User Manual

Profiler Software

(Omics Data Analysis)

Table des matières

Document history

Revision	Author(s)	Changes	Effective date
0.0	Y. Zirem, L. Ledoux	Creation	2025/05/13
1.0	L. Ledoux	Updates	2025/06/23

1- Introduction

In the fast-paced world of biomedical research, the complexity of data is increasing rapidly. Researchers are generating vast amounts of omics data, yet the analytical bottleneck remains a significant challenge. Profiler, a cutting-edge software developed by the **PRISM U1192 Lab** at the University of Lille, provides a powerful and user-friendly solution to address this challenge.

The proliferation of omics technologies, including mass spectrometry-based proteomics, RNA sequencing and metabolomics, has revolutionized biomedical research. However, the heterogeneity of omics datasets and the computational expertise required for their analysis continue to pose significant hurdles. Although several tools are available, they are often fragmented, domain-specific, or require programming skills, limiting their accessibility to non-specialist users.

We introduce **Profiler**, a web-based software platform developed to streamline and unify the analysis of multi-omics datasets. Profiler is designed to be both comprehensive and accessible, integrating essential workflows from data conversion to advanced machine learning and survival analysis.

Developed by **Yanis Zirem**, a second-year PhD student (2025), under the supervision of **Prof. Michel Salzet and Prof. Isabelle Fournier**, Profiler aims to democratize high-throughput data analysis through a modular and intuitive web-based interface.

Why use Profiler ?

- **Multi-Omics Compatibility**: Seamlessly handle data from mass spectrometry, transcriptomics, metabolomics, and more.
- **Raw Data Conversion**: Effortlessly convert vendor-specific mass spectrometry formats (Bruker, Waters, Thermo Fisher) into open formats like mzML, mzXML, mzDB, or mz5.
- **Preprocessing Made Simple**: Normalize, filter, bin, correct batch effects, and impute missing values with built-in preprocessing tools, no coding required.
- **Smart Data Exploration**: Visualize distributions, correlations, similarities, and feature spectra across classes with ease.
- **Integrated AI & Statistics**: Train over 23 machine learning models, deploy deep learning architectures, and apply classical statistical tests, all in one place.
- **Biomarker Discovery & Explainability**: Use SHAP, LIME, and volcano plots to identify and interpret predictive biomarkers.
- **Survival Analysis Tools**: Perform Kaplan-Meier and Cox regression analysis directly within the platform.
- **Pathway Enrichment Analysis**: Gain insights into biological pathways with integrated enrichment analysis with a hundred databases.
- **Wizard Mode Automation**: Run real-time predictions or conduct post-hoc analyses with just a few clicks.

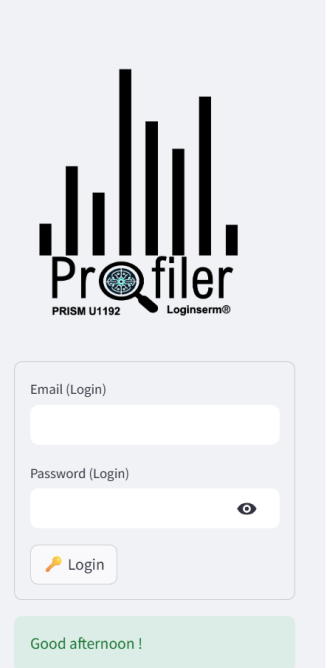
Who is it for ?

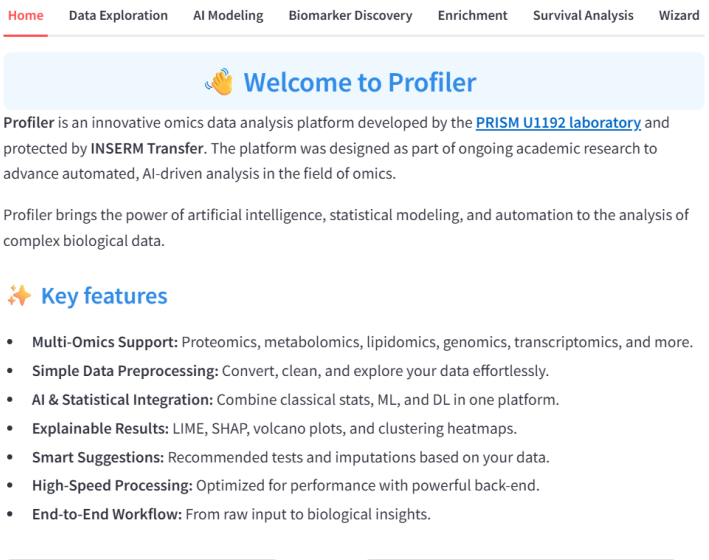
- **Researchers** looking for an all-in-one omics analysis platform.
- **Clinicians** interested in biomarker discovery and prognosis modeling.
- **Students and bioinformaticians** wishing to learn or prototype pipelines.
- **Core facilities** seeking reproducible and shareable workflows.

All features of Profiler are detailed in this User Manual, which includes step-by-step explanations and illustrative screenshots.

Note: To access the entire software, simply create an account (e-mail address + password required), and you'll have unlimited access to the software.

⚠ **Note:** The session is reset by the 'Session Management' expander when the software becomes slow or when a new dataset needs to be used — all without requiring a logout.





2- Side bar

Data Conversion

Profiler allows users to convert raw data from various mass spectrometry instruments vendors (Bruker, Waters, and Thermo Fisher) into standardized formats such as mzML, mzXML, mz5, and mzDB.

To perform a conversion, simply drop raw files zipped (**1**). Then, choose the input file type (**2**) and the desired output format (**6**).

Several conversion options are available:

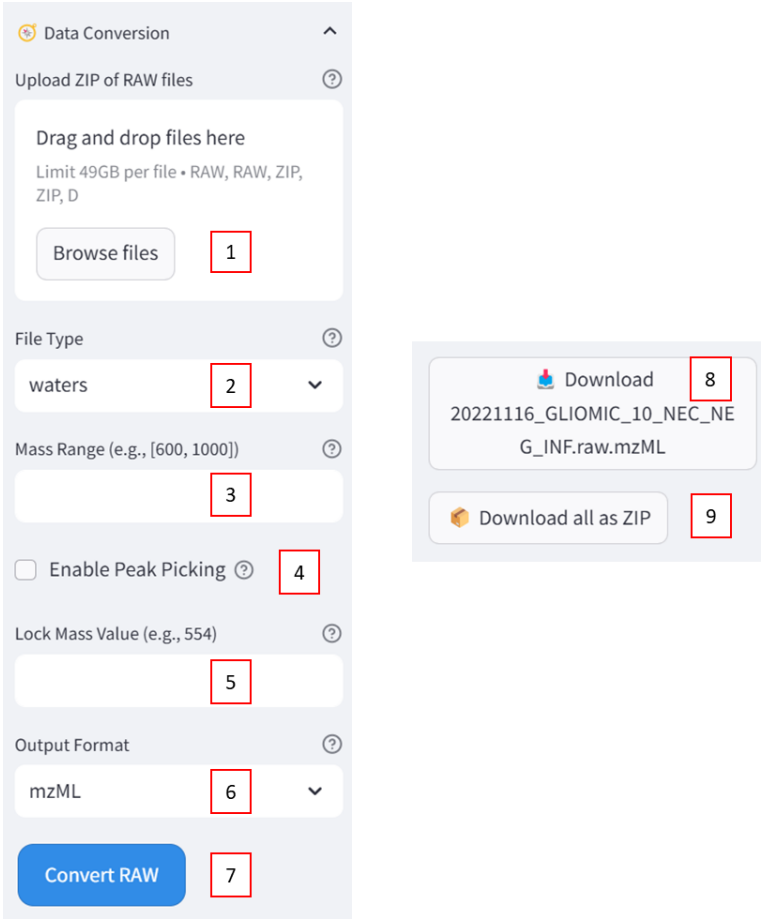
Mass range (**3**): If you wish to convert only a specific mass range, indicate it in the format [600,1000]. Leave this field blank to convert the entire mass range.

Enable peak picking (**4**): Check this option if peak picking is required.

Lock mass (**5**): Specify a lock mass if one was used during the mass spectrometry acquisition to improve mass accuracy during conversion. This option is only available for Waters files as Waters instruments have a lock mass feature integrated into their data acquisition process.

When the Convert RAW button (**7**) is clicked: If successful, a success message is displayed. If an error occurs, an error message is displayed with the exception details.

The converted files can then be downloaded one by one (**8**) or all together az ZIP file (**9**).



