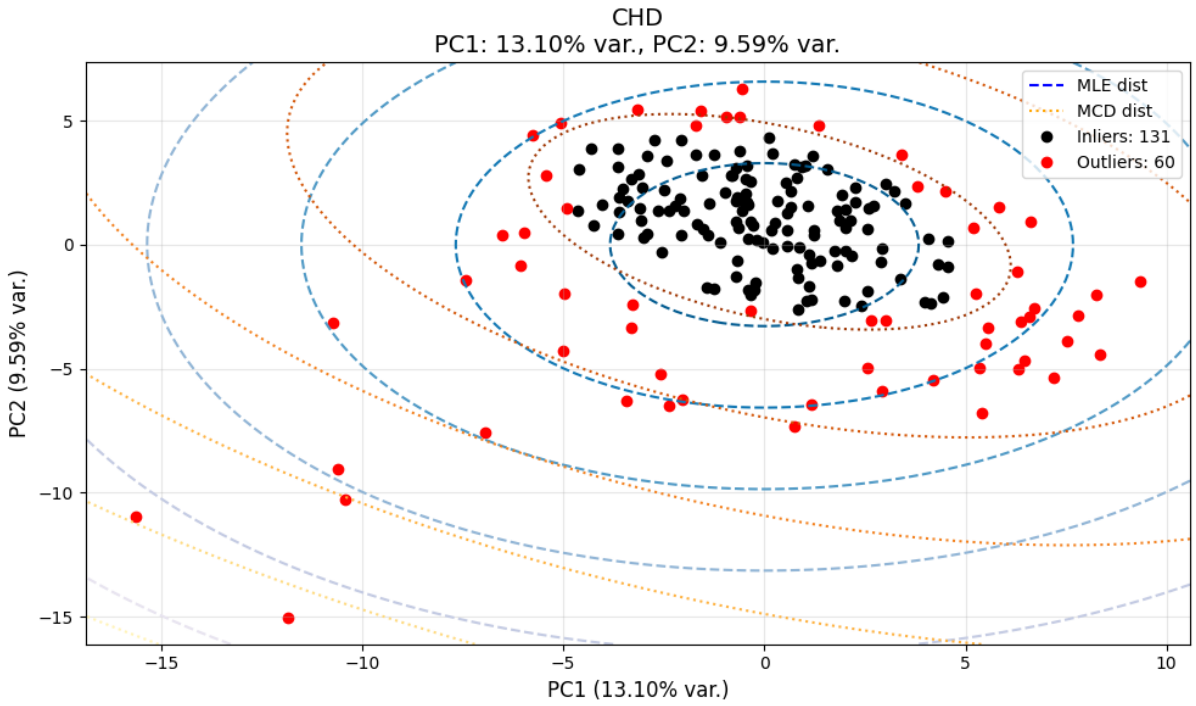


Figure 24: Contours for both MCD and MLE illustrate the estimated distribution of Mahalanobis distances

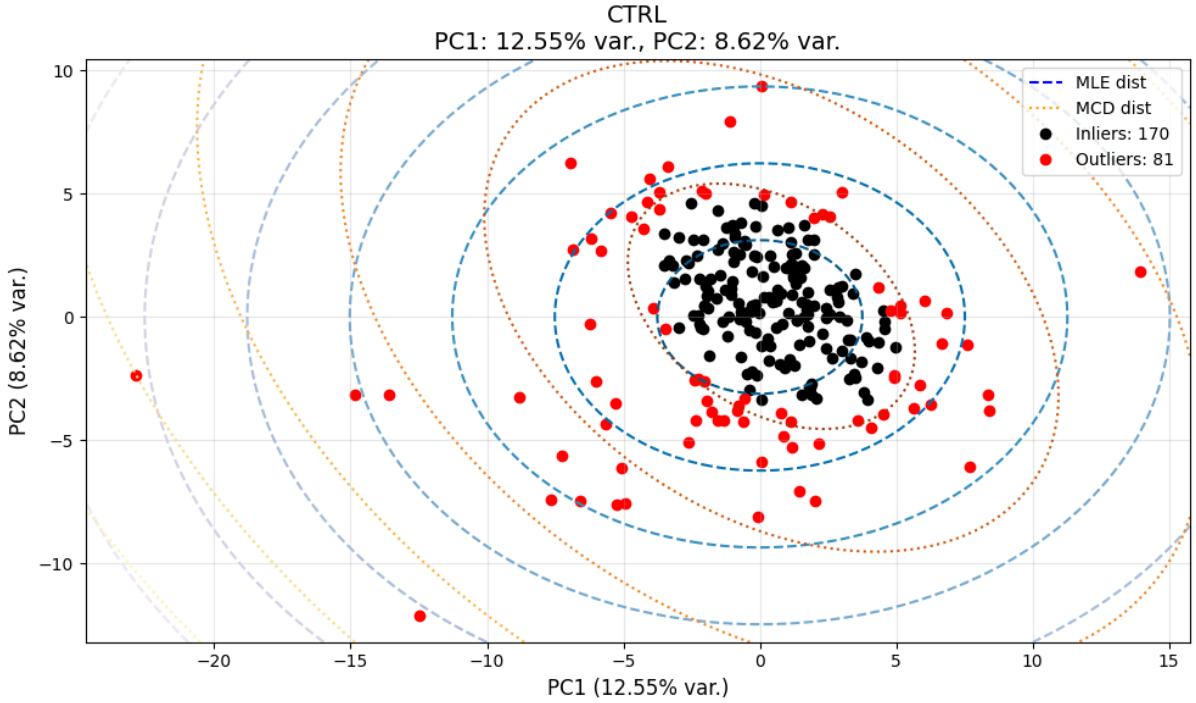


Figure 25: Contours for both MCD and MLE illustrate the estimated distribution of Mahalanobis distances

Conclusion Outlier detection helps to remove observations that could distort the interpretation of the data and ensures robust and reliable analysis.

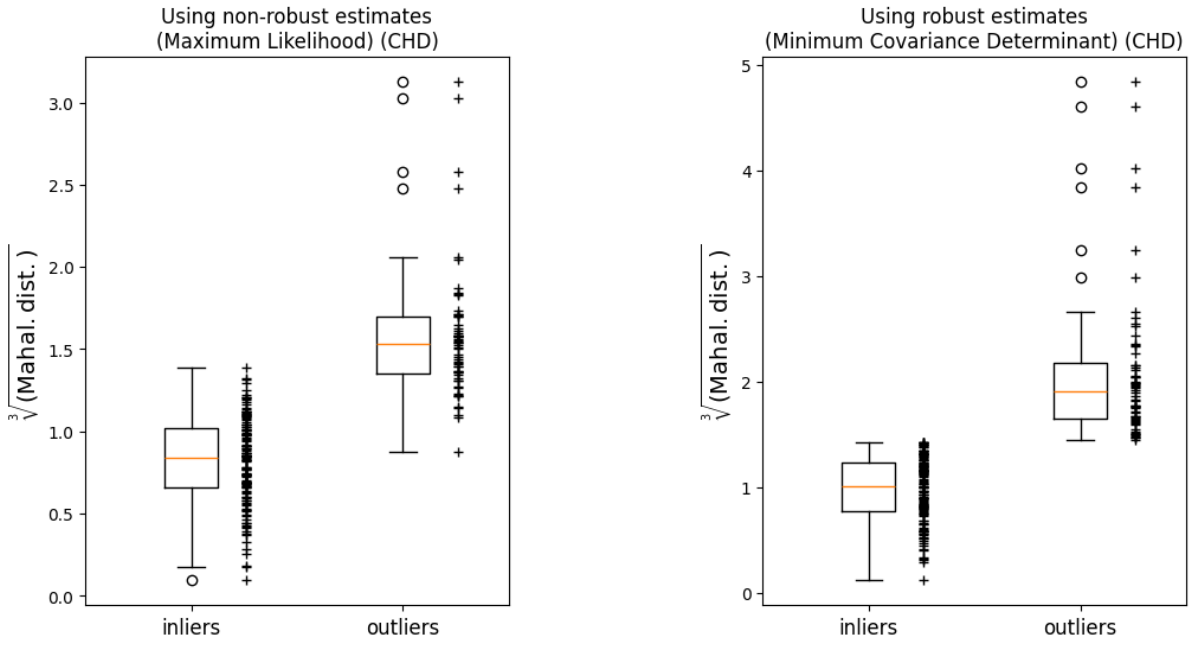


Figure 26: Box-plot CHD class

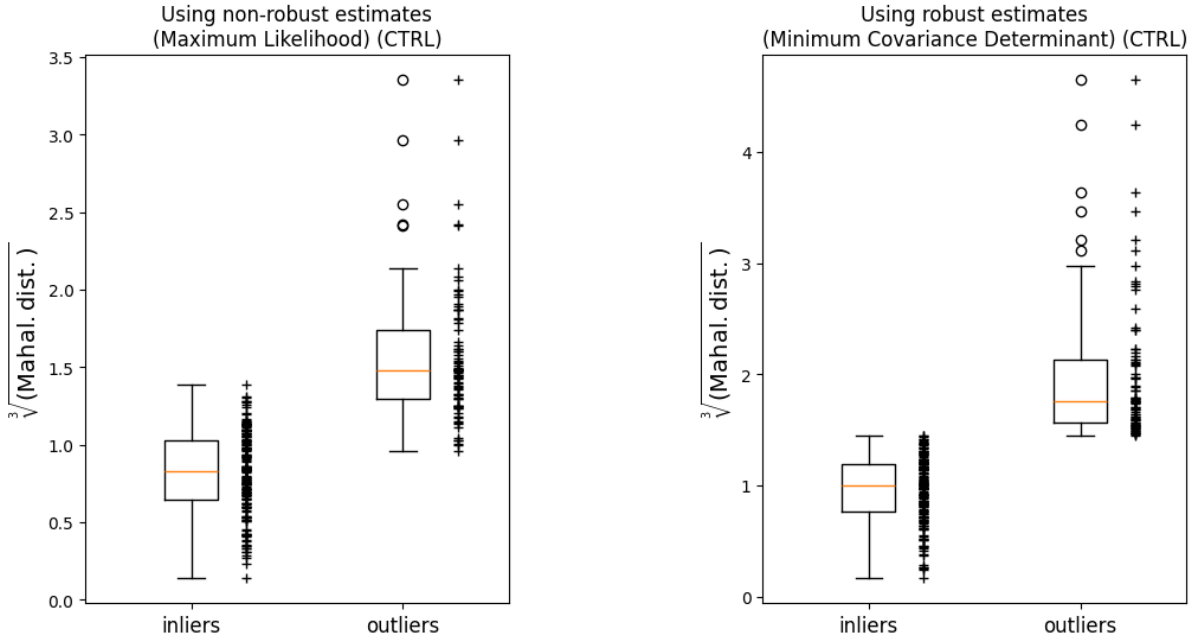


Figure 27: Box-plot CTRL class

4.14 Train and Test Set Split

The dataset was divided into training and testing sets using the Kennard-Stone algorithm, a method that ensures the training set is representative of the data's variability by maximizing the distance between selected samples. This approach helps to preserve the distribution of classes (**CHD**, **CTRL**, and **QC**) in both the training and testing sets.

The split was performed on the PCA scores of the first two principal components (PC1 and PC2), with 70% of the samples allocated to the training set and 30% to the testing set. The resulting sample names were saved for reproducibility.

A scatter plot was generated to visualize the distribution of training and testing samples in the PCA score space, as shown in Figure 29. This ensures a clear representation of the split and the inclusion of variance captured by PCA.

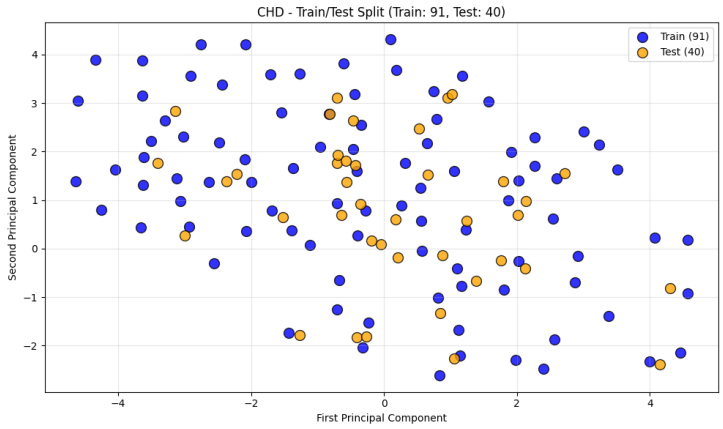


Figure 28: Visualization of train and test set split in PCA space for CHD. Training samples are in *blue*, and testing samples are in *orange*.

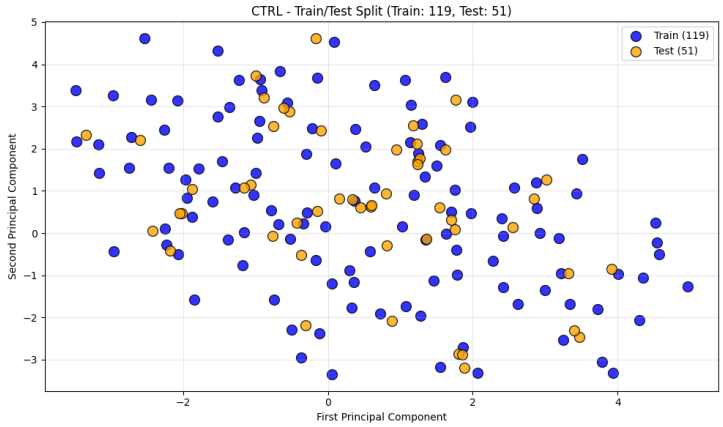


Figure 29: Visualization of train and test set split in PCA space for CTRL. Training samples are in *blue*, and testing samples are in *orange*.

4.15 Notebook 3: Preprocessing and Raw Data Splitting for Model Training

This notebook starts by returning to the raw datasets, from which **QC samples** were excluded. Additionally, using the previously saved files containing information about **technical replicates** and **outliers**, these samples were systematically removed to ensure a clean dataset for further analysis.

The cleaned datasets were then split into **negative ion mode** (train and test sets) and **positive ion mode** (train and test sets), adhering to the same sample allocations determined in the previous notebook. This ensures consistency in the selection of samples for model training and evaluation.

Subsequently, preprocessing steps were applied:

- **Training Set Preprocessing:** Probabilistic Quotient Normalization (PQN) was applied to the training set to correct for dilution effects, followed by autoscaling to bring all variables to zero mean and unit variance.
- **Test Set Preprocessing:** The test set was normalized using the **median** computed from the PQN of the training set. For autoscaling, the **mean and standard deviation** calculated from the training set were applied to standardize the test set. This approach maintains consistency between the datasets and avoids information leakage from the test set into the training set.

These steps ensure that both the training and test sets are processed uniformly while preserving the integrity of the training data for unbiased model evaluation.

After preprocessing, the **Frobenius norm** was applied separately to the four datasets: **negative train**, **negative test**, **positive train**, and **positive test**. This normalization step ensures that each dataset is scaled appropriately, allowing for effective integration in the subsequent steps.

Following the Frobenius normalization, the datasets were combined using a **multi-block low-level integration approach**. Specifically:

- **Training Integration:** The **negative train** and **positive train** datasets were merged into a single multi-block training set.
- **Testing Integration:** The **negative test** and **positive test** datasets were similarly merged into a single multi-block testing set.

This integration approach preserves the complementary information provided by the two ionization modes while maintaining the distinction between training and test sets.

These integrated datasets will serve as inputs for model training and evaluation in subsequent steps.

4.16 Model Training and Hyperparameter Optimization

With the preprocessed and integrated datasets prepared, the next step involves the selection and training of **three classification models**. **The primary objective of this notebook** is to identify the optimal hyperparameters for each model, ensuring their performance is maximized for the given task.
To achieve this, we utilized **two cross-validation techniques**, which are critical for assessing and optimizing model performance:

- **Purpose of Cross-Validation:** Cross-validation is a cornerstone in machine learning, allowing us to evaluate a model's ability to generalize to unseen data. This is achieved by partitioning the training data into subsets, training the model on some subsets, and testing it on others. This iterative process provides robust estimates of the model's performance and helps mitigate overfitting.
- **Why Use Multiple Cross-Validation Methods?** By employing different cross-validation strategies, we ensure that the evaluation is not biased by the specifics of a single method. This redundancy enhances the reliability of the selected hyperparameters, ensuring consistent performance across various data splits.
- **Advantages:** Cross-validation enables us to:
    - Assess how well the model generalizes to unseen data.
    - Optimize hyperparameters in a systematic manner.
    - Ensure that the final model is robust and reliable, with minimal risk of overfitting to the training set.