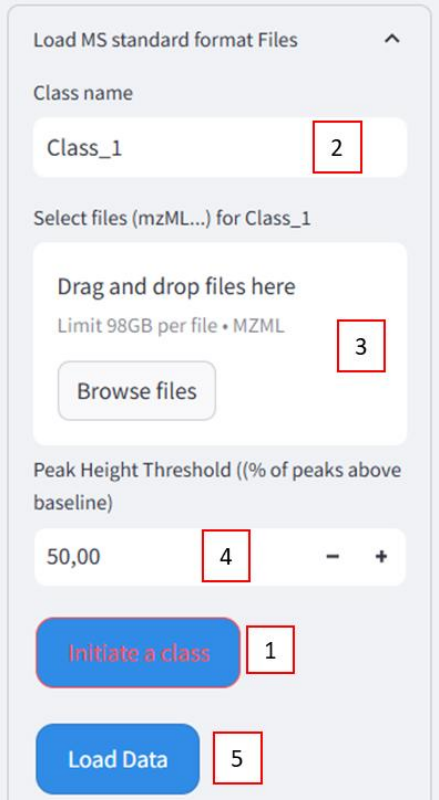
Load MS standard format Files

This second expander is used to load previously converted MS data so that it's possible to:

- Start data analysis.
- Create a CSV file which can then be loaded into the third "Load Structured Data" section.

First, a Class (**1**) needs to be initiated for each category to be included in the dataset. For each class, simply enter its name (**2**) and browse all the corresponding files to be associated with it (**3**).

To apply a peak height threshold (**4**) (chromatogram peak selection according to the baseline), just choose the % according to your data (% of peak above baseline).



Once all files have been assigned to their respective classes, click on the "Load data" button to import the data (**5**).

Note: If the aim is to create a csv file to keep and reuse at will, just pre-process the data, with or without normalization. This will produce an excel file of all the initial data to be saved as a csv easily.

Load Tabular Data

The purpose of this third expander is to import pre-structured files (metabolomics, proteomics and transcriptomics):

- either previously created with the software and saved in CSV format (Scenario 1).

- either obtained from other software in CSV, XLSX, TXT and TSV formats, such as directly processed outputs from software like MaxQuant, DIA-NN and Perseus (Scenario 2).
- or obtained from other software in the same formats but not from this software (Scenario 3).

Scenario 1

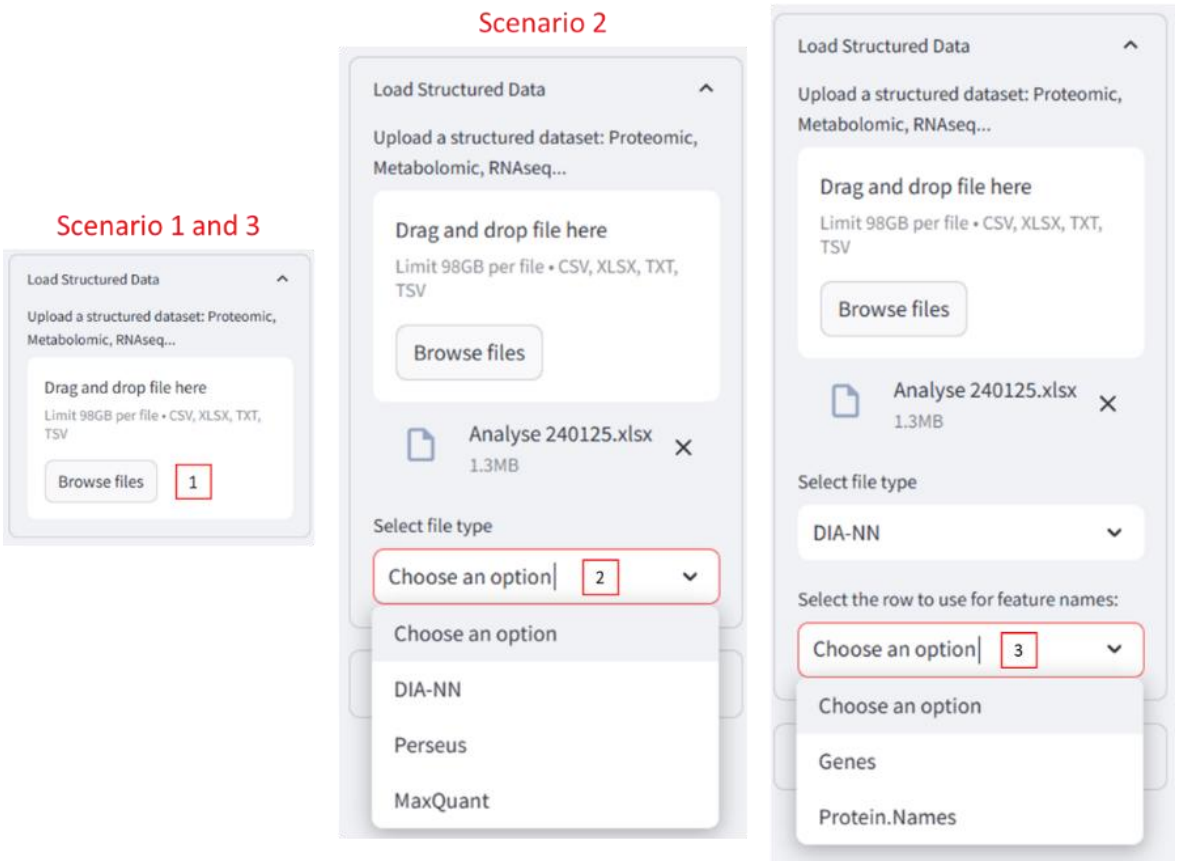
When the csv file has been created using Profiler, the structure will already be fully adapted for loading using this section. Simply load the file directly (**1**).

Scenarios 2 and 3

To be loaded, a file (in any format accepted) must contain a first column named “Class” and the features (such as names of genes, proteins, ions ...) in the following columns. In addition, the required format is precised (blue square).

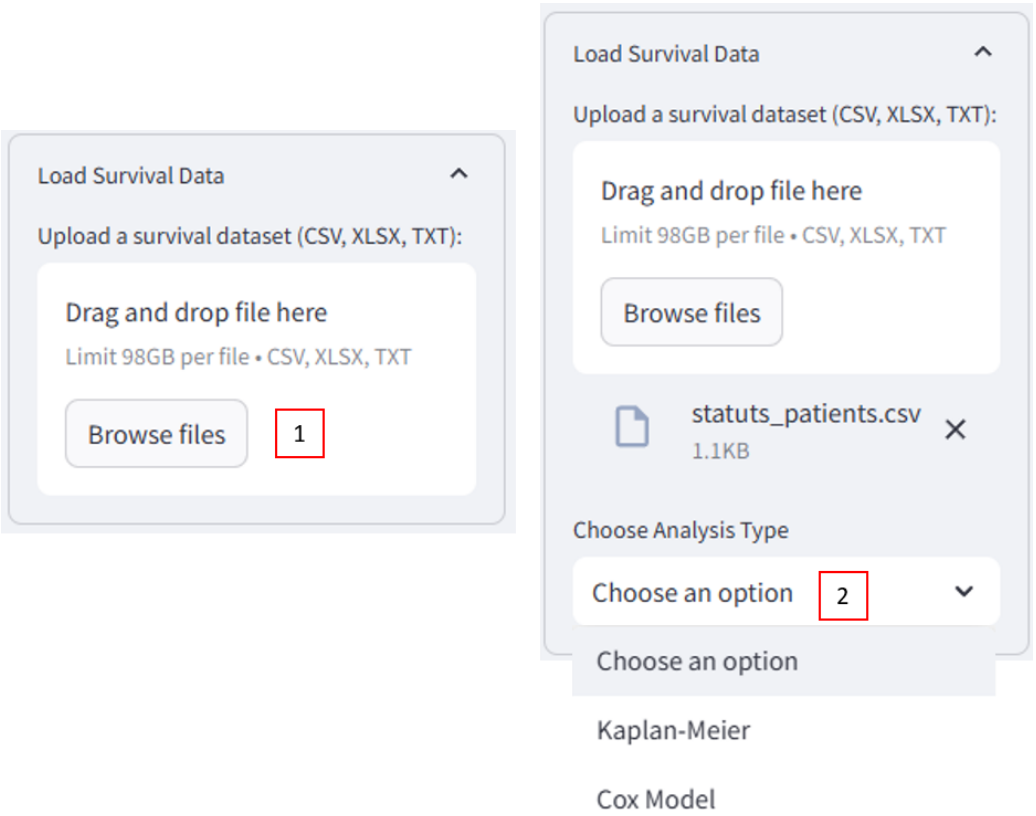
If there is no “Class” column, a dialog box will be displayed and then there are two scenarios:

- either the file is from MaxQuant, DIA-NN and Perseus and it possible to select the type of file (between this three options) and the file will be structured automatically to enable the importation (**2**). In addition, another dialog box will allow you to keep either the gene names or the protein names as features (**3**).
- Or the file is from any of these softwares. In this case, the file needs to be structured manually, adding the “Class” column and putting features in columns too. After Once structured, the file can be loaded in the usual way, as explained in Scenario 1.