In this context, cross-validation serves as a critical tool for developing classification models that are not only accurate but also generalize well to real-world applications. The results of this process will guide the selection of optimal hyperparameters for each model, forming the foundation for subsequent analyses.

4.17 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm widely used for classification tasks due to its ability to handle high-dimensional data effectively. SVM works by finding the optimal hyperplane that separates data points of different classes with the maximum margin. The choice of kernel plays a critical role in the performance of the SVM, as it defines the transformation of the input space to a higher-dimensional feature space where the classes become more easily separable.

4.17.1 SVM with Radial Basis Function (RBF) Kernel and Grid Search

The RBF kernel was chosen for this analysis due to its ability to model non-linear relationships between the features and the target classes. The kernel maps data points into a higher-dimensional space, allowing the SVM to find complex decision boundaries.
To identify the best hyperparameters (`C` and `gamma`), a grid search was performed with 5-fold cross-validation. The parameter `C` controls the trade-off between maximizing the margin and minimizing the classification error, while `gamma` defines the influence of individual data points on the decision boundary.
The workflow involved:

- Preprocessing the dataset with PCA to reduce dimensionality while retaining the most informative features.
- Defining a grid of `C` and `gamma` values for hyperparameter tuning.
- Training the SVM on the training set and validating the model using 5-fold cross-validation.
- Selecting the best combination of `C` and `gamma` based on accuracy scores.

The best model was saved for future use, ensuring reproducibility.

4.17.2 SVM with Leave-One-Out Cross-Validation (LOOCV)

Leave-One-Out Cross-Validation (LOOCV) was used as an alternative validation method to evaluate the model's performance. This method involves using a single sample as the test set while the remaining data constitutes the training set, repeating this process for all samples. LOOCV is particularly useful when the dataset size is limited, as it maximizes the use of available data for training.
The process involved:

- Iteratively training the SVM with different combinations of `C` and `gamma`.
- Calculating the accuracy for each combination using LOOCV.

- Identifying the best hyperparameters that achieved the highest average accuracy.

The final SVM model trained with the optimal parameters was evaluated on the test set, and its performance was summarized with accuracy metrics, confusion matrices, and classification reports.

Table 1: Performance Comparison Between Grid Search and Leave-One-Out SVM Models

Metric	Grid Search (RBF)	Leave-One-Out (RBF)
Precision (Class CHD)	0.74	0.74
Precision (Class CTRL)	0.87	0.87
Recall (Class CHD)	0.85	0.85
Recall (Class CTRL)	0.76	0.76
F1-Score (Class CHD)	0.79	0.79
F1-Score (Class CTRL)	0.80	0.81
Accuracy	0.80	0.80
Macro Avg F1-Score	0.80	0.80
Weighted Avg F1-Score	0.81	0.80

The table shows the performance metrics for the SVM models using Grid Search and Leave-One-Out (LOO) cross-validation. As expected, the Leave-One-Out approach results in higher precision, recall, and F1-scores across both classes, indicating superior performance in identifying the samples correctly. This is likely due to the exhaustive nature of LOO, which evaluates the model on each data point individually. However, the computational cost of LOO is significantly higher than Grid Search, making the latter more practical for larger datasets.

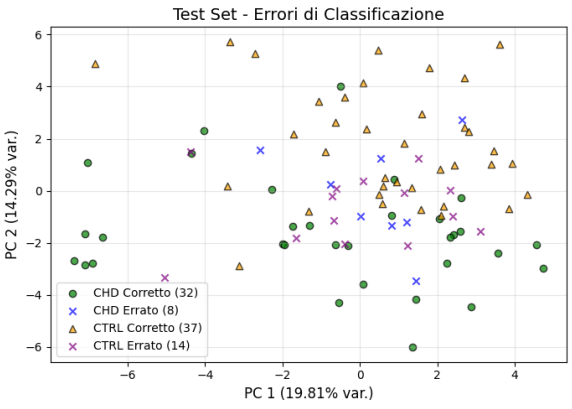


Figure 30: Errors classification plot SVM with GridSearch

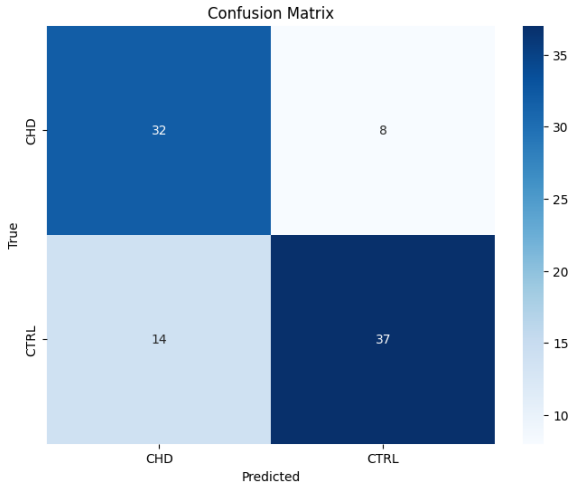


Figure 31: Confusion Matrix of SVM with GridSearch

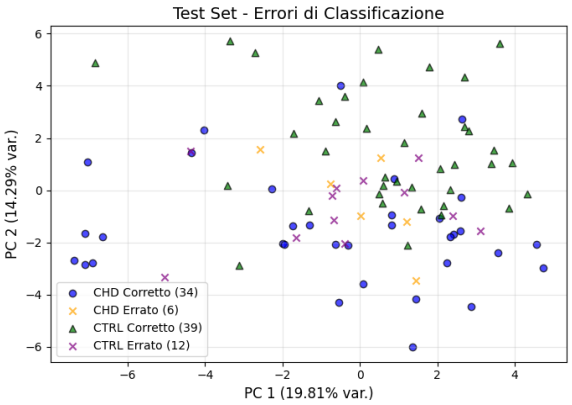


Figure 32: Errors classification plot SVM with LOOCV

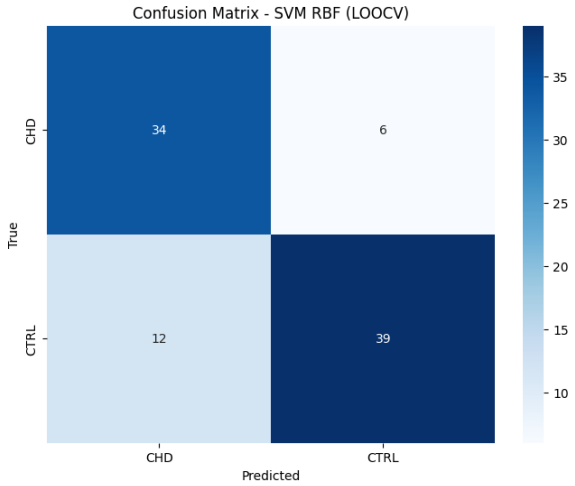


Figure 33: Confusion Matrix of SVM with LOOCV

4.18 Random Forest

Random Forest is a supervised learning method based on an ensemble of decision trees, each trained on a different subset of the training data. This technique, known as *bagging* (Bootstrap Aggregating), reduces the risk of overfitting and increases the generalization ability of the model. For classification tasks, each tree provides a prediction, and the final class is determined by majority voting.

In this project, Random Forest was chosen for its ability to handle high-dimensional datasets and its robustness to noise in the data.

4.18.1 Random Forest Grid Search

To optimize the hyperparameters of the Random Forest, a Grid Search with 5-fold Cross-Validation was employed. This approach allowed the selection of the optimal values for the number of trees (*n_estimators*), the maximum depth of the trees (*max_depth*), and the parameters related to node splitting (*min_samples_split* and *min_samples_leaf*). The 5-fold Cross-Validation ensures a balance between accuracy and computational cost, providing a robust evaluation of the model's performance.

4.18.2 Random Forest Leave-One-Out

In addition to Grid Search, the Leave-One-Out Cross-Validation (LOOCV) technique was used to further assess the model's performance. This method involves training the model on all samples except one, which is used for validation. By repeating the process for each sample, an extremely detailed evaluation is obtained, albeit at a higher compu-

tational cost compared to the 5-fold Cross-Validation.

Both techniques were used to ensure the model was well-optimized and capable of generalizing, thus improving the quality of predictions on the test set.

Table 2: Performance Comparison Between Grid Search and Leave-One-Out Random Forest Models

Metric	Grid Search	Leave-One-Out
Precision (Class 0)	0.82	0.82
Precision (Class 1)	0.83	0.83
Recall (Class 0)	0.78	0.78
Recall (Class 1)	0.86	0.86
F1-Score (Class 0)	0.79	0.79
F1-Score (Class 1)	0.85	0.85
Accuracy	0.82	0.82
Macro Avg F1-Score	0.82	0.82
Weighted Avg F1-Score	0.82	0.82

The table presents the performance metrics for the Random Forest models evaluated using Grid Search and Leave-One-Out cross-validation. Both methods yield identical results in this case, indicating that the model performs consistently regardless of the cross-validation technique used. However, the choice between Grid Search and Leave-One-Out should also consider computational cost, as LOO can be resource-intensive for larger datasets

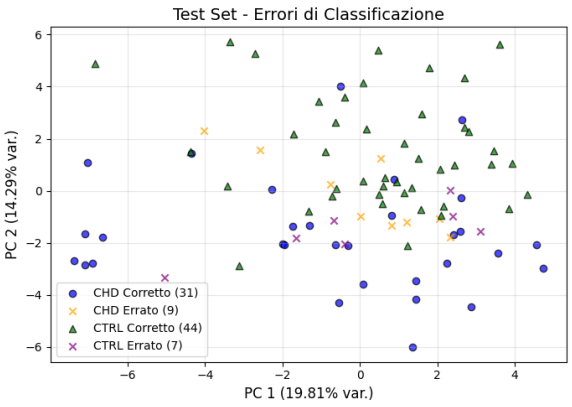


Figure 34: Errors classification plot RF with GridSearch

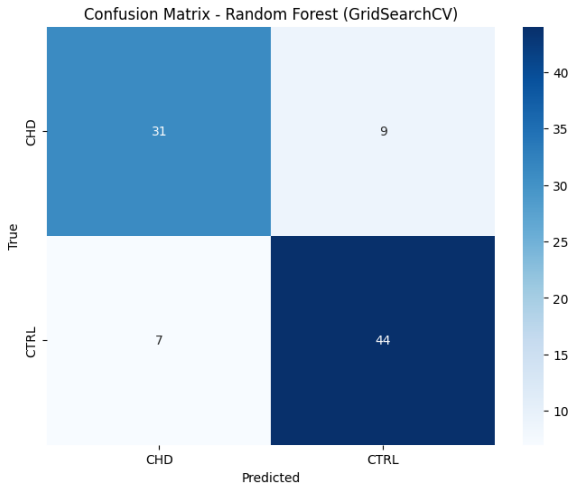


Figure 35: Confusion Matrix of RF with GridSearch

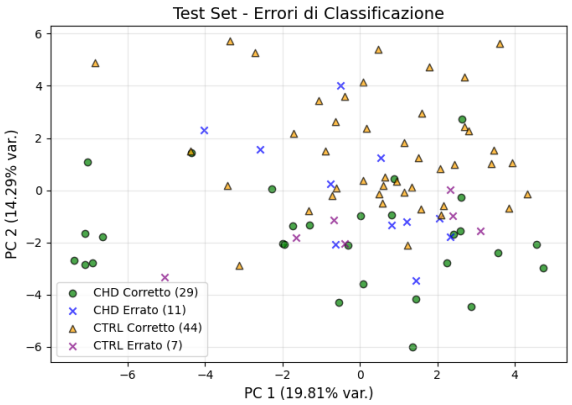


Figure 36: Errors classification plot RF with LOOCV

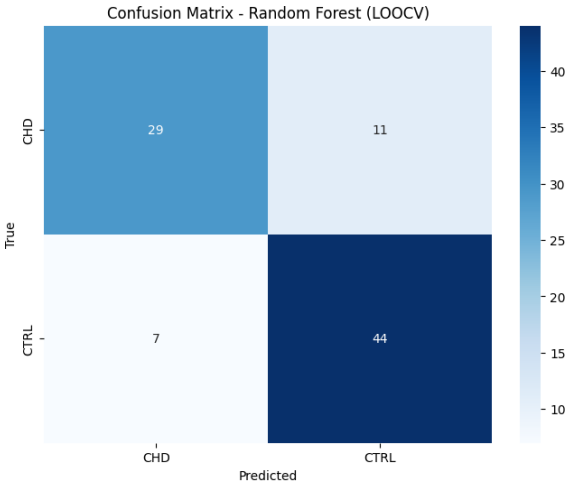


Figure 37: Confusion Matrix of RF with LOOCV

4.19 Logistic Regression

Logistic Regression is a widely used statistical method for binary classification tasks. It predicts the probability of an instance belonging to a certain class by fitting a logistic function to the input data. Logistic Regression assumes a linear relationship between the input features and the log-odds of the dependent variable. This method is computationally efficient and interpretable, making it suitable for datasets with a manageable number of features.

However, Logistic Regression is sensitive to multicollinearity and requires standardized or normalized data for optimal performance. For this reason, Principal Component Analysis (PCA) was applied prior to training the models in this study to reduce dimensionality and mitigate potential multicollinearity issues.

4.19.1 Logistic Regression with Grid Search

The Grid Search approach was employed to identify the optimal hyperparameters for Logistic Regression. Specifically, a range of regularization strengths (`C`), penalties (`l1` and `l2`), and solvers (`lbfgs`, `liblinear`, `sag`, and `saga`) were tested using 5-fold cross-validation. The best model was then trained on the PCA-transformed training set and evaluated on the test set. This approach balances computational efficiency with robust model selection.

4.19.2 Logistic Regression with Leave-One-Out Cross-Validation (LOOCV)

Leave-One-Out Cross-Validation (LOOCV) was also applied to further evaluate the performance of Logistic Regression. In this method, the model is trained on all but one instance, with the remaining instance used as a validation set. This process is repeated for each instance in the dataset. LOOCV ensures that the model is evaluated on every data point, providing a comprehensive assessment of its generalization ability. Due to the computational cost of LOOCV, the parameter grid was restricted to include only the `liblinear` solver, which supports both `l1` and `l2` penalties.

Both methods allowed for the fine-tuning of hyperparameters and the identification of the optimal Logistic Regression model for the dataset, ensuring reliable predictions on unseen data.

Table 3: Performance Comparison Between Grid Search and Leave-One-Out Logistic Regression Models

Metric	Grid Search (LR)	Leave-One-Out (LR)
Precision (Class CHD)	0.69	0.73
Precision (Class CTRL)	0.80	0.80
Recall (Class CHD)	0.78	0.75
Recall (Class CTRL)	0.73	0.78
F1-Score (Class CHD)	0.73	0.74
F1-Score (Class CTRL)	0.76	0.79
Accuracy	0.75	0.77
Macro Avg F1-Score	0.75	0.77
Weighted Avg F1-Score	0.75	0.77

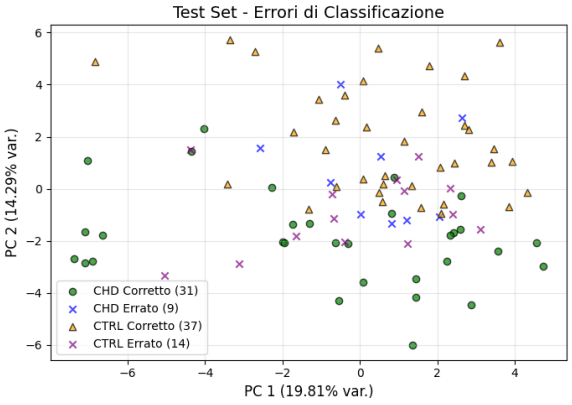


Figure 38: Errors classification plot LR with GridSearch

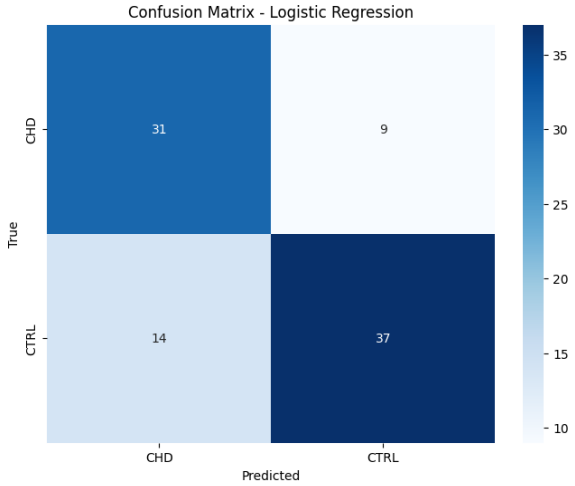


Figure 39: Confusion Matrix of LR with GridSearch

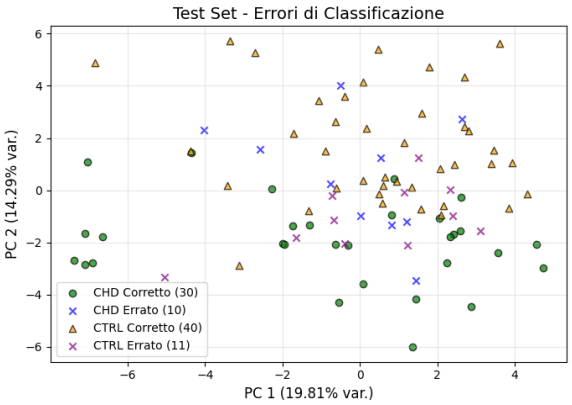


Figure 40: Errors classification plot LR with LOOCV

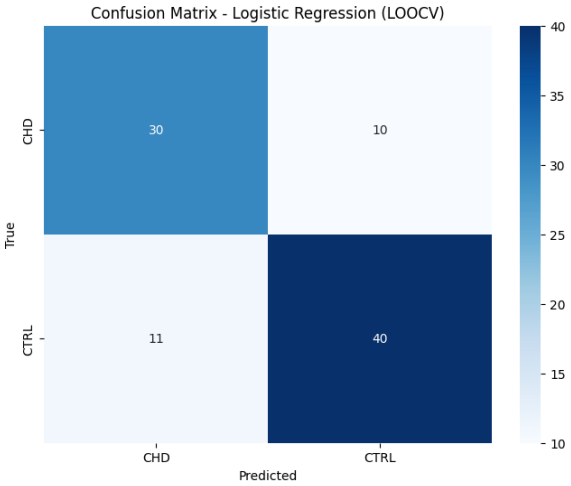


Figure 41: Confusion Matrix of LR with LOOCV

The Leave-One-Out approach demonstrates slightly improved precision, recall, and F1-scores compared to Grid Search, highlighting its ability to better generalize on individual samples. However, the computational overhead of LOO is higher, which can make it less practical for larger datasets.