Load Survival Data

This fourth expander allows users to upload datasets (**1**) to perform survival analyses (either Cox model or Kaplan Meier (**2**)).



This file should be well structured. Indeed, there is two possibilities:

- All required columns (Overall survival, State) are present, data is loaded successfully.
- Some columns are missing; an error message appears.

The two required columns are:

- “Overall survival” column, corresponds to the duration a patient remains alive from a defined starting point.
- "State" column, corresponds to the survival status: 0 indicates the patient is alive, while 1 represents the event of death.

In addition, the required format is specified (blue square).

3- Main Menu

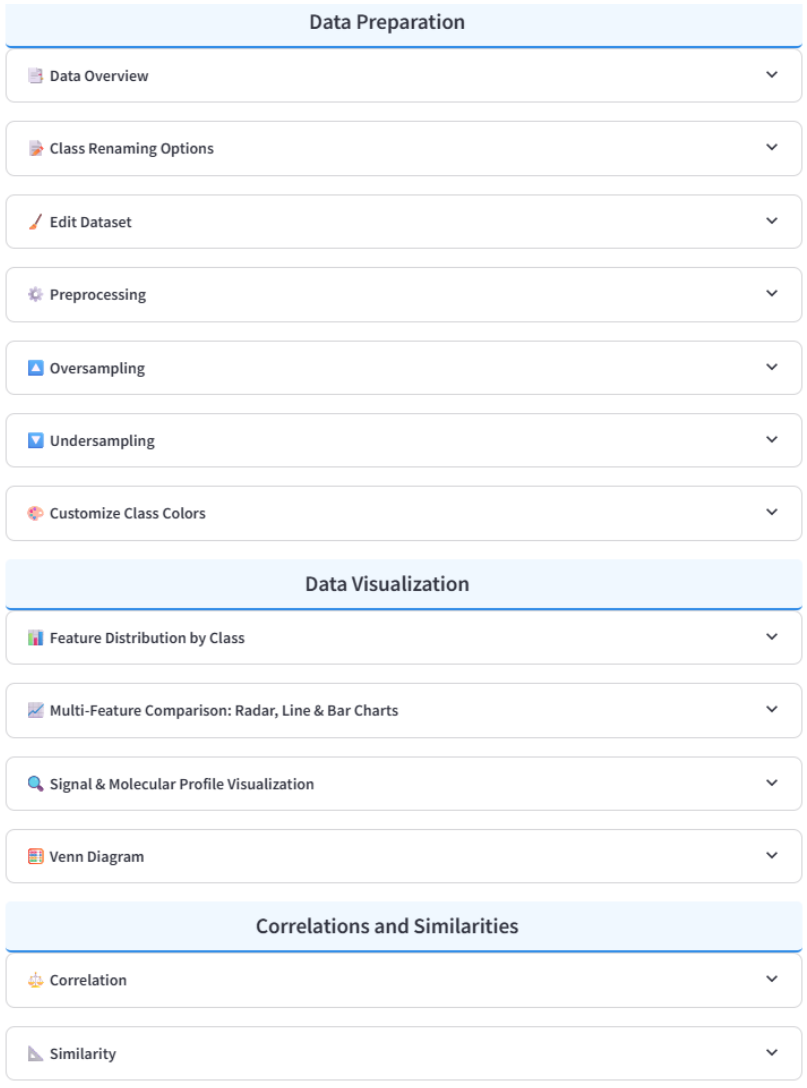
The main interface offers seven Tabs; Home, Data Exploration, AI modeling, Biomarker Discovery, Enrichment, Survival Analysis and Wizard.

Home

This first tab provides an overview of the software, including guidance on how to cite it, access to available resources such as this documentation and test data, and information on how to request support in case of errors during usage.

Data Exploration

This second tab is divided into 3 sub-categories: data preparation, data visualization and correlations and similarities.



Data Preparation

This sub-category is itself divided into seven expanders, which will be explained one by one throughout this tutorial.

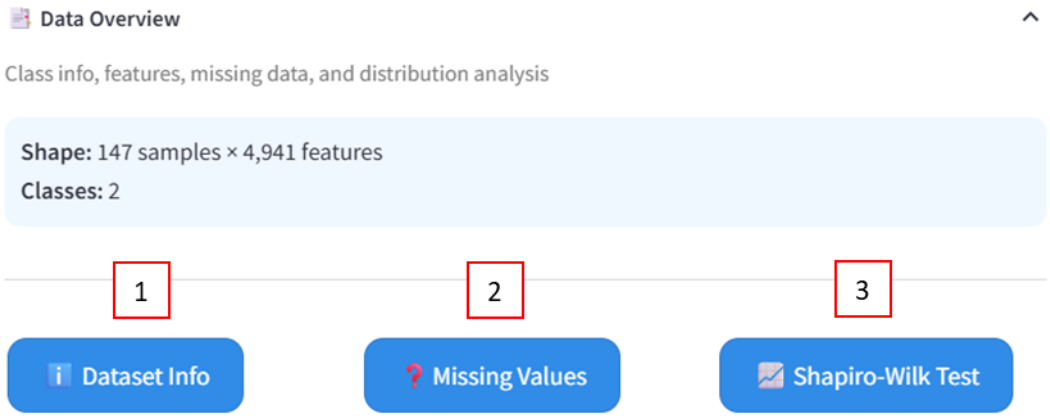
Data overview

This section provides an overview of the dataset, including the total number of classes and features (**1**). When clicking on Dataset Info, it displays the number of replicates per class in a structured format, allowing to easily verify class distribution. A pie chart visualization provides a clear view of the proportion of each class in the dataset. Additionally, an interpretation is included to assess whether the classes are balanced or imbalanced, along with recommendations in case of imbalance.

It allows to assess data completeness by viewing the number of missing values (NaN) in the entire dataset and for each individual feature (**2**). Both counts and percentages are automatically calculated, helping to determine whether imputation or additional preprocessing steps are necessary.

The final button (**3**) allows to perform a feature-wise normality assessment using the Shapiro-Wilk test. Upon clicking, a summary is generated showing the number of features that follow a normal distribution and the average p-value. If the majority of features are not normally distributed, a recommendation for appropriate imputation methods (e.g., median imputation) is provided.

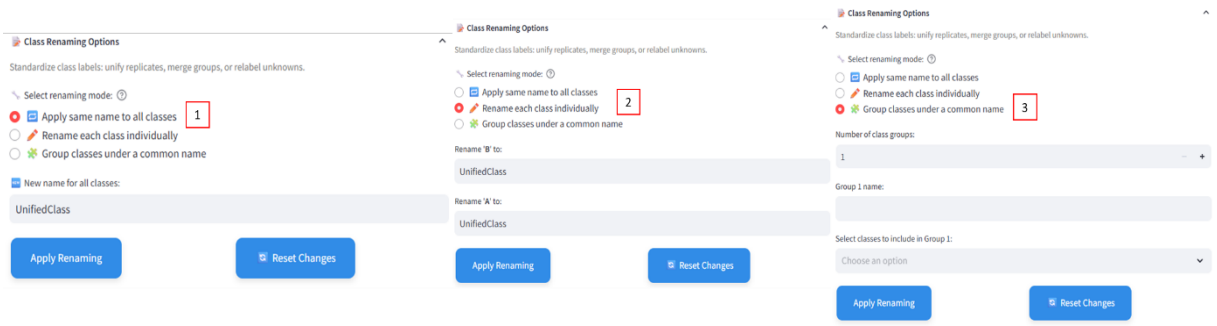
Based on the normality results, a list of recommended statistical tests, parametric or non-parametric, is displayed. Additional guidance is included to explain when normality assumptions can be relaxed, such as through the application of the Central Limit Theorem.



Class renaming options

This second expander allow different changes:

- Unify replicates by selecting the following method « apply the same name to all classes » and writing the new name for all classes (**1**).
- Invert or change labels by selecting the following method « Rename each class individually » and putting the new name for each class (**2**).
- Group classes for a common name (**3**) by choosing the number of groups to rename and putting the common name for each new group.

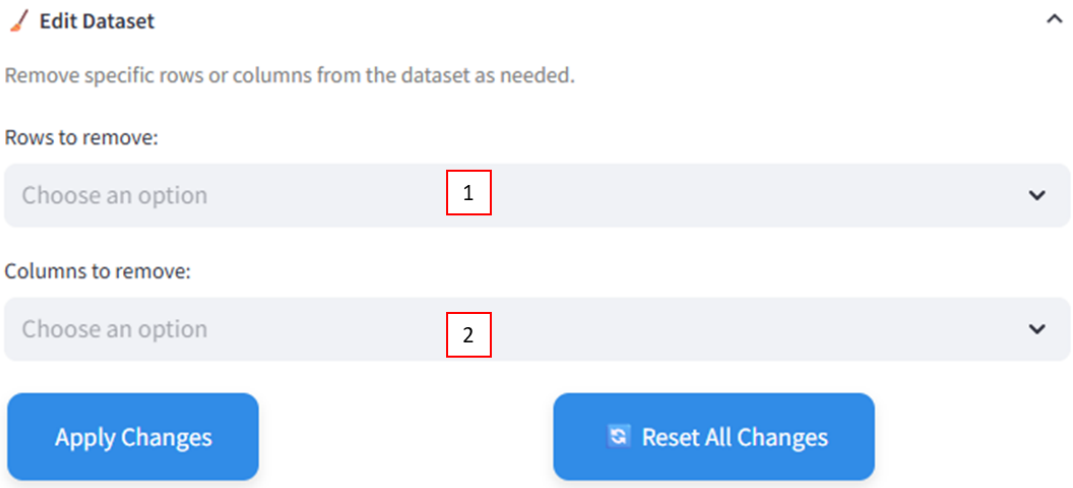


⚠ **Note:** A "Reset" button allows you to return to the original data if an error is made.

Edit dataset

This third expander is only used to remove a row (**1**) or a row (**2**) from the dataset if required.

Just select the one to be removed and press the « apply changes » button. An updated preview of the data is displayed.



⚠ **Note:** A "Reset" button allows you to return to the original data if an error is made.

Preprocessing

The fourth expander is dedicated to data preprocessing, including binning, mass range delimitation, normalization, and missing value imputation.

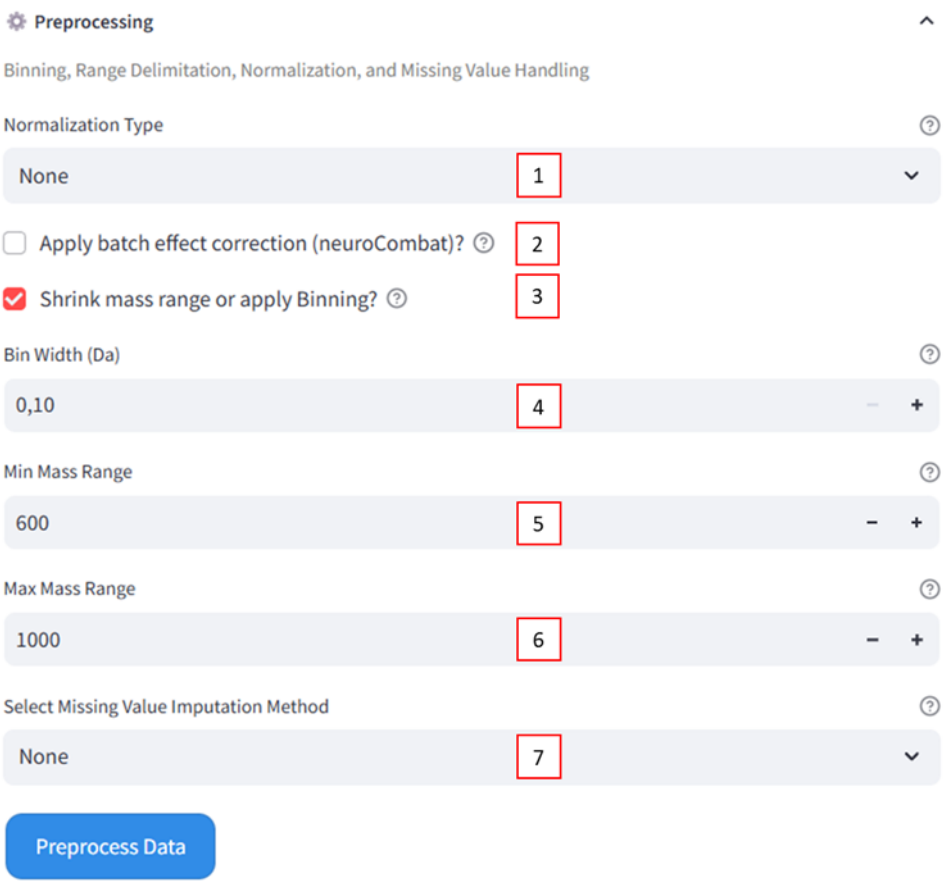
Several normalization methods (**1**) are available, such as TIC (Total Ion Count), RMS (Root Mean Square), BasePeak, QNorm, Log Normalization, Log10 and Log2.

Note: To assist in the selection of an appropriate normalization method for datasets, a brief explanation of each approach is provided.

- **TIC (Total Ion Current)**: The intensity of each feature (ion) is scaled by the total intensity of all features in the sample. This makes the data relative to the total signal, adjusting for differences in overall signal intensity. It is useful when differences in total intensity across samples need to be corrected. For example, if one sample has a higher total intensity, TIC ensures that the intensities of individual peaks are made comparable across all samples.
- **RMS (Root Mean Square):** The data is normalized by the root mean square (RMS) of each sample or spectrum. The RMS is calculated by taking the square root of the

average of the squared values in a sample. This method is useful when the data needs to be standardized in a way that accounts for both the magnitude and the spread of values in each sample, making them comparable even when overall intensity varies.

- **BasePeak normalization**: Each intensity value is normalized by dividing it by the highest intensity value in that sample. The "BasePeak" is identified as the highest intensity peak for a sample, so all other peaks are scaled relative to it.
- **Log2 normalization**: A base-2 logarithm ($\log_2(1 + x)$) is used. This approach is useful when data exhibits a strong multiplicative relationship, such as doubling or halving of quantities. It is often applied when changes are analyzed in terms of "fold changes"—for example, how many times a value increases or decreases.
- **Log transformation**: Log normalization takes the natural logarithm of the data. The formula is log(1 + x). It makes patterns in the data more apparent and reduces the impact of outliers.
- **Log10 transformation**: is similar to log normalization but uses the base-10 logarithm (log10(1 + x)). It's particularly useful for transforming exponential data.
- **QNorm (Quantile Normalization)**: Values in each sample are ranked and then adjusted to share the same rank-based distribution. This method is widely used in genomics to ensure that the distribution of values across different datasets is made consistent.



NeuroCombat can be used to remove batch effects based on 'class' (**2**).

Note: NeuroCombat is a specialized tool used for correcting batch effects in datasets. Is an adaptation of the Combat algorithm, which was originally developed for genomic data. This method models the batch effect as a location and scale adjustment, effectively removing the unwanted variation while preserving the biological variability of interest. When enabled, NeuroCombat will adjust the data based on the specified 'Class' variable, ensuring that subsequent analyses are not confounded by batch effects.

It also possible to adjust the mass range and/or apply binning to reduce data dimensionality by ticking « Shrink mass range or apply binning » (**3**). Smaller values for the bin width give finer resolution but more extensive computationally (**4**). For mass range delimitation, precise the min mass range (**5**) and the max mass range (**6**).

Additionally, the type of imputation method for handling missing values can be selected (**7**). The available methods are the following: Mean, median, mode and KNN imputation in addition to delete the missing values.