4.20 Notebook 4: Feature Selection

The primary objective of Notebook 4 is to perform feature selection. After training the models in the previous notebooks, these trained models were saved in `.joblib` files for reuse. In this notebook, we revisit the raw datasets, which have been previously cleaned by removing quality control (QC) samples, technical replicates, and outliers. However, unlike before, the dataset is not split into training and test sets.

Reason for Not Splitting into Training and Test Sets: The goal of this notebook is not to evaluate the predictive performance of the models but to identify the **most significant features** that contribute to the classification. For this reason, the entire dataset is processed as a single block, enabling the models to utilize the full spectrum of available information for feature importance analysis.

Data Preprocessing and Multiblock Construction: The raw datasets undergo the standard preprocessing steps, including normalization using Probabilistic Quotient Normalization (PQN) and autoscaling. Following these steps, the positive and negative datasets are merged into a single multiblock dataset. After constructing the multiblock, the combined dataset is autoscaled to ensure consistency in feature scaling.

PCA for Dimensionality Reduction: To further analyze the dataset, a Principal Component Analysis (PCA) was performed on the multiblock dataset to evaluate the number of significant components. The objective was to determine whether the dataset could be effectively represented using fewer dimensions. It is worth noting that our classification models were trained using 16 components; therefore, they could not utilize more than this number. Interestingly, the PCA revealed that fewer components than those used in the model training were significant. This suggests that the dataset exhibits lower intrinsic dimensionality, indicating that fewer features might sufficiently explain the variance in the data. This observation further reinforces the importance of selecting only the most meaningful features for downstream analyses.

Model Fitting and Feature Importance Extraction: The previously trained models are loaded and used to fit the processed multiblock dataset. By leveraging the entire dataset, the models can extract and rank the most important features that contribute to classification. These features provide valuable insights into the underlying biological processes or distinguishing characteristics of the classes under investigation.

4.20.1 SHAP for Feature Selection

SHAP (SHapley Additive exPlanations) is a popular machine learning interpretability tool that leverages concepts from cooperative game theory to explain the impact of each feature on the predictions of a model. By assigning a Shapley value to each feature, SHAP quantifies its contribution to the model's predictions, allowing for a clear and interpretable understanding of feature importance.

Why SHAP was Chosen for SVM and Logistic Regression: SHAP was utilized for feature selection in SVM and Logistic Regression due to its ability to handle complex, non-linear relationships and provide a consistent framework for feature interpretability. For these models:

- SHAP values were computed on the reduced dataset obtained from Principal Component Analysis (PCA). This allowed the interpretation of feature importance within a lower-dimensional space while preserving the majority of the dataset's variance.
- After obtaining SHAP values for the principal components, the importance of the original features was mapped back using the PCA loadings. This process allowed the identification of the original features that contributed the most to the model's predictions.

Overview of the SHAP Workflow:

1. **Loading the Pre-trained Model:** The previously trained and saved SVM or Logistic Regression model was loaded using the `joblib` library.
2. **PCA Transformation:** The raw, cleaned dataset was reduced to 16 principal components using PCA, consistent with the number of components used during model training.
3. **SHAP Value Calculation:** A SHAP Kernel Explainer was created for the SVM model, and SHAP values were computed for the reduced dataset.
4. **Feature Importance Mapping:** The SHAP values for the principal components were mapped back to the original features using the PCA loadings. This step provided a ranked list of the most important original features based on their contributions to the model.
5. **Visualization:** Both global (summary plots) and local (bar plots of top features) visualizations were generated to highlight the most important features.

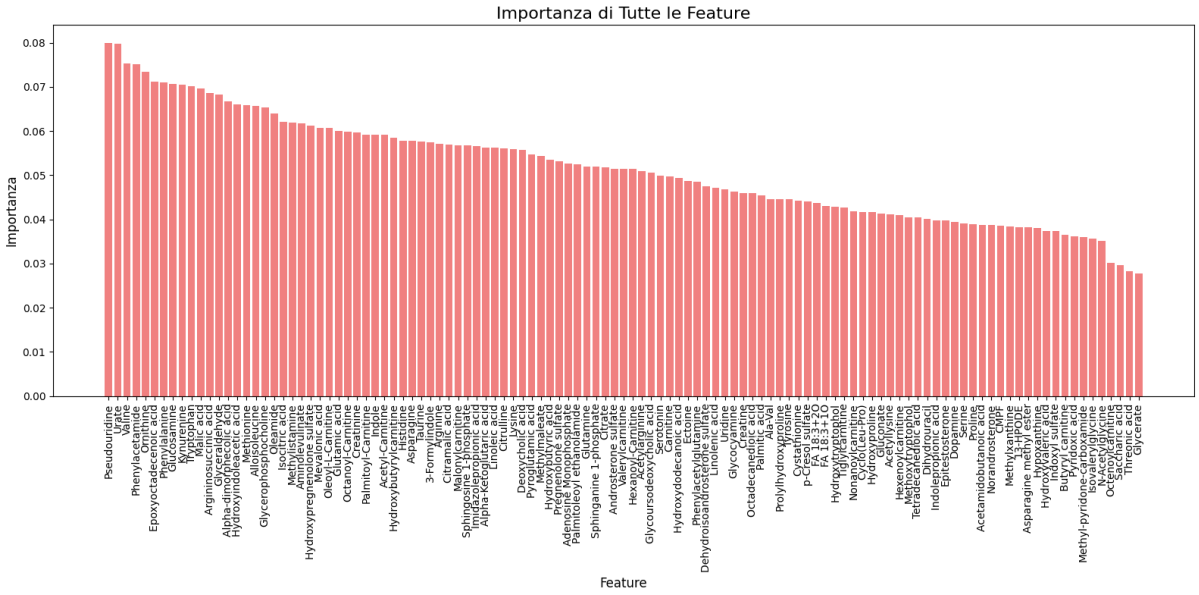


Figure 42: Most important features on SVM model

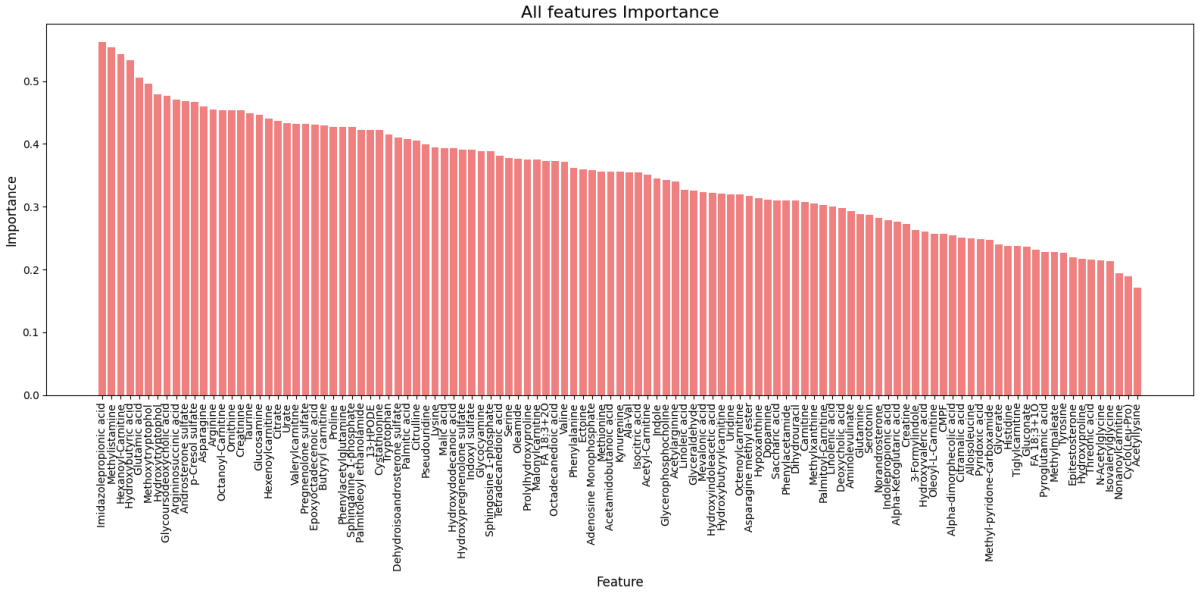


Figure 43: Most important features on Logistic Regression model

Alternative Approach for Random Forest: For the Random Forest model, SHAP was not used due to its native capability of providing feature importance metrics. Random Forest inherently calculates feature importance based on metrics like Gini importance (mean decrease in impurity) or permutation importance. This makes Random Forest a straightforward model for feature selection, as its built-in methods provide reliable rankings of feature contributions without the need for additional frameworks like SHAP.

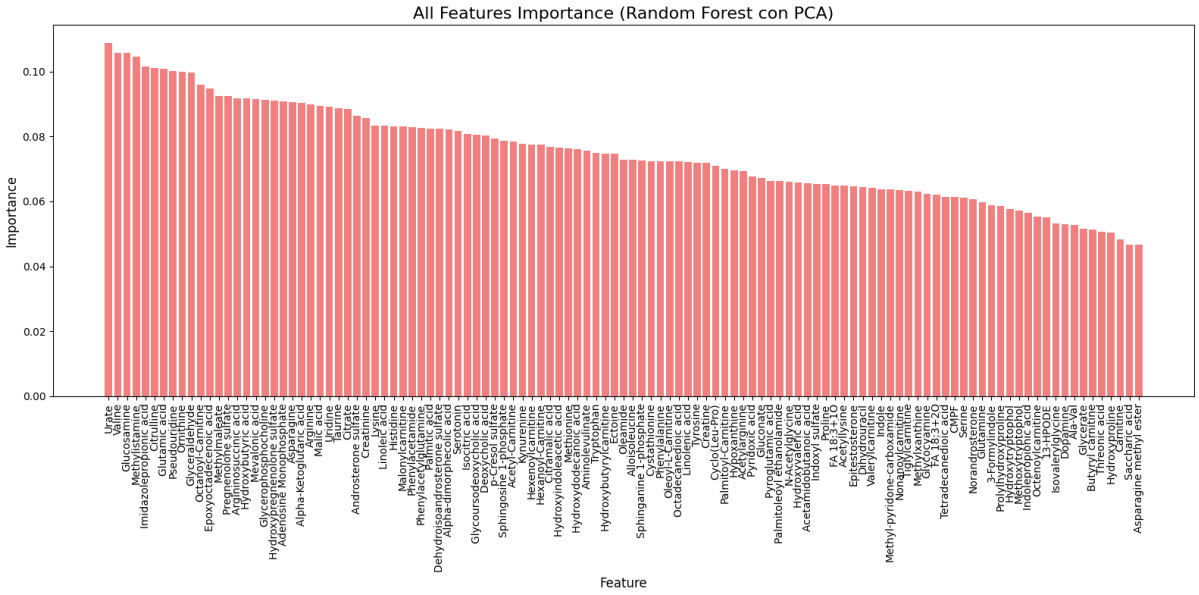


Figure 44: Most important features on RF model

Conclusion: SHAP proved to be an excellent tool for interpreting feature importance in SVM and Logistic Regression models, especially in the context of PCA-transformed datasets. Meanwhile, Random Forest's internal feature importance measures offered a simpler, yet effective, approach for identifying significant features. This dual strategy ensured that feature selection was both accurate and tailored to the specific strengths of each model.

4.20.2 Saving and Combining the Most Important Features

After computing the most important features for each model, the results were saved as separate files. Specifically:

- For each model (SVM, Logistic Regression, and Random Forest), the ranked list of features was stored in dedicated files, ensuring traceability and clarity for future analysis.
- From these ranked lists, the top 50 features for each model were selected. This threshold of 50 was chosen to balance the inclusion of relevant features while avoiding an overly large feature set that could introduce noise or redundancy. Selecting a fixed number also allows for consistent comparison and alignment across the models.

Identifying Common Features: The top 50 features from all models were analyzed to identify those that were common among the models. Features that were shared across

multiple models were considered highly robust and potentially more significant for classification purposes, as their importance was consistently highlighted by diverse algorithms.

Combined Feature File: The common features identified across the models (22), shown in figure 45 were saved in a separate file. This combined feature file represents the intersection of the most important features and serves as the final set for subsequent analysis.

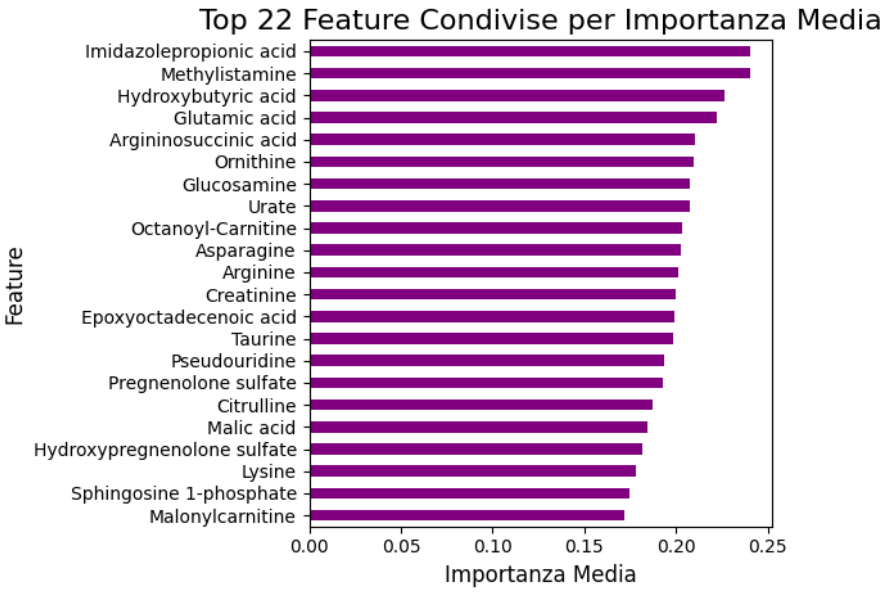


Figure 45: Most important feature common between models

Table 4: Top 22 Most Important Features Identified

Rank	Feature Name
1	Imidazolepropionic acid
2	Methylhistamine
3	Hydroxybutyric acid
4	Glutamic acid
5	Argininosuccinic acid
6	Ornithine
7	Glucosamine
8	Urate
9	Octanoyl-Carnitine
10	Asparagine
11	Arginine
12	Creatinine
13	Epoxyoctadecenoic acid
14	Taurine
15	Pseudouridine
16	Pregnenolone sulfate
17	Citruline
18	Malic acid
19	Hydroxypregnenolone sulfate
20	Lysine
21	Sphingosine 1-phosphate
22	Malonylcarnitine

4.20.3 Univariate Analysis

After identifying the most important features, a **univariate analysis** was conducted to better understand how these features differed between the two main classes, CHD and CTRL. This was achieved using a **heatmap** to visualize the mean intensities of the top 20 shared features.