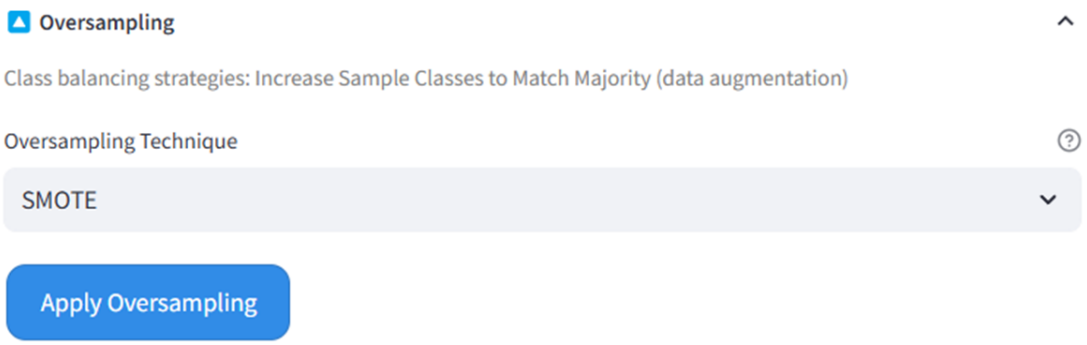
Note: To assist in the selection of an appropriate imputation method for datasets, a brief explanation of each approach is provided.

- **Mean Imputation**: This method replaces missing values with the mean of the observed values for that variable. It is simple and preserves the mean of the data. It suitable when the variable (feature) fellow normal distribution.
- **Median Imputation**: Similar to mean imputation, this method replaces missing values with the median of the observed values. It is more robust to outliers compared to mean imputation. It suitable for data that not fellow normal distribution.
- **Mode Imputation**: This method replaces missing values with the mode (most frequent value) of the observed values. It is typically used for categorical data.
- **Delete Missing Values**: This method removes any columns (features) with missing values. While it ensures that the analysis is performed on complete data, it can lead to a significant loss of information if missing values are widespread.
- **KNN Imputation**: K-Nearest Neighbors (KNN) imputation replaces missing values with the mean or weighted mean of the k-nearest neighbors. This method considers the similarity between observations and can provide more accurate imputations compared to simple mean or median imputation.

After clicking the « Preprocess Data » button, a progress bar will appear, providing feedback on the preprocessing stages.

Oversampling

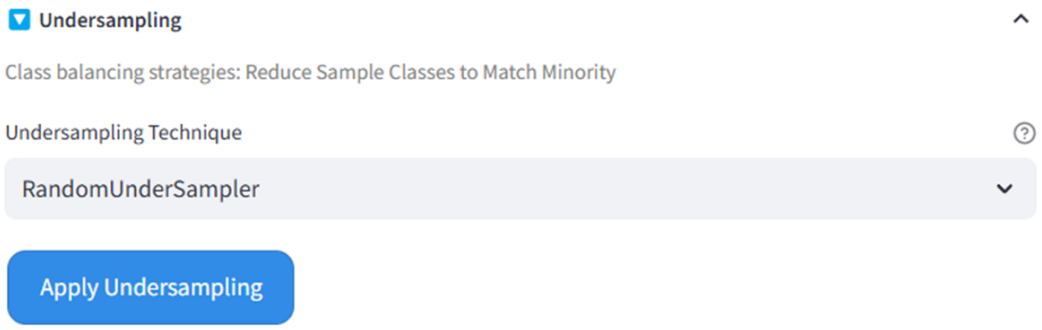
This fifth expander aims to oversample the dataset. Indeed, two techniques are available to increase the data in minority classes (data augmentation): SMOTE and ADASYN.



Note: In the context of biological data, such as comparing healthy (control) and cancer data, techniques like SMOTE (Synthetic Minority Over-sampling Technique) and ADASYN (Adaptive Synthetic Sampling) are employed to address imbalanced datasets in machine learning called also data augmentation techniques.
SMOTE works by generating synthetic examples of the minority class, rather than simply duplicating existing ones. It does this by selecting a sample from the minority class and creating new samples along the lines connecting this sample to its nearest neighbors. This approach helps balance the dataset, making it easier for machine learning models to learn from both minority and majority classes.
ADASYN, on the other hand, is an extension of SMOTE but with a focus on the difficulty of classification. It generates synthetic samples for the minority class, similar to SMOTE, but places more emphasis on minority instances that are harder to classify. Specifically, ADASYN generates more synthetic samples for the minority class that are near the decision boundary, where the model might struggle the most. This targeted approach helps improve the classification performance by focusing on the most challenging cases.
In **summary**, SMOTE treats all minority class samples equally, whereas ADASYN prioritizes samples that are harder to classify, making it particularly useful in scenarios where the boundary between different samples is complex.

Undersampling

Conversely, this sixth expander aims to reduce the data in majority classes to match minority. Two techniques are also available: RandomUnderSampler and NearMiss.



Note: Unlike SMOTE and ADASYN, which generate synthetic samples to augment the minority class, RandomUnderSampler and NearMiss work by decreasing the number of samples in the majority class. This approach can be particularly useful when the majority

class is significantly larger than the minority class, leading to a biased machine learning model. **RandomUnderSampler** reduces the number of samples in the majority class by randomly selecting a subset of these samples. The size of this subset is chosen to match the number of samples in the minority class. **NearMiss** is a more sophisticated under-sampling technique that selects samples from the majority class based on their proximity to samples in the minority class.

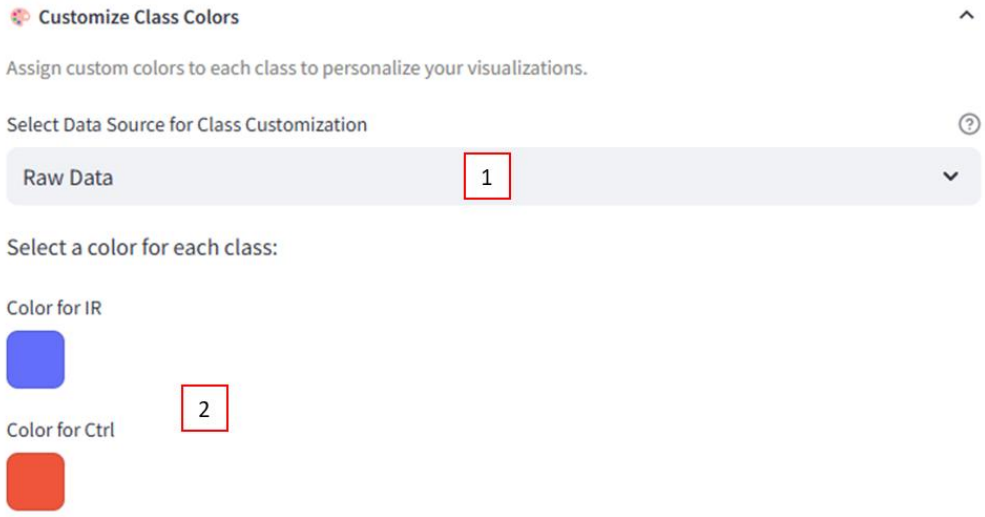
These last two expanders are both designed to balance the classes, thus improving classification model performances.

Customize Class Colors

The last expander is used to customize the class colors.

A data source (**1**); raw data, preprocessed data, oversampled data, or undersampled data; must be selected, and a color assigned to each class present in the dataset (**2**).

These choices will be retained and applied consistently across all analyses performed within the software.



Data Visualization

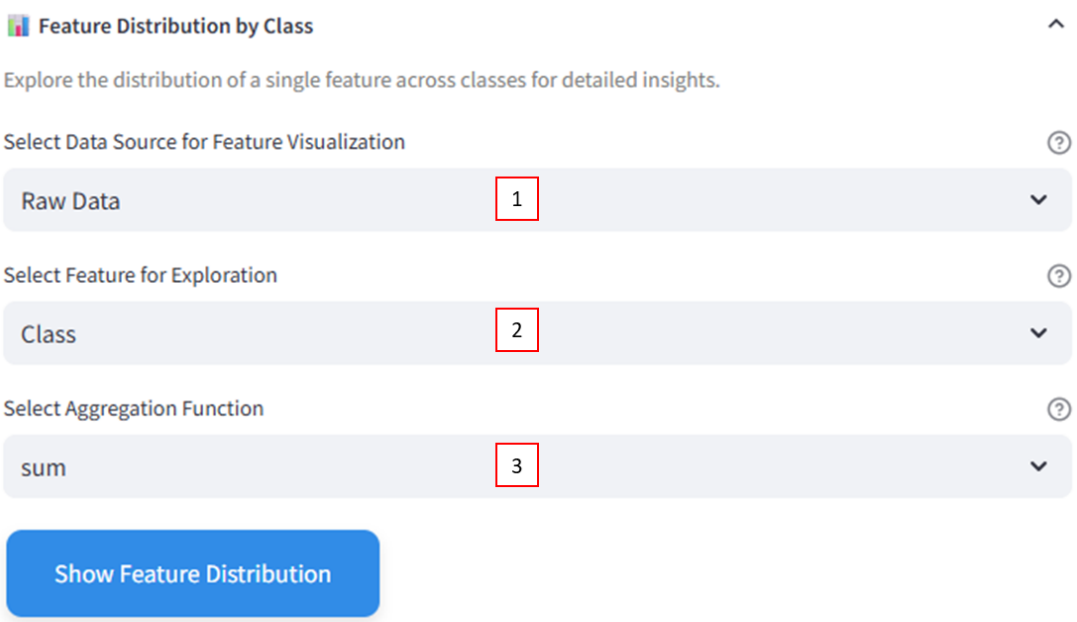
This-sub-category is itself divided into four expanders, which will be explained one by one throughout this tutorial. Its purpose is to generate visualizations such as bar charts of class distributions, average spectra/features and individual spectra/features and Venn diagrams for feature comparisons.