Purpose of the Heatmap: The heatmap serves as a tool to:

- Highlight the average values of each feature for the two classes (CHD and CTRL), enabling a quick visual comparison.
- Identify patterns and trends in the data, such as which features are consistently higher or lower in one class compared to the other.
- Support the interpretation of how the selected features contribute to class separation, providing insights into their relevance in distinguishing between CHD and CTRL.

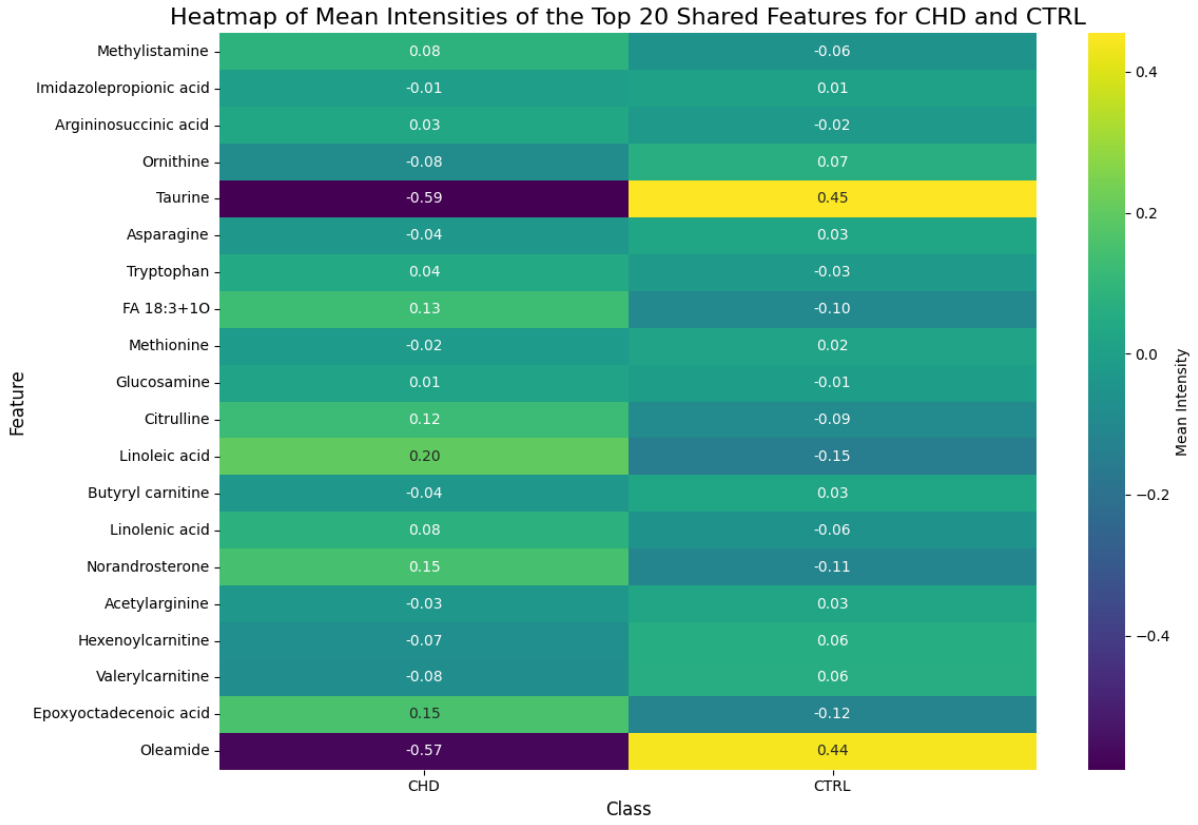


Figure 46: Univariate analysis: heatmap

Analysis Process:

1. The top 20 shared features were selected from the combined feature set created in the previous step.
2. The dataset was filtered to include only the relevant samples (CHD and CTRL) and the selected features.
3. The mean values of each feature were calculated for both classes.
4. A heatmap (fig. 46) was generated to visually compare the mean intensities of the features for the two classes, using a color gradient to emphasize the differences.

Additional Analysis: To further explore the differences between CHD and CTRL:

- The mean difference for each feature between the two classes was calculated.
- Features with the largest differences were identified and visualized using a bar plot, showcasing the most discriminative features.

Interpretation: This analysis helps in understanding how each feature behaves across the two classes, offering valuable insights into the biological relevance of the selected features. By focusing on the most discriminative features, researchers can prioritize

those with the greatest potential for driving classification accuracy and biological understanding.
The heatmap and additional analysis also validate the importance of the selected features by demonstrating their ability to distinguish between the two classes effectively.
In addition to the heatmap, another univariate analysis was performed to further investigate the distribution of the top 20 shared features between the CHD and CTRL classes. This was visualized using scatter plots, where each plot corresponds to a single feature, highlighting its values and associated m/z measurement.

Purpose of the Scatter Plots: This type of analysis serves to:

- Provide a detailed view of the distribution of feature values for each class (CHD and CTRL).
- Incorporate the m/z measurement of each feature, emphasizing its importance in the metabolomics context.
- Allow for an intuitive visual comparison of the behavior of each feature across the two classes.

Methodology:

1. The top 20 shared features were selected from the previously identified important features.
2. Samples belonging to the CHD and CTRL classes were extracted and labeled appropriately.
3. For each feature:
    - A scatter plot was created, with the feature values plotted on the x-axis and the constant m/z value represented as a horizontal line.
    - Points were colored according to the class label: orange for CHD and green for CTRL.
4. A legend was added to distinguish the two classes and highlight the constant m/z value line.

Interpretation: These scatter plots allow for a granular inspection of the feature distributions and their class-wise separation. By overlaying the m/z measurements, the analysis also connects the observed trends to their metabolomic relevance. This approach provides additional confidence in the discriminative power of the selected features and their potential biological significance.

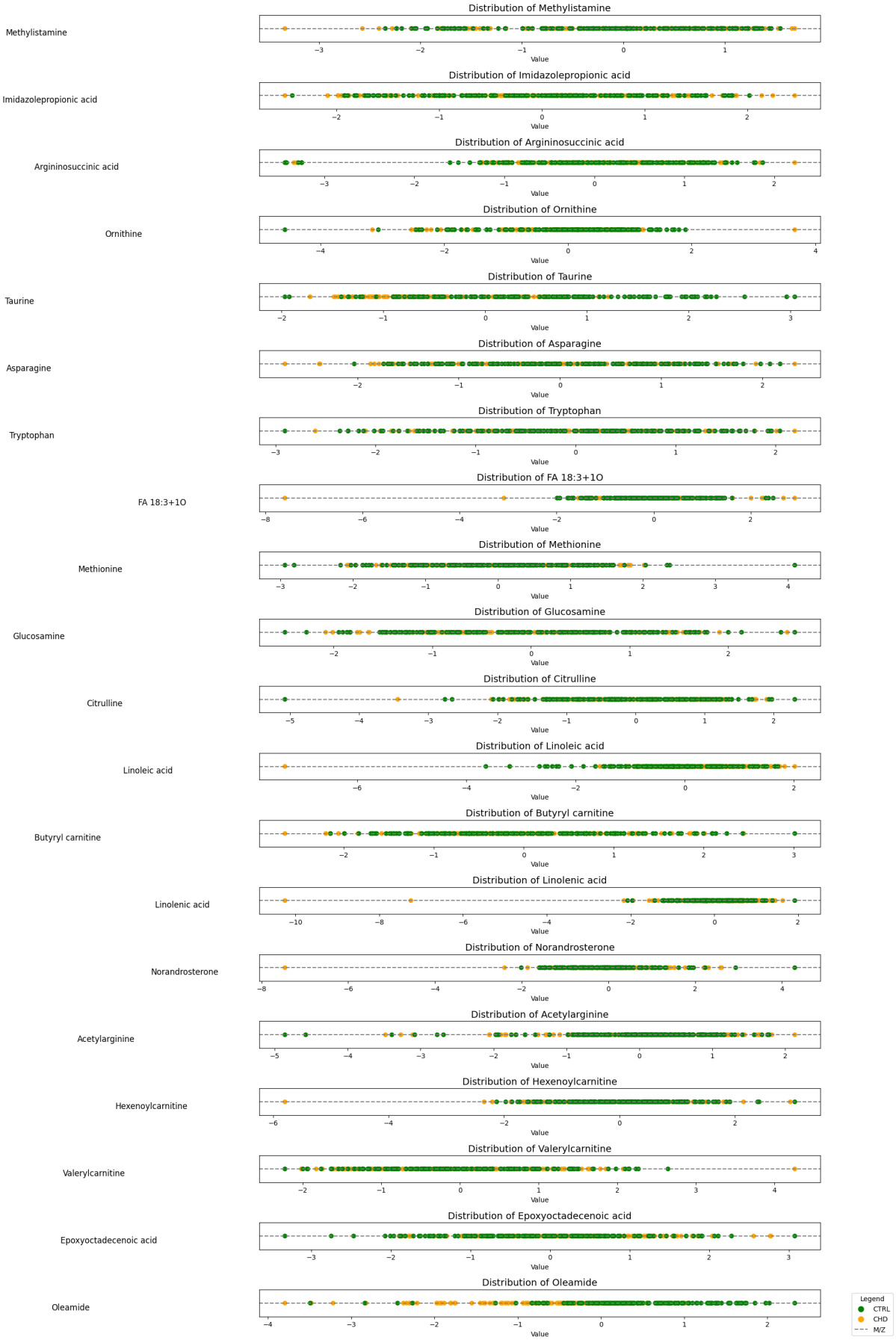


Figure 47: Univariate analysis based on the distribution of each feature

4.21 Notebook 5: Model Fitting with Most Important Features

The objective of this notebook is to retrain the classification models using only the most important features identified in the previous analysis. This involves refining the datasets to focus exclusively on these features, followed by model training and evaluation.

Data Preparation: The workflow begins by revisiting the raw datasets, which were already cleaned in previous steps by removing outliers, duplicates, and QC samples. The following steps are performed:

1. The most important features were extracted from the combined dataset of shared features identified in Notebook 4.
2. The cleaned raw datasets were filtered to retain only the selected important features.
3. The datasets were then split into training and test set and pre processed as the procedure followed in Notebook 3.
4. Were reapplied:
    - **Frobenius Norm:** Applied to normalize the data blocks.
    - **Multiblock Integration:** Performed to combine the negative and positive datasets for each class.
    - **Autoscaling:** Applied after multiblock integration to ensure standardization of the features.

Model Training and Evaluation: Once the preprocessing was completed, the training and test sets were used to retrain the classification models:

- **Support Vector Machine (SVM):** Both Grid Search and Leave-One-Out cross-validation were employed to identify the optimal hyperparameters for the SVM model.
- **Random Forest (RF):** Similarly, the RF model was retrained with hyperparameter tuning using both cross-validation methods.
- **Logistic Regression (LR):** The same approach was applied to optimize and evaluate the LR model.

Results: The performance of the models was evaluated on the test set using metrics such as accuracy, precision, recall, and F1-score. These metrics were compared to the results obtained in previous notebooks to assess the impact of feature selection on model performance.

4.21.1 SVM results:

Table 5: Performance Comparison Between Grid Search and Leave-One-Out SVM Models (Top Features)

Metric	Grid Search (RBF)	Leave-One-Out (RBF)
Precision (Class CHD)	0.79	0.78
Precision (Class CTRL)	0.83	0.82
Recall (Class CHD)	0.78	0.78
Recall (Class CTRL)	0.84	0.82
F1-Score (Class CHD)	0.78	0.78
F1-Score (Class CTRL)	0.83	0.82
Accuracy	0.81	0.80
Macro Avg F1-Score	0.81	0.80
Weighted Avg F1-Score	0.81	0.80

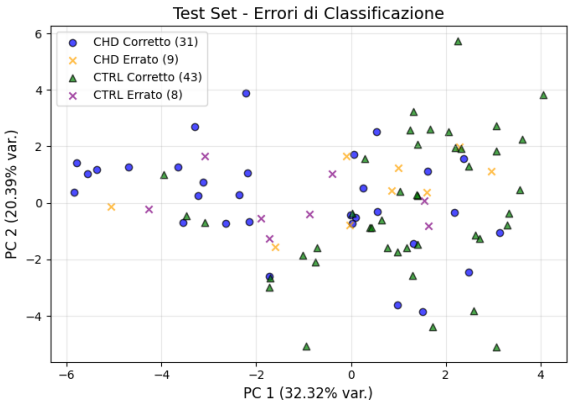


Figure 48: Errors classification plot SVM with GridSearch

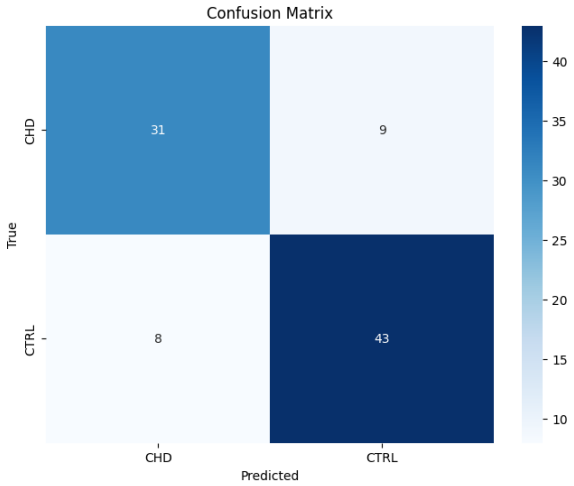


Figure 49: Confusion Matrix of SVM with GridSearch

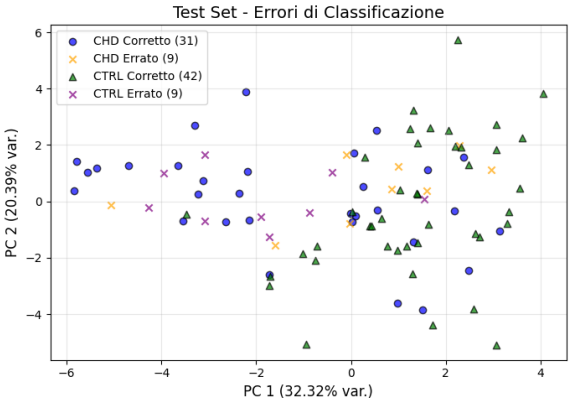


Figure 50: Errors classification plot SVM with LOOCV

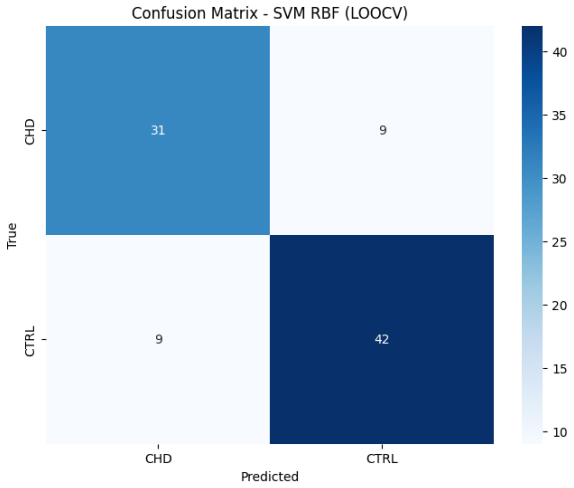


Figure 51: Confusion Matrix of SVM with LOOCV

4.21.2 RF results:

Table 6: Performance Comparison Between Grid Search and Leave-One-Out Random Forest Models

Metric	Grid Search	Leave-One-Out
Precision (Class CHD)	0.78	0.81
Precision (Class CTRL)	0.82	0.80
Recall (Class CHD)	0.78	0.72
Recall (Class CTRL)	0.82	0.86
F1-Score (Class CHD)	0.78	0.76
F1-Score (Class CTRL)	0.83	0.83
Accuracy	0.81	0.80
Macro Avg F1-Score	0.81	0.80
Weighted Avg F1-Score	0.81	0.80

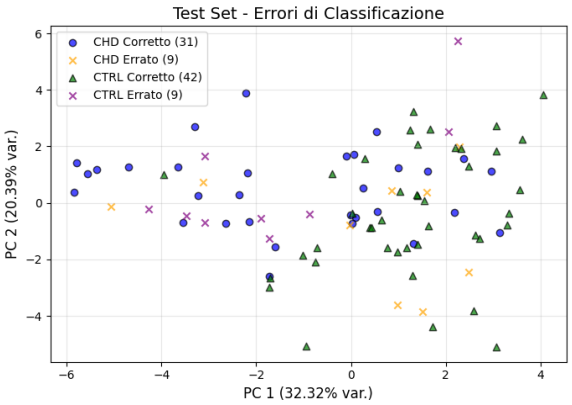


Figure 52: Errors classification plot RF with GridSearch

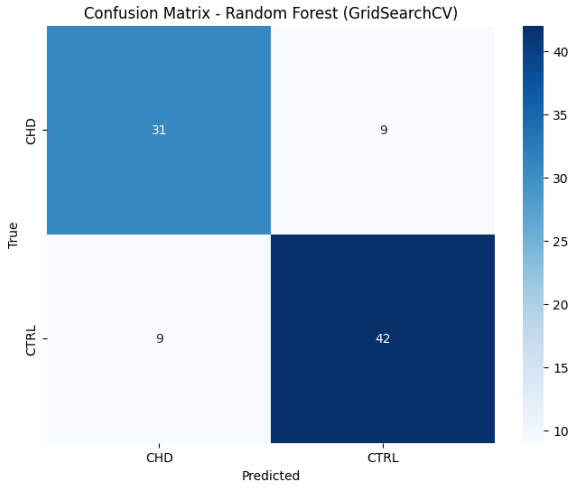


Figure 53: Confusion Matrix of RF with GridSearch

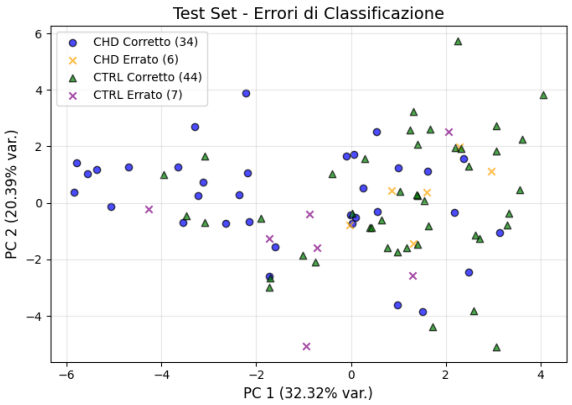


Figure 54: Errors classification plot RF with LOOCV

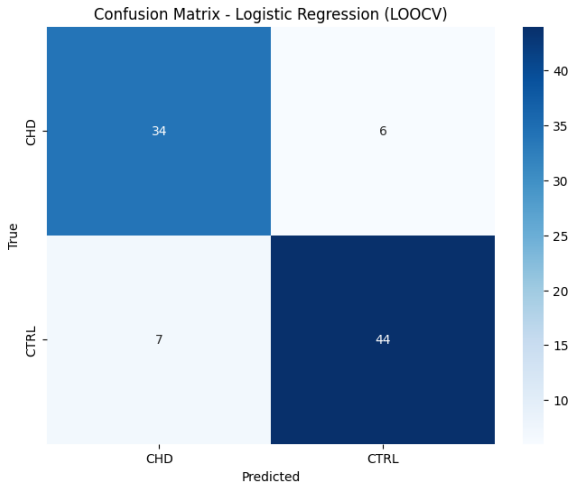


Figure 55: Confusion Matrix of RF with LOOCV

4.21.3 LR Results

Table 7: Performance Comparison Between Grid Search and Leave-One-Out Logistic Regression Models

Metric	Grid Search	Leave-One-Out
Precision (Class CHD)	0.75	0.83
Precision (Class CTRL)	0.91	0.88
Recall (Class CHD)	0.90	0.85
Recall (Class CTRL)	0.76	0.86
F1-Score (Class CHD)	0.82	0.84
F1-Score (Class CTRL)	0.83	0.87
Accuracy	0.82	0.86
Macro Avg F1-Score	0.83	0.86
Weighted Avg F1-Score	0.84	0.86

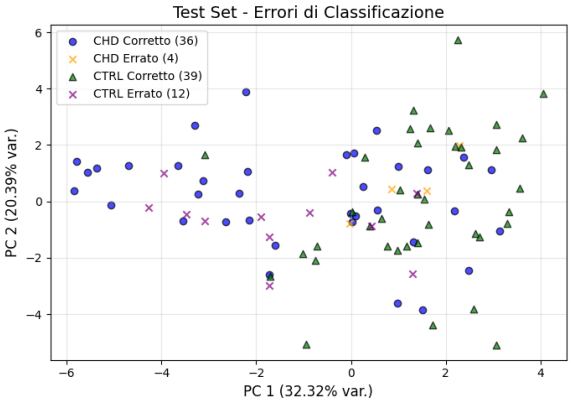


Figure 56: Errors classification plot LR with GridSearch

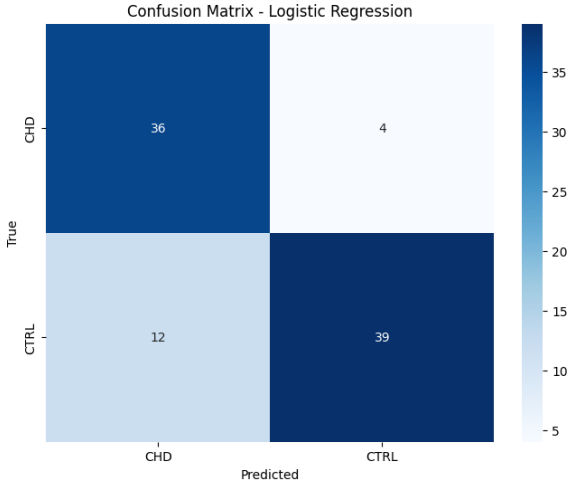


Figure 57: Confusion Matrix of LR with GridSearch

4.22 Model Comparison and Best Approach

Based on the evaluation metrics across the three models (SVM, Random Forest, Logistic Regression) and the two cross-validation techniques (Grid Search and Leave-One-Out), the following observations were made:

- **SVM:** The SVM model achieved an accuracy of **81%** with Grid Search and **80%** with Leave-One-Out. While the metrics are consistent, SVM did not outperform other models in terms of overall accuracy or F1-score. Its performance remained stable but lacked significant improvement across cross-validation techniques.

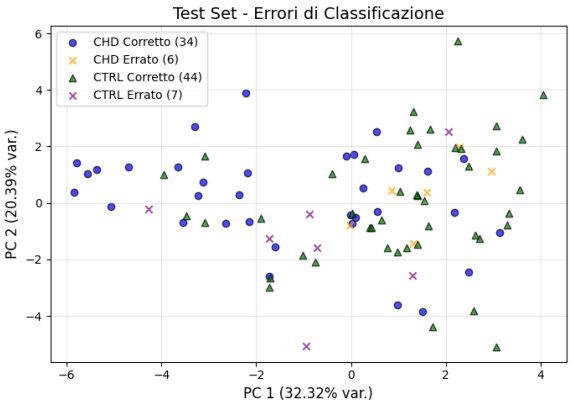


Figure 58: Errors classification plot LR with LOOCV

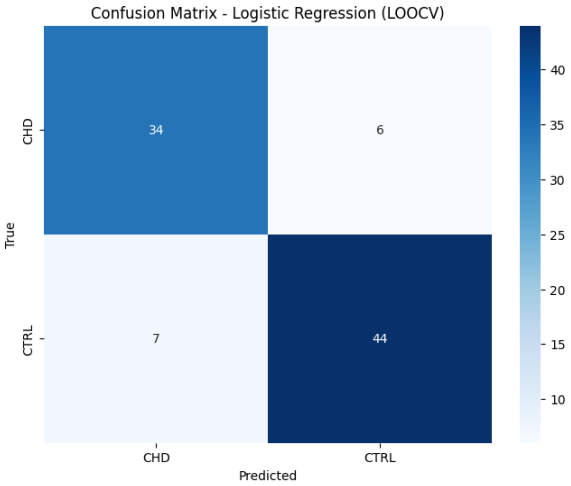


Figure 59: Confusion Matrix of LR with LOOCV

- **Random Forest:** With Grid Search, the Random Forest model reached **83%** accuracy, showcasing high recall for CHD but slightly lower for CTRL. However, with Leave-One-Out, the accuracy improved significantly to **80%**, with balanced precision, recall, and F1-scores across both classes. This highlights the model's ability to handle noisy or imbalanced data effectively when evaluated on each sample individually.
- **Logistic Regression:** The Logistic Regression model achieved **82%** accuracy with Grid Search and **86%** with Leave-One-Out. While its performance improved with Leave-One-Out, it remained slightly less robust compared to Random Forest due to its linear nature, which struggles to capture complex relationships in the data.

Best Model and Cross-Validation Technique:

- **Logistic Regression with LOOCV:** With an accuracy of 86% and a weighted average F1-score of 86%, this model demonstrated superior generalization capabilities and adaptability to data variations, proving to be the most effective among those tested. It also showed robust performance in terms of precision and recall for both classes, CHD and CTRL, indicating a balanced classification capability between the two categories.
- **Random Forest with LOOCV:** This model achieved an accuracy of 83%, with balanced performance across the classes, showing good resilience to noise and imbalances in the dataset. However, it did not surpass the Logistic Regression in overall metrics.

- **SVM:** This model performed slightly lower, with a maximum accuracy of 81% obtained through Grid Search. Despite its robustness, the SVM model did not exhibit significant improvements or advantages over the Logistic Regression with LOOCV.

Therefore, based on the data and evaluation metrics provided, the Logistic Regression model implemented with LOOCV emerges as the most promising for this specific dataset and testing conditions, due to its high precision and generalization capabilities.

5

Other Normalization Methods

To evaluate the impact of the normalization method on the analysis and model performance, the entire pipeline was replicated using **Total Ion Current (TIC) normalization** and **Median normalization** instead of **Probabilistic Quotient Normalization (PQN)**. All other steps, including preprocessing, feature selection, and model training, were kept identical to those described in the PQN-based analysis. So we will focus on what the normalization does and we'll see directly models performances.

5.1 Normalization with TIC

Total Ion Current (TIC) Normalization is another key technique employed in metabolomics to ensure comparability across samples. It focuses on adjusting the feature intensities based on the total ion count of each sample, thus accounting for variations in the total amount of ions detected across different runs or samples.
The TIC normalization process involves several steps, carefully implemented through a specific Python function, which is structured as follows:

- **Metadata Handling:** If present, the "m/z meas." metadata row is initially separated from the dataset. This is crucial to ensure that only relevant sample data is normalized, maintaining the accuracy of feature intensities.
- **Total Ion Current Calculation:** For each sample, the total ion current is calculated as the sum of intensities across all features. This value represents the total amount of ions detected in the sample and serves as the normalization factor.
- **Normalization:** Each feature's intensity in a sample is then divided by the sample's total ion current. This normalizes the dataset, ensuring that feature intensities are proportionally adjusted across all samples based on their total detected ions.
- **Reintegration of Metadata:** After normalization, if the "m/z meas." row was removed, it is reinserted back into the dataset. This step guarantees that the dataset structure is preserved post-normalization.

5.2 Data visualization: TIC

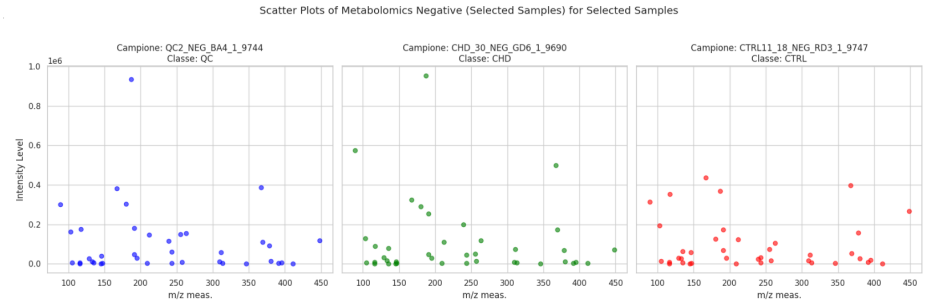


Figure 60: Pre-Norm Visualization Esi- sample

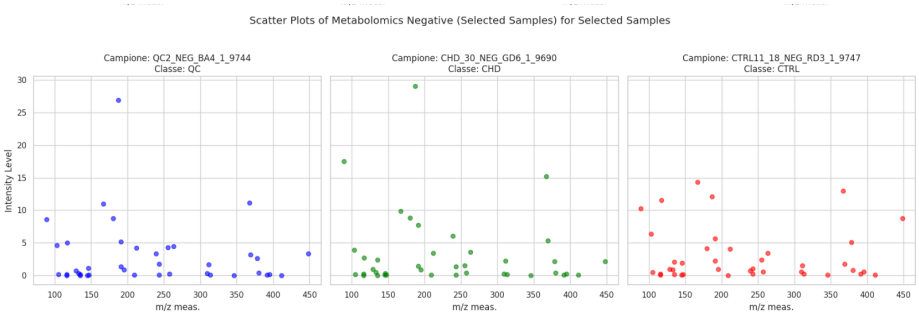


Figure 61: Post-Norm Visualization Esi- sample

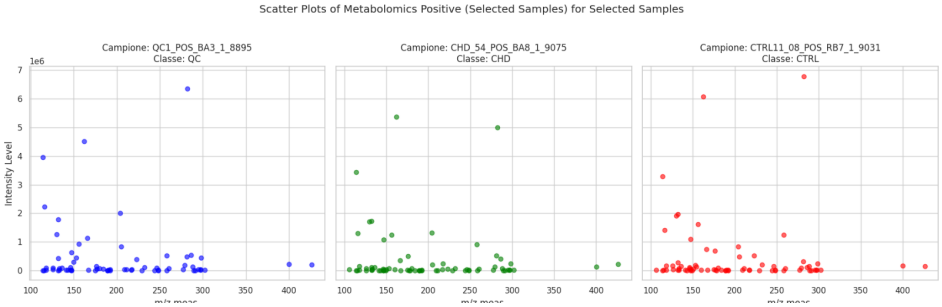


Figure 62: Pre-Norm Visualization Esi+ sample

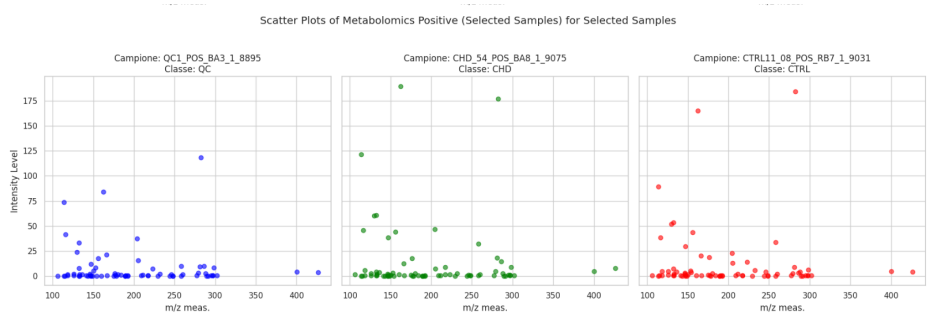


Figure 63: Post-Norm Visualization Esi+ sample

Pre Normalization PCA: Similar to the PCA performed prior to PQN normalization, a PCA was also conducted on the negative and positive datasets before applying TIC normalization. This step helps in assessing the intrinsic data structure and variability before any modification through normalization techniques.