Feature distribution by class

This first expander aims to visualize the features distribution and explore the dataset to gain insights into its characteristics and distribution.

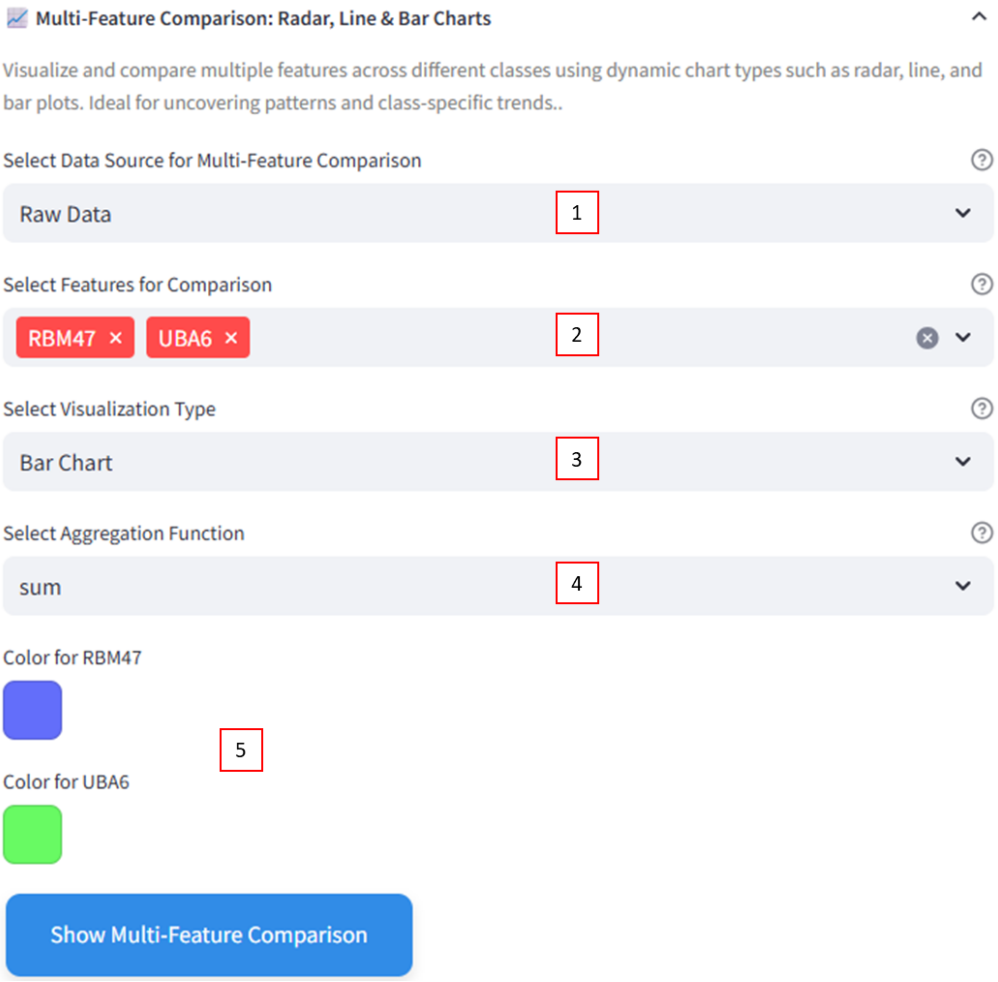
For that, just select a data source (**1**) (either raw data, preprocessed data, oversampled or undersampled data) and also the feature to explore (**2**). A specific feature (either the Class or any features in the dataset) should be picked to have its distribution visualized across different classes.

Finally, define how the values should be aggregated for the feature histogram (e.g., total sum, average, or count per class) (**3**).



Multi-Feature Comparison: Radar, Line & Bar Charts

In addition, multiple features across classes can be compared using various visualization types such as bar, line, and radar charts (**3**), with the aim of gaining deeper insights into the dataset. When the bar chart is selected, colors can be assigned to each feature considered for comparison (**5**).

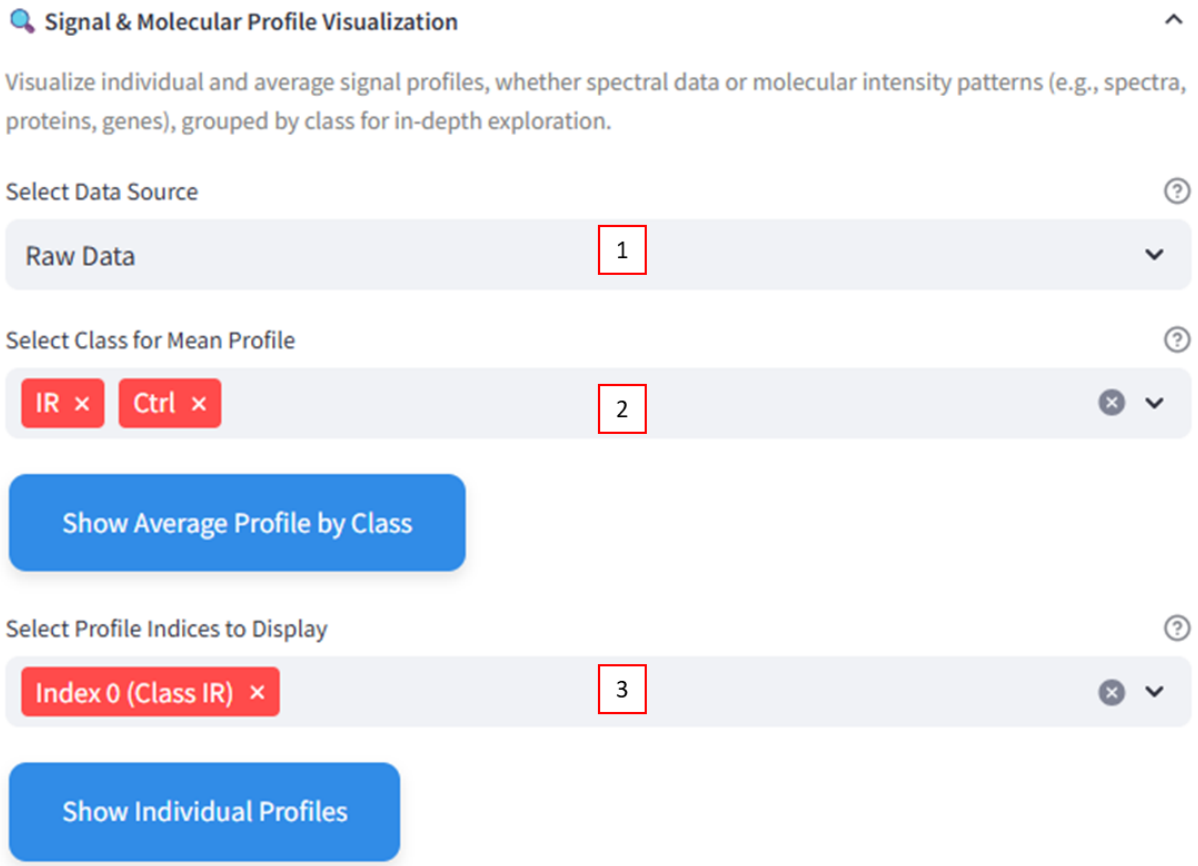


⚠ **Note:** Don't forget to choose the desired features to compare (**2**) and the way how the values should be aggregated (**4**).

⚠ **Note:** As before, don't forget to choose the desired data source (**1**).

Signal & Molecular Profile Visualization

The purpose of this third expander is to display either the average profile (or spectrum) of all samples within a selected class (**2**) or the individual profile (**3**) of a specific sample by choosing its index. Multiple classes can also be selected simultaneously to overlay their average spectra for comparison.