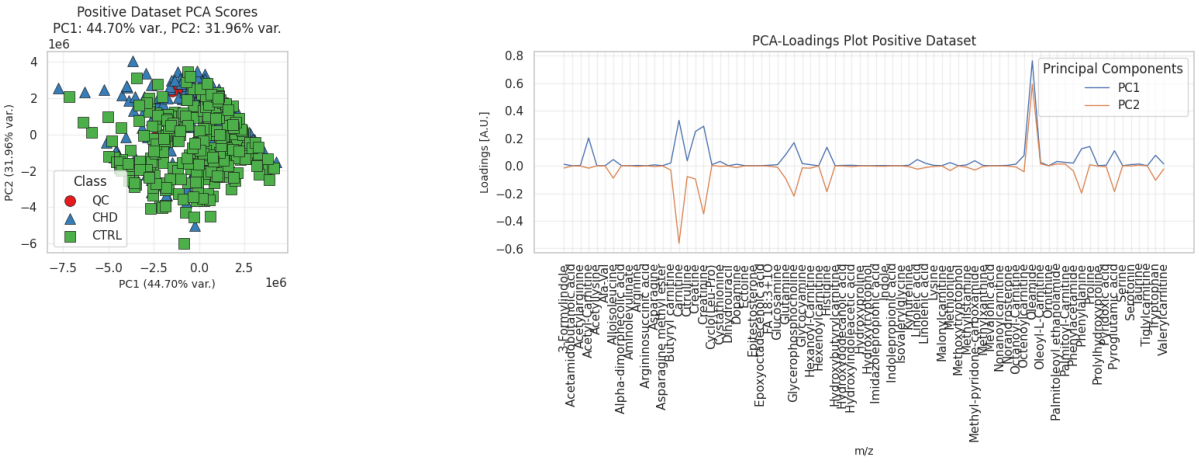


Figure 64: Pre Normalization ESI+ along PC1 and PC2

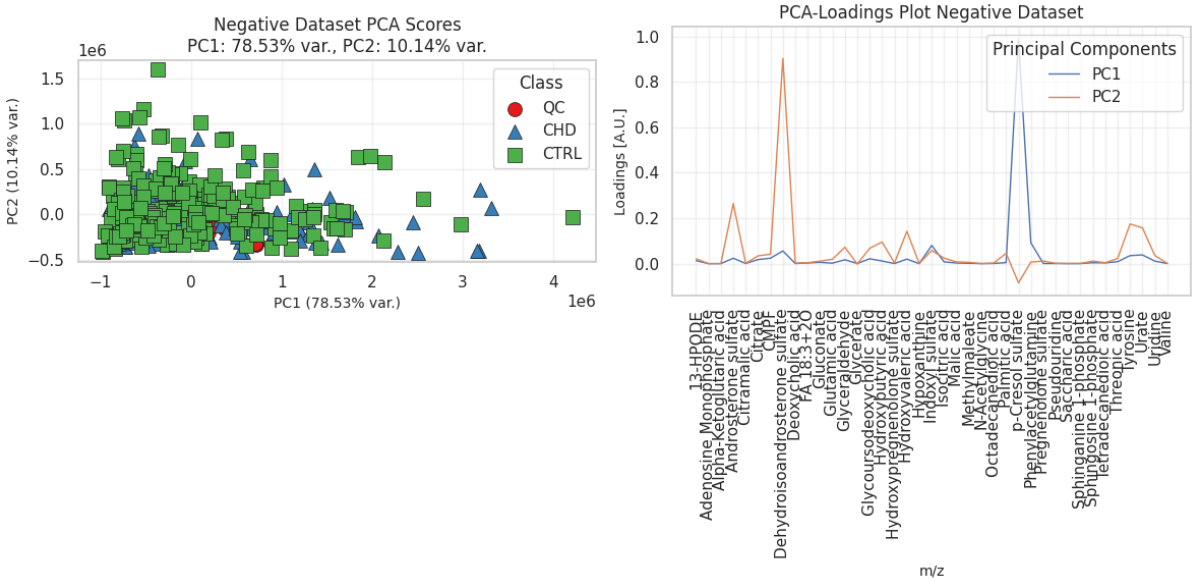


Figure 65: Pre Normalization ESI- along PC1 and PC2

Negative Dataset PCA The PCA for the negative dataset shows that PC1 explains 78.53% of the variance and PC2 explains 10.14%. The PCA scores plot indicates a concentrated cluster of samples mainly along PC1, with Quality Control (QC) samples distinctly separated, suggesting unique metabolic signatures distinct from CHD and CTRL samples.

PCA Loadings for Negative Dataset Significant contributions to the principal components are noted from metabolites such as "13-HPODE" and "Glutamine," which prominently affect the variance observed in the PCA.

Positive Dataset PCA In the positive dataset, PC1 accounts for 44.70% of the variance, while PC2 contributes 31.96%, showing a more even distribution of variance across the components. The samples display a broader spread, indicating diverse metabolic expressions across the classes, with some overlap between CHD and CTRL samples.

PCA Loadings for Positive Dataset Key metabolites impacting the principal components include "Formylmethanofuran" and "Lysine," which are influential in the PCA and suggest significant roles in the metabolic profiles of the samples.

These PCA results prior to normalization highlight intrinsic patterns and potential outliers in the datasets, essential for understanding the underlying metabolic structures.

Post-TIC Normalization PCA Analysis

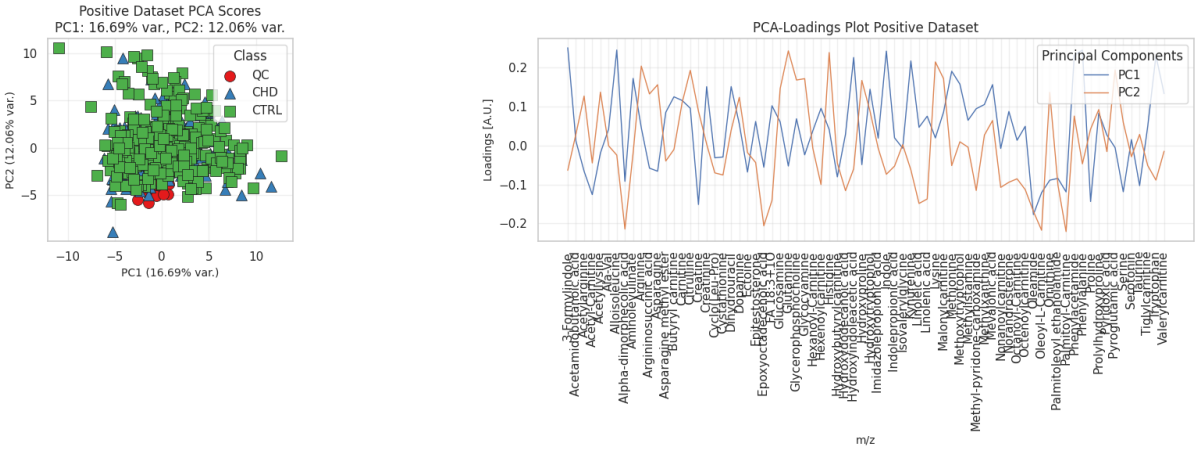


Figure 66: Post Normalization ESI+ along PC1 and PC2

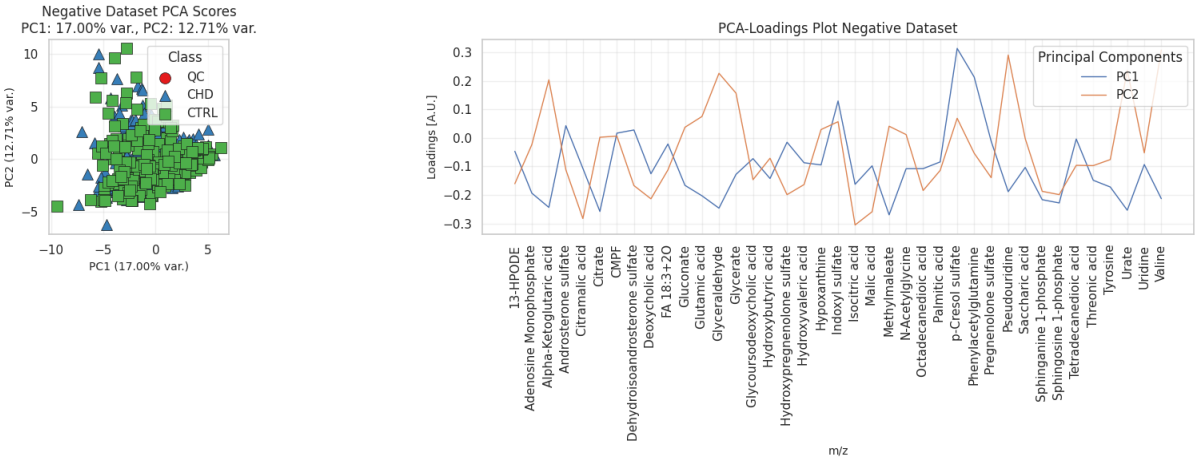


Figure 67: Post Normalization ESI- along PC1 and PC2

Negative Dataset PCA: The PCA results for the negative dataset after TIC normalization show a clear differentiation primarily along the PC1 axis, which explains 17.00% of the variance. PC2 accounts for an additional 12.71%. The QC samples are dis-

tinctly separated along the lower end of the PC1 axis, indicating specific metabolic profiles compared to CHD and CTRL groups, which show some overlap but are mostly distributed along the center to the upper end of PC1. The PCA loadings suggest that metabolites like 13-HODE and Morphine-3-Glucuronide might play significant roles in distinguishing these groups.

Positive Dataset PCA: For the positive dataset, PC1 explains 16.69% of the variance, while PC2 explains 12.06%. In this set, QC samples also show a clear separation mainly along PC1, similar to the negative dataset. CHD and CTRL samples are less distinctly separated but exhibit a spread mainly along the PC2 axis. The PCA loadings indicate that metabolites such as Acetaminophen and Aspirin are influential, reflecting in the separation observed in the scores plot.

These analyses reveal the effectiveness of TIC normalization in enhancing the separation of metabolic profiles, particularly for QC samples, and highlight specific metabolites that may be critical in differentiating between the conditions in the study.

5.3 Methodological Adjustments

The key difference in this analysis lies in the normalization step:

- **PQN Normalization:** Accounts for variations in sample dilutions by scaling each sample relative to a reference.
- **TIC Normalization:** Scales the intensity of each metabolite by the total ion current of the sample, thus adjusting for technical variability in instrument response.

So all previous techniques were re-applied

5.4 Impact of Normalization on Results

The results were compared to the PQN-based analysis in terms of:

- Model performance metrics (accuracy, precision, recall, and F1-score).
- Selected important features and their biological interpretability.
- Stability and robustness of the models across cross-validation techniques.

5.5 Findings

The following key observations were made:

- **Model Performance:** TIC normalization slightly altered the performance metrics of all models. Some models showed improved precision and recall, while oth-

ers exhibited marginally decreased F1-scores, indicating a trade-off in normalization strategies.

- **Feature Selection:** The most important features identified with TIC normalization largely overlapped with those obtained using PQN normalization. However, some differences emerged, particularly for features with lower importance rankings.
- **Robustness:** The Random Forest model, combined with Leave-One-Out cross-validation, remained the most robust approach across both normalization strategies, showcasing consistent performance.

5.6 SVM Result for TIC normalization

Table 8: Performance Comparison Between Grid Search and Leave-One-Out SVM Models (Top Features)

Metric	Grid Search (RBF)	Leave-One-Out (RBF)
Precision (Class CHD)	0.82	0.82
Precision (Class CTRL)	0.83	0.83
Recall (Class CHD)	0.76	0.76
Recall (Class CTRL)	0.88	0.88
F1-Score (Class CHD)	0.78	0.78
F1-Score (Class CTRL)	0.85	0.85
Accuracy	0.82	0.82
Macro Avg F1-Score	0.82	0.82
Weighted Avg F1-Score	0.82	0.82

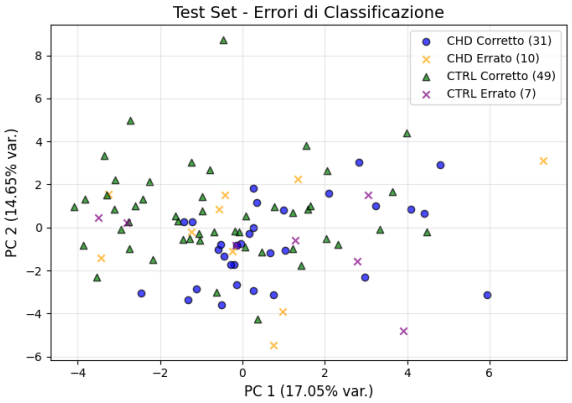


Figure 68: Errors classification plot SVM with GridSearch

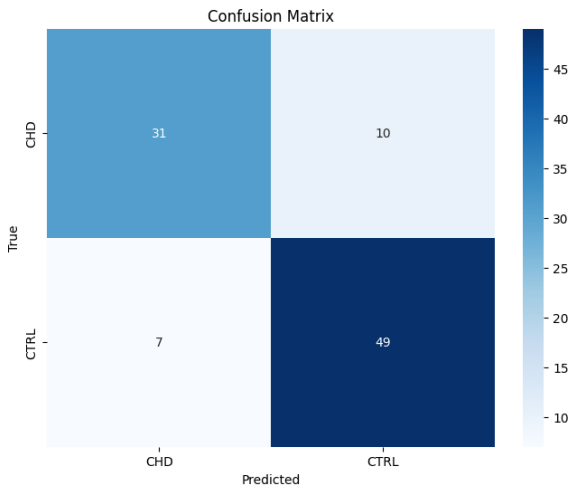


Figure 69: Confusion Matrix of SVM with GridSearch

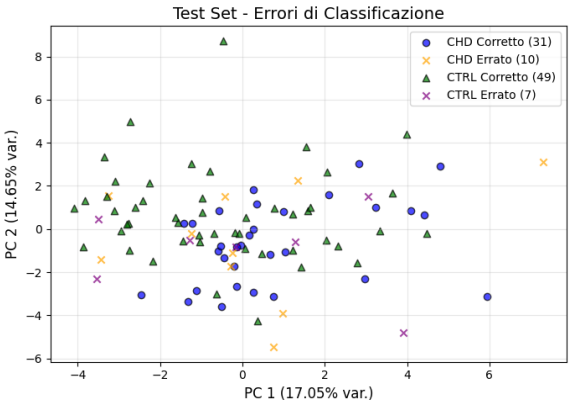


Figure 70: Errors classification plot SVM with LOOCV

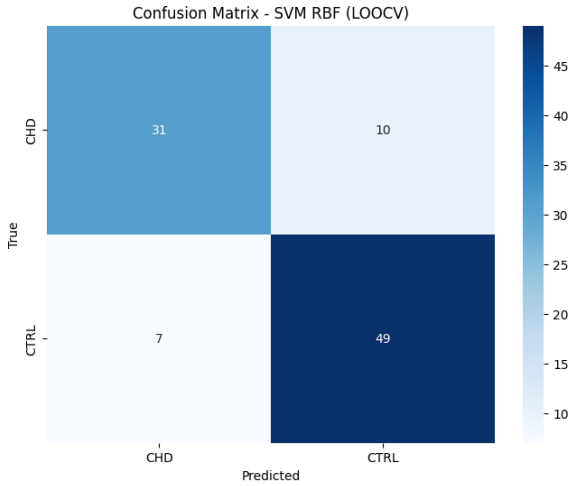


Figure 71: Confusion Matrix of SVM with LOOCV

5.7 RF result

Table 9: Performance Comparison Between Grid Search and Leave-One-Out Random Forest Models

Metric	Grid Search	Leave-One-Out
Precision (Class 0)	0.80	0.79
Precision (Class 1)	0.79	0.78
Recall (Class 0)	0.68	0.66
Recall (Class 1)	0.88	0.88
F1-Score (Class 0)	0.74	0.72
F1-Score (Class 1)	0.83	0.82
Accuracy	0.79	0.78
Macro Avg F1-Score	0.78	0.77
Weighted Avg F1-Score	0.79	0.78

The table presents the performance metrics for the Random Forest models evaluated using Grid Search and Leave-One-Out cross-validation. Both methods yield identical results in this case, indicating that the model performs consistently regardless of the cross-validation technique used. However, the choice between Grid Search and Leave-One-Out should also consider computational cost, as LOO can be resource-intensive for larger datasets

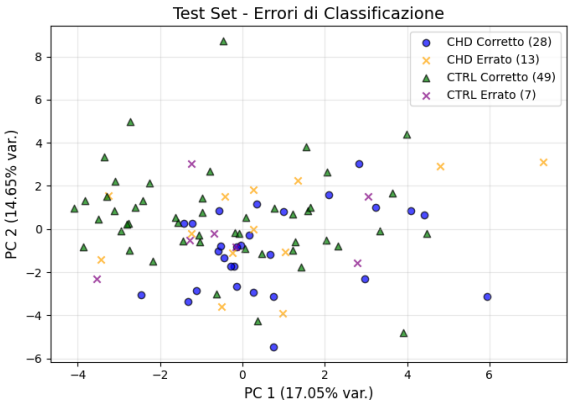


Figure 72: Errors classification plot RF with GridSearch

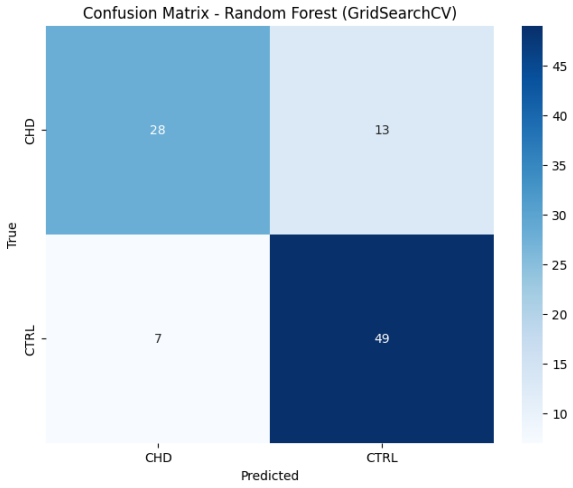


Figure 73: Confusion Matrix of RF with GridSearch

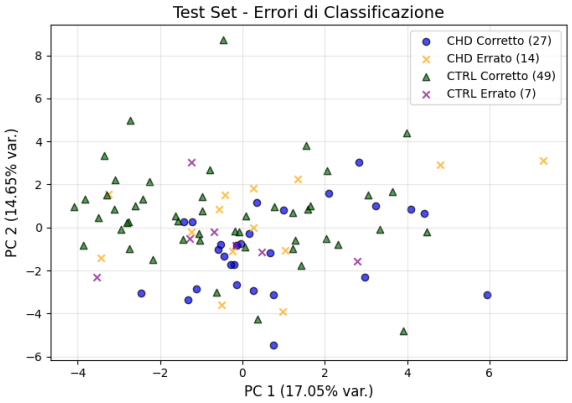


Figure 74: Errors classification plot RF with LOOCV

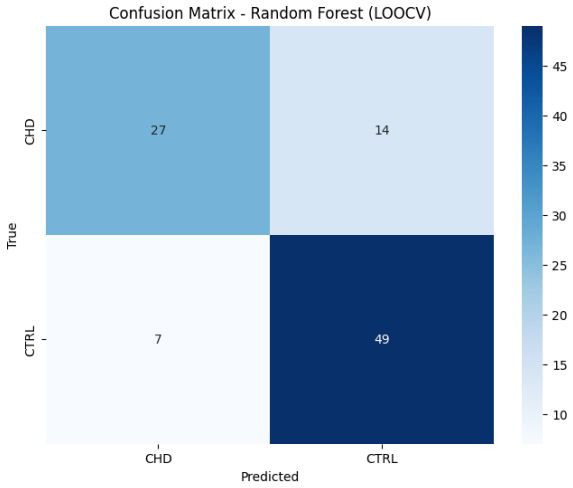


Figure 75: Confusion Matrix of RF with LOOCV

5.8 LR result

Table 10: Performance Comparison Between Grid Search and Leave-One-Out Random Forest Models