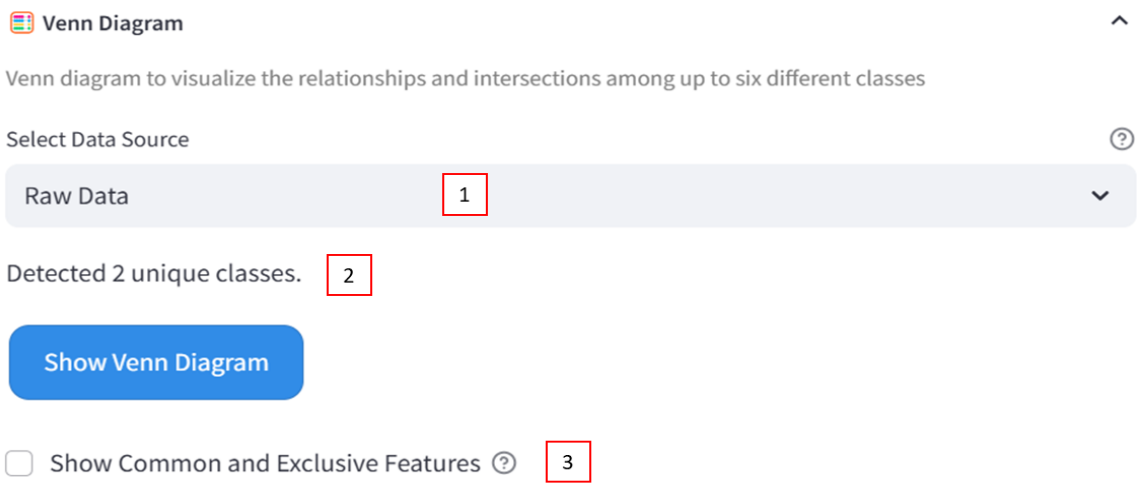


⚠ **Note:** As before, don't forget to choose the desired data source (**1**).

Venn Diagram

The fourth and final expander in this subcategory is designed to visualize the relationships and intersections among up to six different classes.

An informative message (**2**) will appear once the data source is selected (**1**), indicating the number of classes detected in the dataset. By clicking the "Show Venn Diagram" button, the diagram will be displayed. Additionally, by selecting option (**3**), the features that are unique to each class or shared among multiple classes can be visualized.



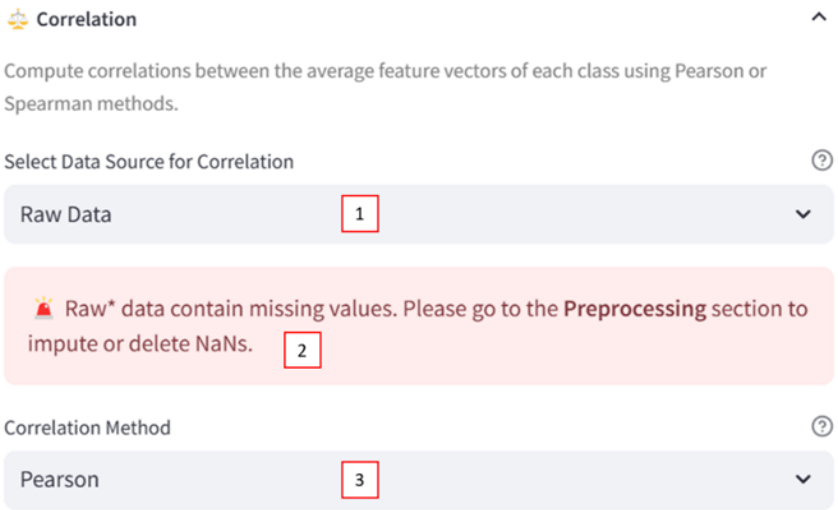
Correlations and Similarities

This-sub-category is itself divided into two expanders, which will be explained individually throughout this tutorial.

Correlation

This first expander is designed to compute correlations between the average feature vectors of each class, using either the Pearson or Spearman method (**3**).

- Pearson correlation should be used when the data is normally distributed. It measures the linear relationship between two continuous variables.
- Spearman correlation is more appropriate when the data is non-parametric or does not follow a normal distribution. It assesses the strength and direction of a monotonic relationship based on rank values.



⚠ **Note**: If the dataset contains missing values, an error message will be displayed, indicating that preprocessing steps such as imputing or removing NaNs are required before proceeding (**2**).

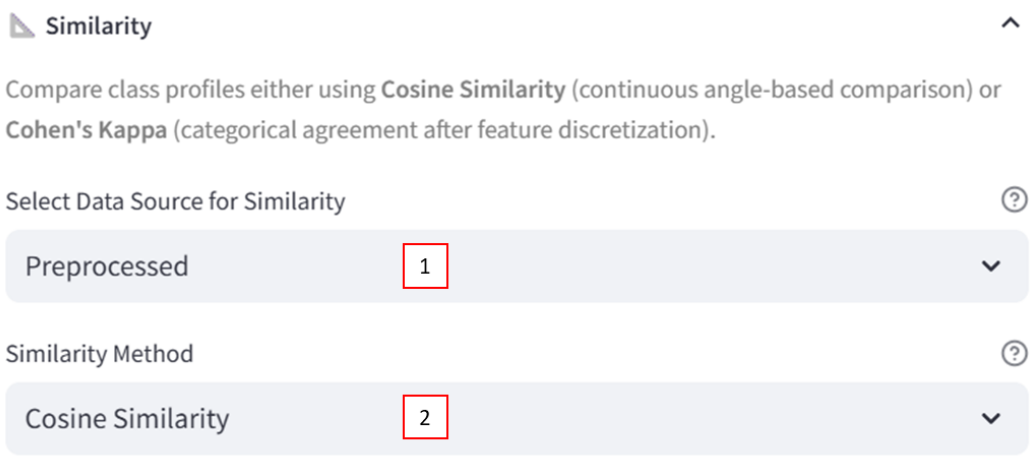
⚠ **Note:** As before, don't forget to choose the desired data source (**1**).

In the result, each cell shows the correlation coefficient between two classes, based on the average of all numeric features.

Similarity

This second expander is designed to compare class profiles either using Cosine similarity (continuous angle-base comparison) or Cohen's Kappa (categorical agreement after feature discretization) (**2**).

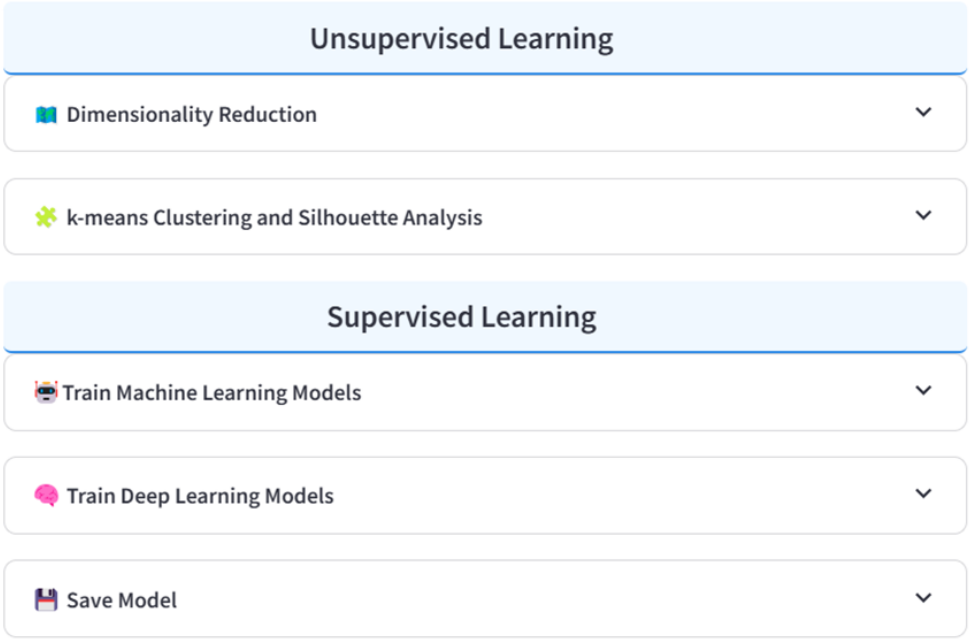
- Cosine similarity measures the angle between feature vectors of each class (1 = identical direction, 0 = orthogonal).
- Cohen's Kappa evaluates the agreement in categorized feature profiles. 1 = perfect agreement, 0 = random, <0 = disagreement. Discretization splits each class feature vector into 3 categories based on intensity ranks, like transforming raw values into 'Low', 'Medium', and 'High' expressions. This allows Kappa to measure agreement on patterns, not exact numbers.