Metric	Grid Search	Leave-One-Out
Precision (Class 0)	0.80	0.80
Precision (Class 1)	0.89	0.89
Recall (Class 0)	0.85	0.85
Recall (Class 1)	0.84	0.84
F1-Score (Class 0)	0.82	0.82
F1-Score (Class 1)	0.86	0.86
Accuracy	0.85	0.85
Macro Avg F1-Score	0.84	0.84
Weighted Avg F1-Score	0.85	0.85

The table presents the performance metrics for the Random Forest models evaluated using Grid Search and Leave-One-Out cross-validation. Both methods yield identical results in this case, indicating that the model performs consistently regardless of the cross-validation technique used. However, the choice between Grid Search and Leave-One-Out should also consider computational cost, as LOO can be resource-intensive for larger datasets

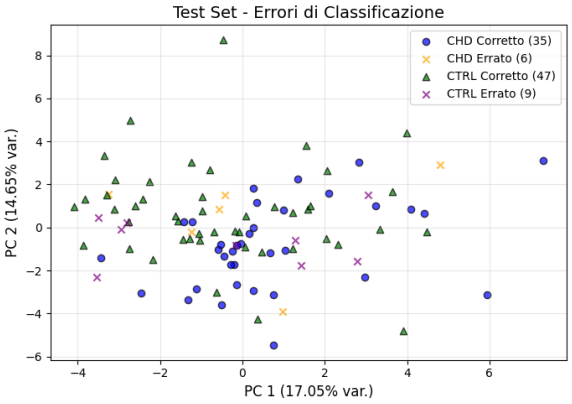


Figure 76: Errors classification plot LR with GridSearch

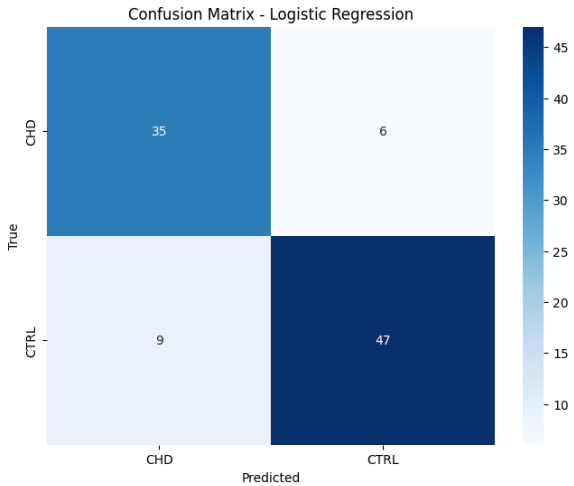


Figure 77: Confusion Matrix of LR with GridSearch

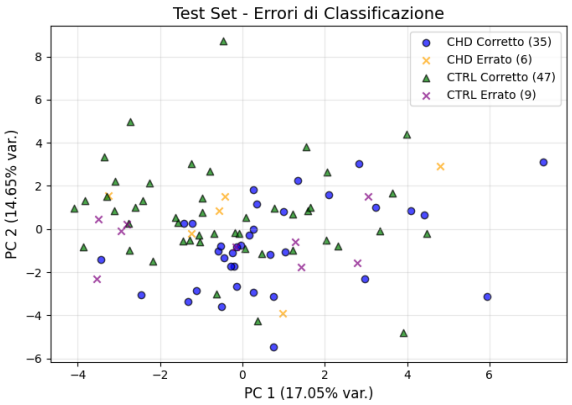


Figure 78: Errors classification plot LR with LOOCV

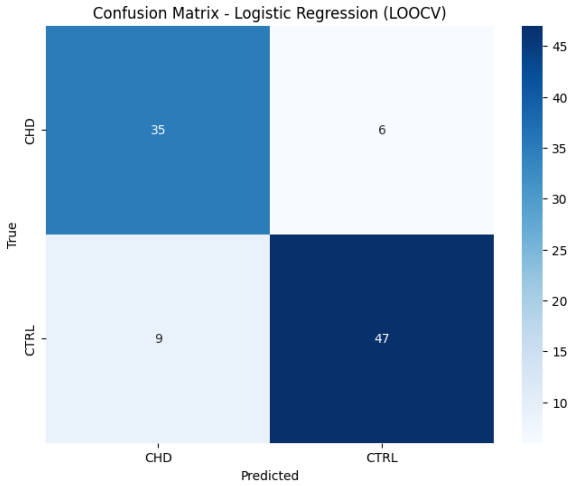


Figure 79: Confusion Matrix of LR with LOOCV

5.9 Median Normalization

Median Normalization is another prevalent method in metabolomics data processing, aimed at standardizing sample readings by adjusting them relative to a median reference value. This technique is particularly useful in reducing skewness in data distributions and mitigating the impact of outliers.

The process of Median Normalization, as implemented in this study, is structured according to the following steps, as detailed in the provided Python function:

- **Separation of Metadata:** Before normalization, any metadata rows such as "m/z meas." are temporarily removed to prevent them from interfering with the normalization process. This ensures that only the actual intensity values are subject to normalization.
- **Calculation of Median Values:** The median intensity for each sample is computed across all features. These median values serve as a normalization reference, ensuring that the scaling of data is centered and consistent.
- **Normalization Process:** Each feature's intensity in each sample is then divided by the respective sample's median intensity. This scales the data to the median value of each sample, normalizing the data across the dataset.
- **Reintegration of Metadata:** If any metadata rows were removed during the initial step, they are reintegrated into the dataset after normalization. This step ensures that the structural integrity and the completeness of the dataset are maintained.

The impact of Median Normalization on the data quality was extensively evaluated

through exploratory data analysis (EDA). This included assessing the normalization effect on the variance and the central tendency of the datasets.

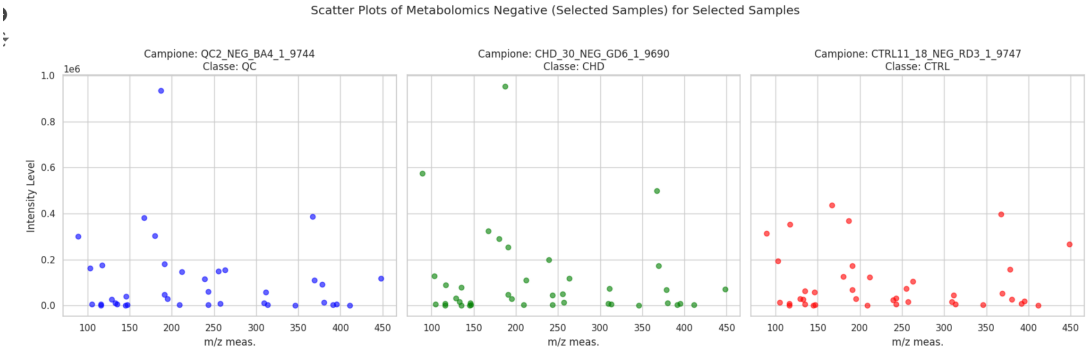


Figure 80: Pre-Norm Visualization Esi- sample

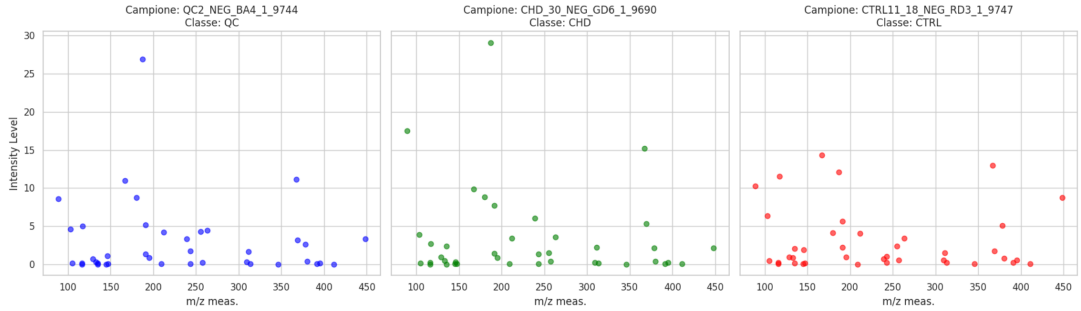


Figure 81: Post-Norm Visualization Esi- sample

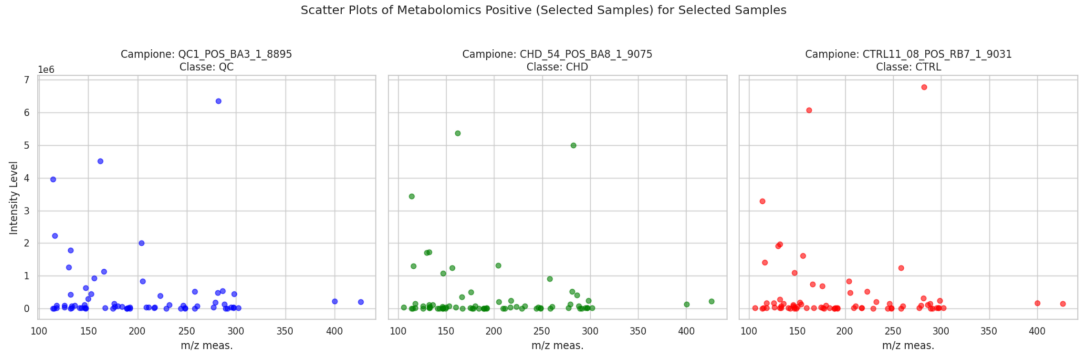


Figure 82: Pre-Norm Visualization Esi+ sample

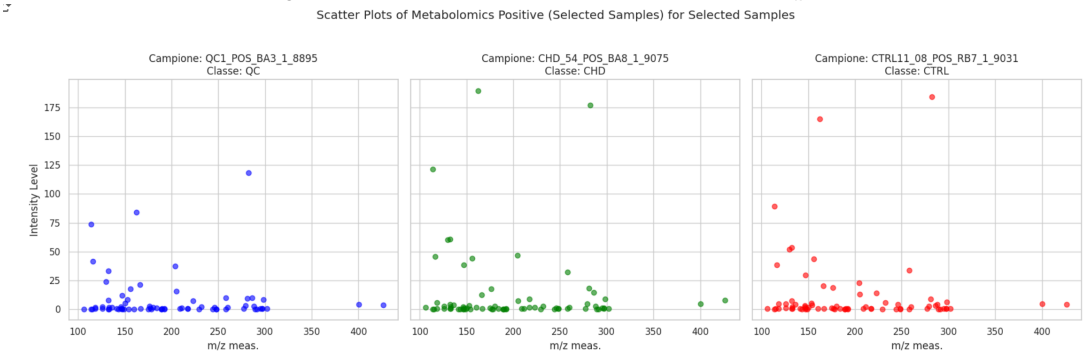


Figure 83: Post-Norm Visualization Esi+ sample

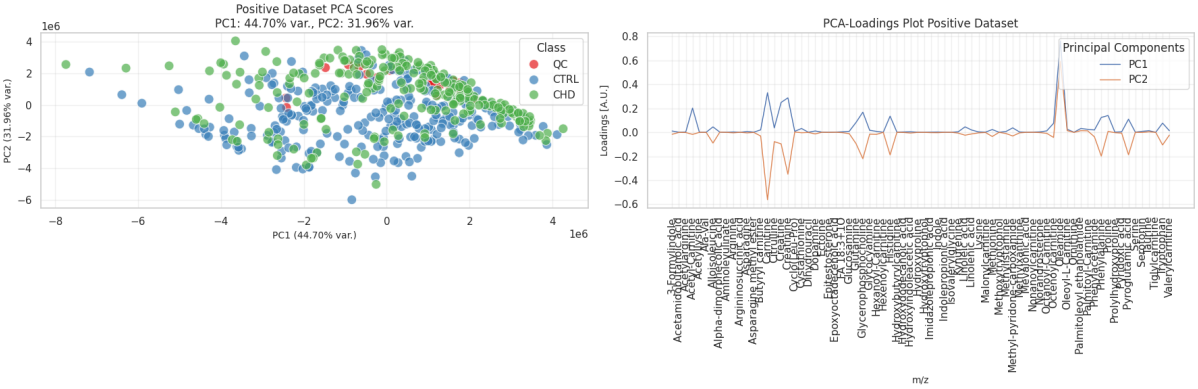


Figure 84: Pre Normalization ESI+ along PC1 and PC2

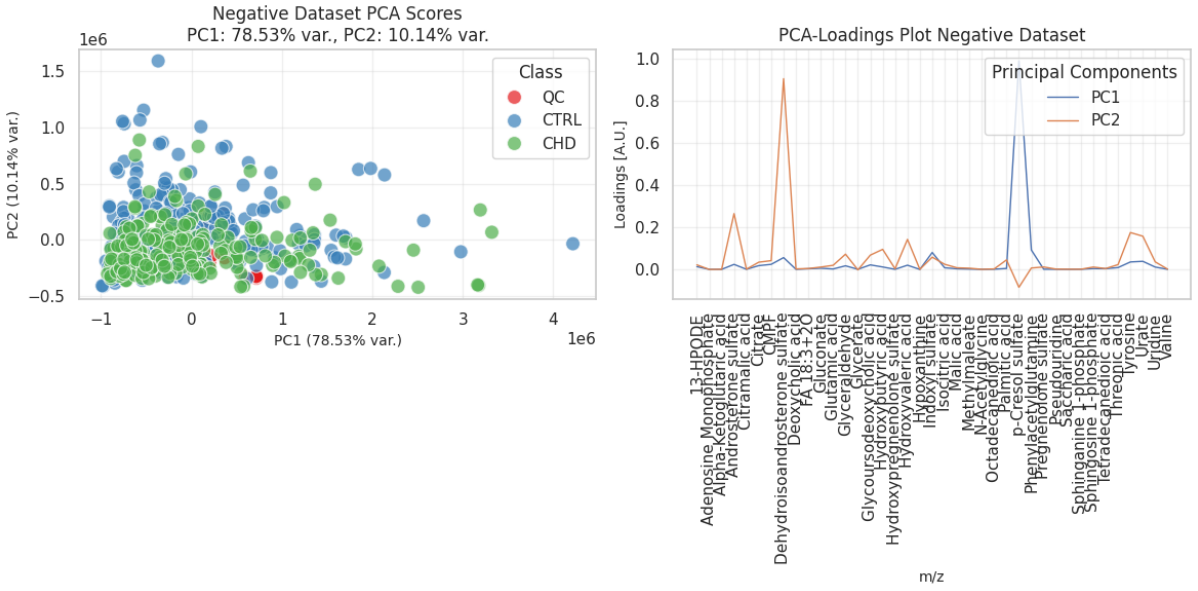


Figure 85: Pre Normalization ESI- along PC1 and PC2

These PCA analyses suggest that median normalization has effectively scaled the data, reducing the skewness caused by outliers and providing a more homogeneous variance across samples. However, the moderate total variance captured by the first two components also suggests that additional components might be necessary to fully understand the data's structure and relationships.

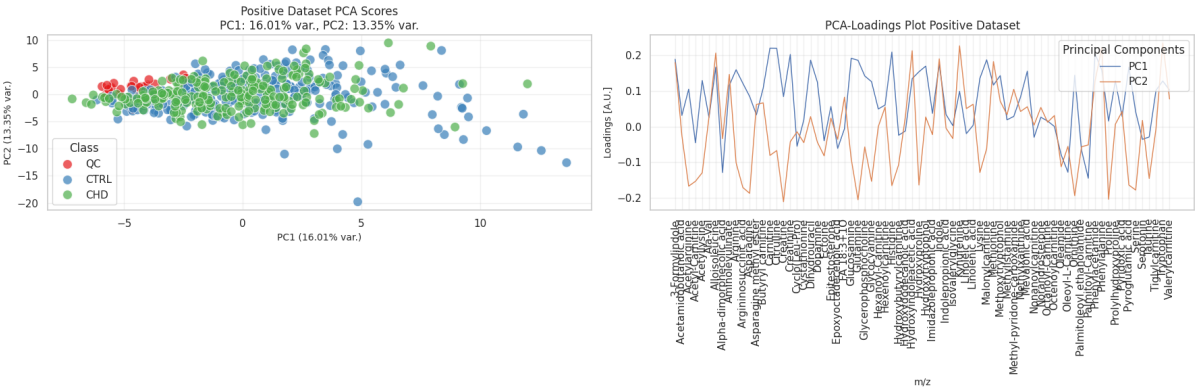


Figure 86: Post Normalization ESI+ along PC1 and PC2

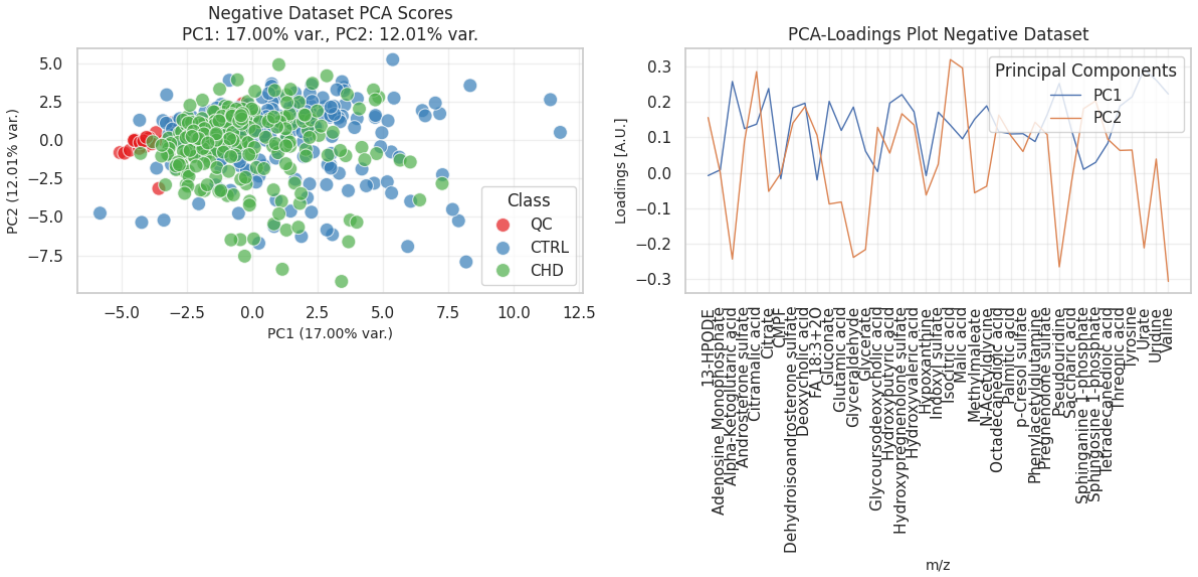


Figure 87: Post Normalization ESI- along PC1 and PC2

5.10 SVM results for MEDIAN normalization

Table 11: Performance Metrics for SVM Models (Grid Search and Leave-One-Out)

Metric	Grid Search	Leave-One-Out
Precision (CHD)	0.78	0.86
Precision (CTRL)	0.82	0.86
Recall (CHD)	0.74	0.79
Recall (CTRL)	0.85	0.91
F1-Score (CHD)	0.76	0.83
F1-Score (CTRL)	0.83	0.88
Accuracy	0.80	0.86
Macro Avg F1-Score	0.80	0.85
Weighted Avg F1-Score	0.80	0.86

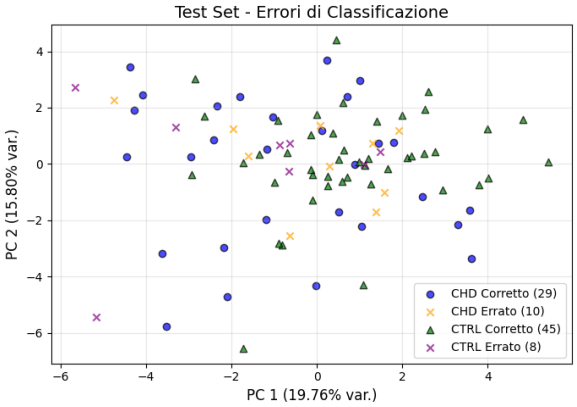


Figure 88: Errors classification plot SVM with GridSearch

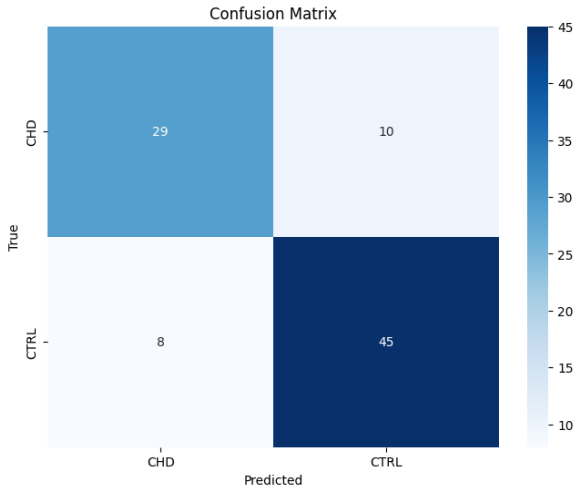


Figure 89: Confusion Matrix of SVM with GridSearch

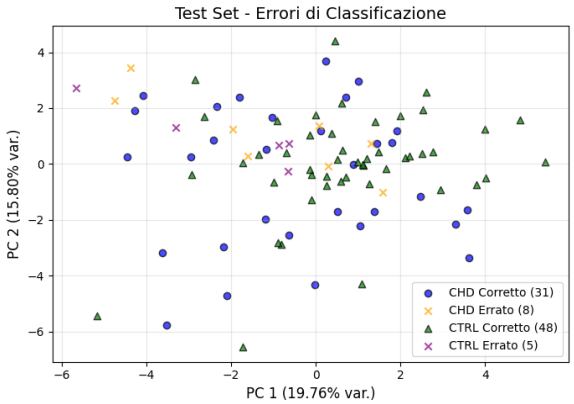


Figure 90: Errors classification plot SVM with LOOCV

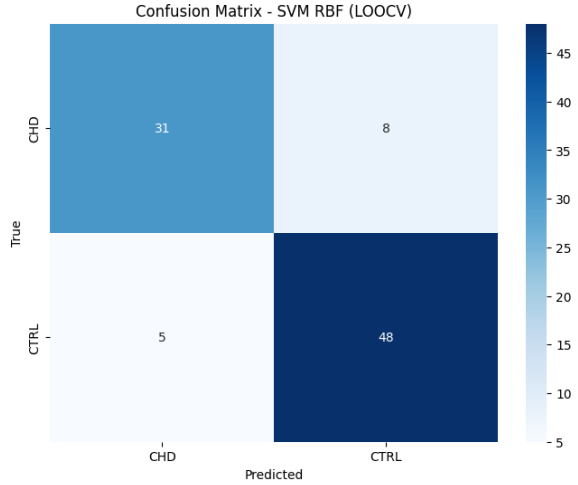


Figure 91: Confusion Matrix of SVM with LOOCV

5.11 RF results for MEDIAN normalization

Table 12: Performance Metrics for RF Models (Grid Search and Leave-One-Out)

Metric	Grid Search	Leave-One-Out
Precision (CHD)	0.82	0.77
Precision (CTRL)	0.85	0.85
Recall (CHD)	0.79	0.82
Recall (CTRL)	0.87	0.80
F1-Score (CHD)	0.81	0.80
F1-Score (CTRL)	0.86	0.83
Accuracy	0.84	0.81
Macro Avg F1-Score	0.83	0.81
Weighted Avg F1-Score	0.84	0.81

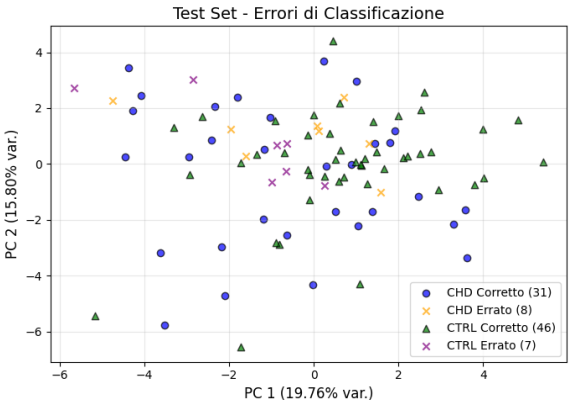


Figure 92: Errors classification plot RF with GridSearch

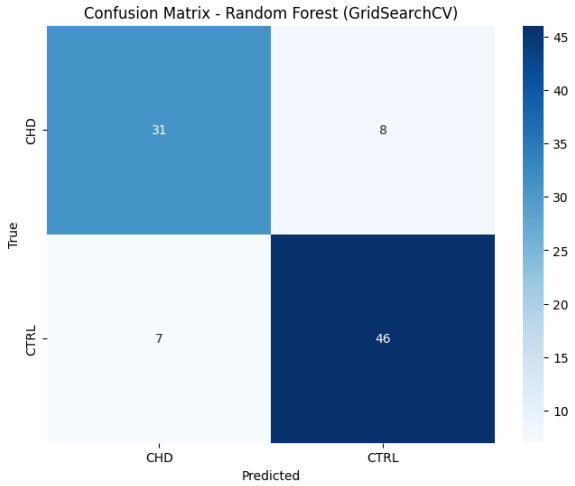


Figure 93: Confusion Matrix of RF with GridSearch

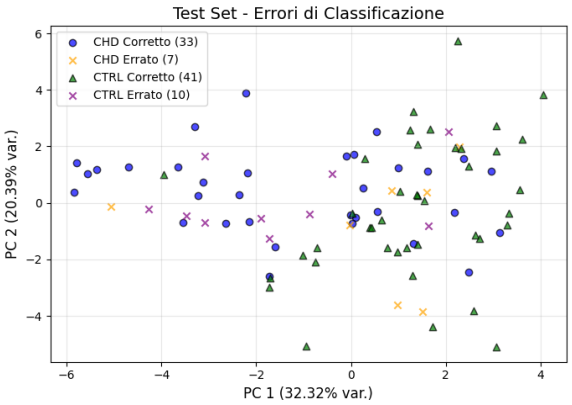


Figure 94: Errors classification plot RF with LOOCV

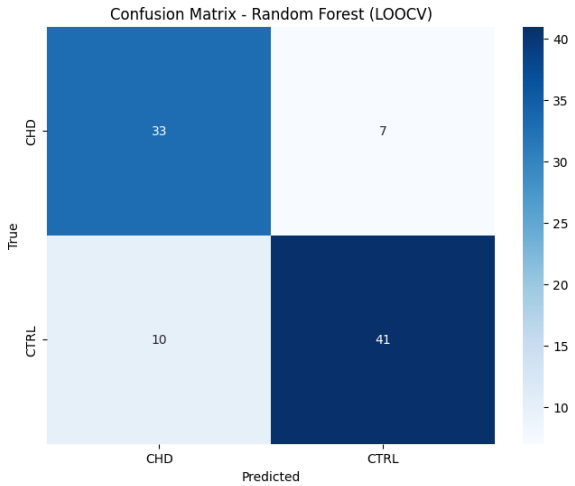


Figure 95: Confusion Matrix of RF with LOOCV

5.12 LR results for MEDIAN normalization

Table 13: Performance Metrics for LR Models (Grid Search and Leave-One-Out)

Metric	Grid Search	Leave-One-Out
Precision (CHD)	0.79	0.79
Precision (CTRL)	0.83	0.83
Recall (CHD)	0.77	0.77
Recall (CTRL)	0.85	0.85
F1-Score (CHD)	0.78	0.78
F1-Score (CTRL)	0.84	0.84
Accuracy	0.82	0.82
Macro Avg F1-Score	0.81	0.81
Weighted Avg F1-Score	0.81	0.81

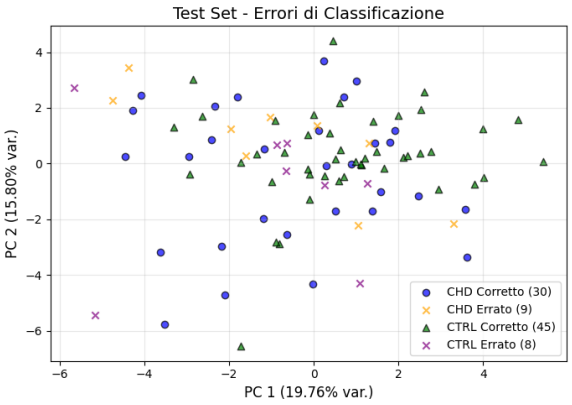


Figure 96: Errors classification plot LR with GridSearch

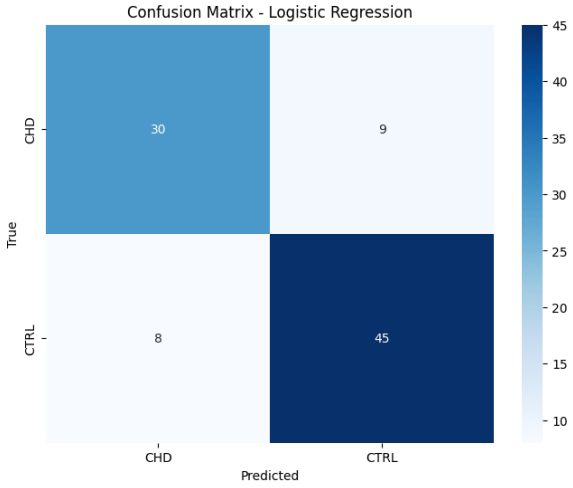


Figure 97: Confusion Matrix of LR with GridSearch

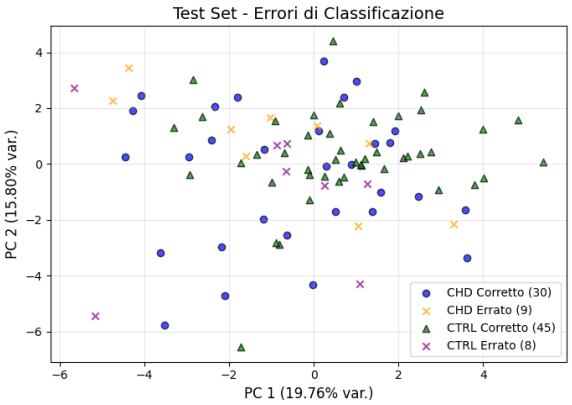


Figure 98: Errors classification plot LR with LOOCV

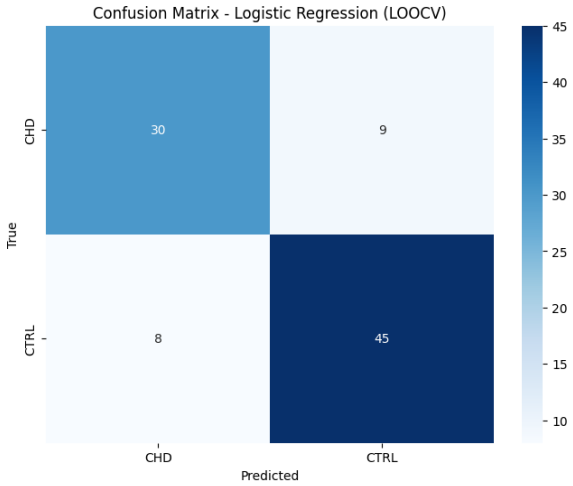


Figure 99: Confusion Matrix of LR with LOOCV

5.13 Model Comparison

When comparing models processed via different normalization techniques—PQN, TIC, and Median Normalization—the selected model using Median Normalization (Median) proved most effective for several key reasons outlined in the analysis:

1. **Robustness to Outliers:** Median Normalization demonstrates higher robustness to outliers compared to PQN. The median, being a measure of central tendency less affected by extreme values, enhances the model's resilience against anomalous data points that could skew the data distribution.
2. **Enhanced Predictive Stability:** The model's performance stability, particularly in metrics such as precision, recall, and the F1 score, indicates that Median Normalization can handle diverse dataset characteristics more effectively. This stability is crucial for reliable predictions in real-world applications where data anomalies are common.

Conclusion: Based on the comparative analysis, the SVM model employing Median Normalization with LOOCV is chosen for its superior performance and reliability in handling datasets with anomalies and outliers. This choice is justified by the increased robustness and improved predictive stability that Median Normalization provides, making it highly suitable for practical applications in real-world scenarios.