⚠ **Note:** As before, don't forget to choose the desired data source (**1**).

AI Modeling

This third tab is divided into 2 sub-categories: unsupervised and supervised learning.



Unsupervised Learning

This section focuses on dimensionality reduction and clustering techniques to explore and visualize the structure of the dataset in an unsupervised way.

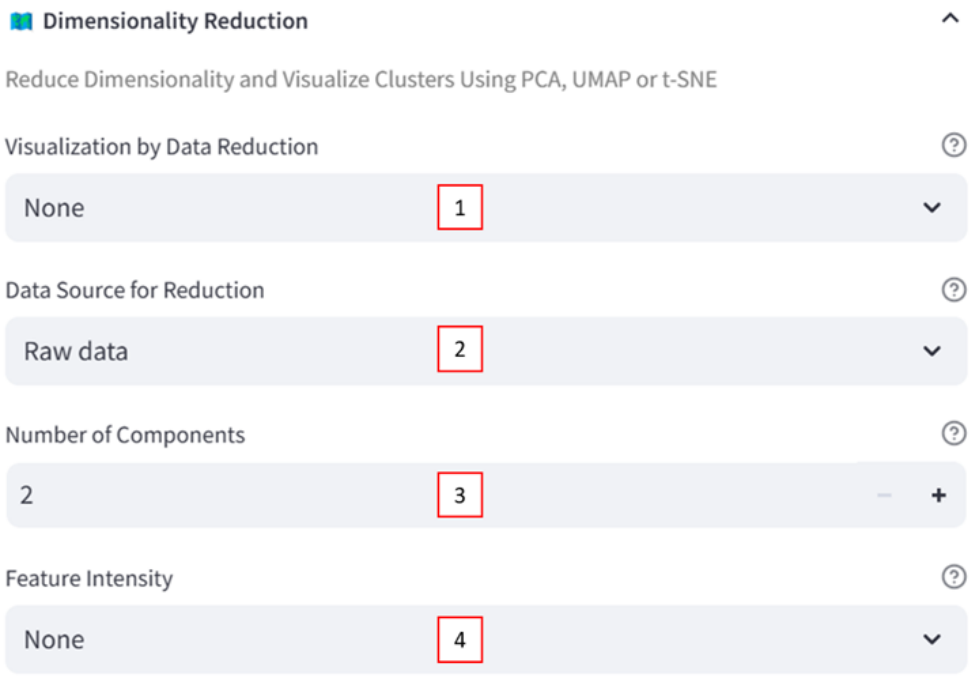
Dimensionality reduction

This first expander is intended for dimensionality reduction and/or cluster visualization using three methods: PCA, UMAP, and t-SNE.

The method can be selected in section (**1**). The appropriate data source must be defined in section (**2**).

The desired number of components is specified in section (**3**). A 2D plot is generated when two components are selected, while three or more components result in a 3D visualization.

A higher number of components generally leads to greater data compression.



For PCA specifically, the explained variance for each component is displayed, along with the contribution of each ion to the corresponding components.

A specific feature can optionally be selected to highlight its intensity in the visualization (**4**).

k-means clustering and Silhouette analysis

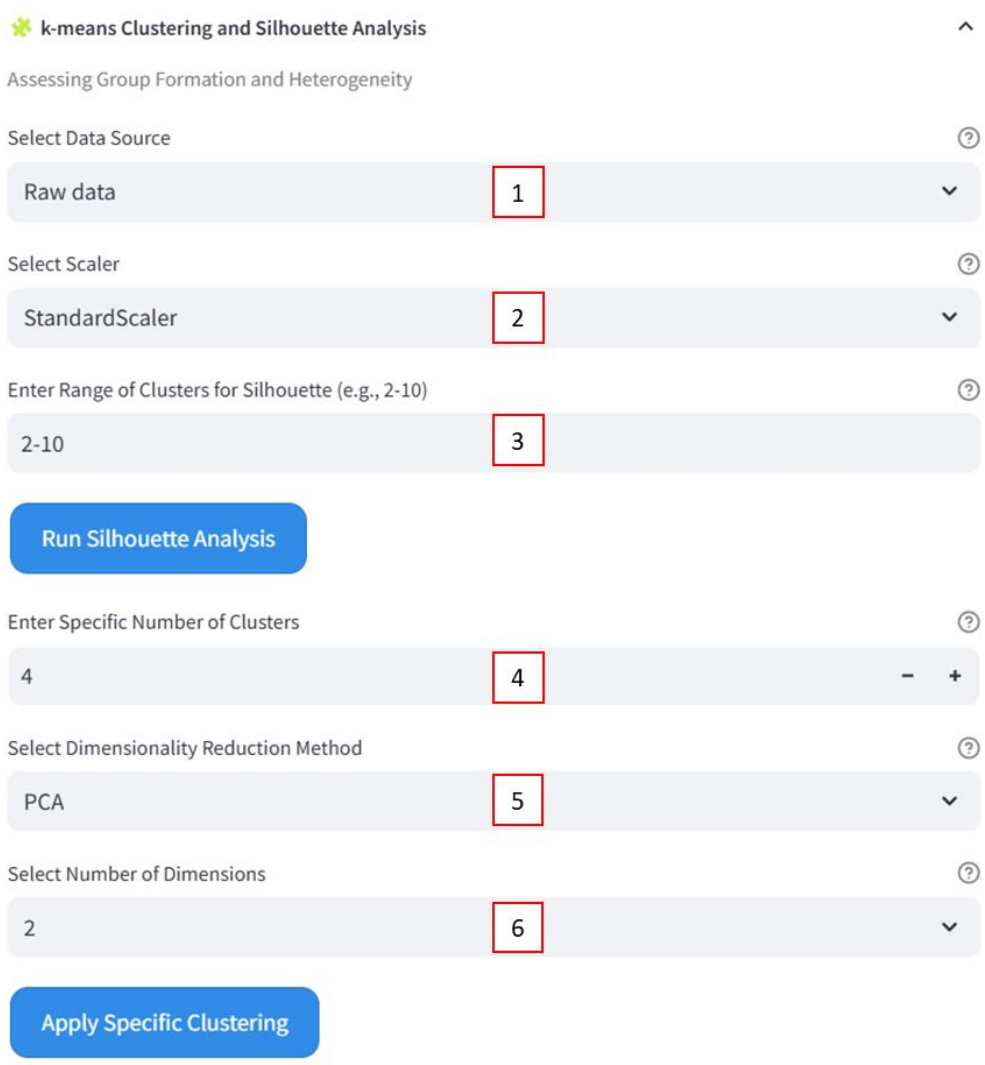
This second expander is intended for assessing group formation and heterogeneity within the dataset.

First, define the appropriate data source in section (**1**).

Next, select a method to normalize the features prior to clustering—options include StandardScaler, RobustScaler, and MinMaxScaler (2).

Note: To assist in the selection of an appropriate clustering normalization method for datasets, a brief explanation of each approach is provided.

- **StandardScaler** is applied to standardize features by removing the mean and scaling to unit variance. This results in a distribution with a mean of 0 and a standard deviation of 1. It is recommended when the data follows a Gaussian (normal) distribution or when a transformation to zero mean and unit variance is desired. However, since it is sensitive to outliers, it may not be appropriate for datasets with significant outliers.
- **RobustScaler** is used to scale features using statistics that are robust to outliers. The median is subtracted, and scaling is performed according to the interquartile range (IQR). This method is suitable when the data contains outliers or when minimizing the impact of outliers on the scaling process is important. It is particularly effective for datasets with skewed distributions or extreme values.
- **MinMaxScaler** is employed to scale features to a specified range, typically between 0 and 1. The data is transformed so that the minimum value becomes 0 and the maximum becomes 1. This approach is suitable when the original distribution of the data needs to be preserved and when features have bounded ranges or when interpretability of scaled values is desired.



In section (**3**), specify the range of cluster numbers to evaluate for silhouette analysis (format: start–end). This analysis helps determine the optimal number of clusters a priori, before performing the actual clustering.

Once the optimal number of clusters has been identified—or if a specific number of clusters is already known—set the exact number to use for final clustering and visualization (**4**).

If necessary, dimensionality reduction can be applied before clustering (**5**).

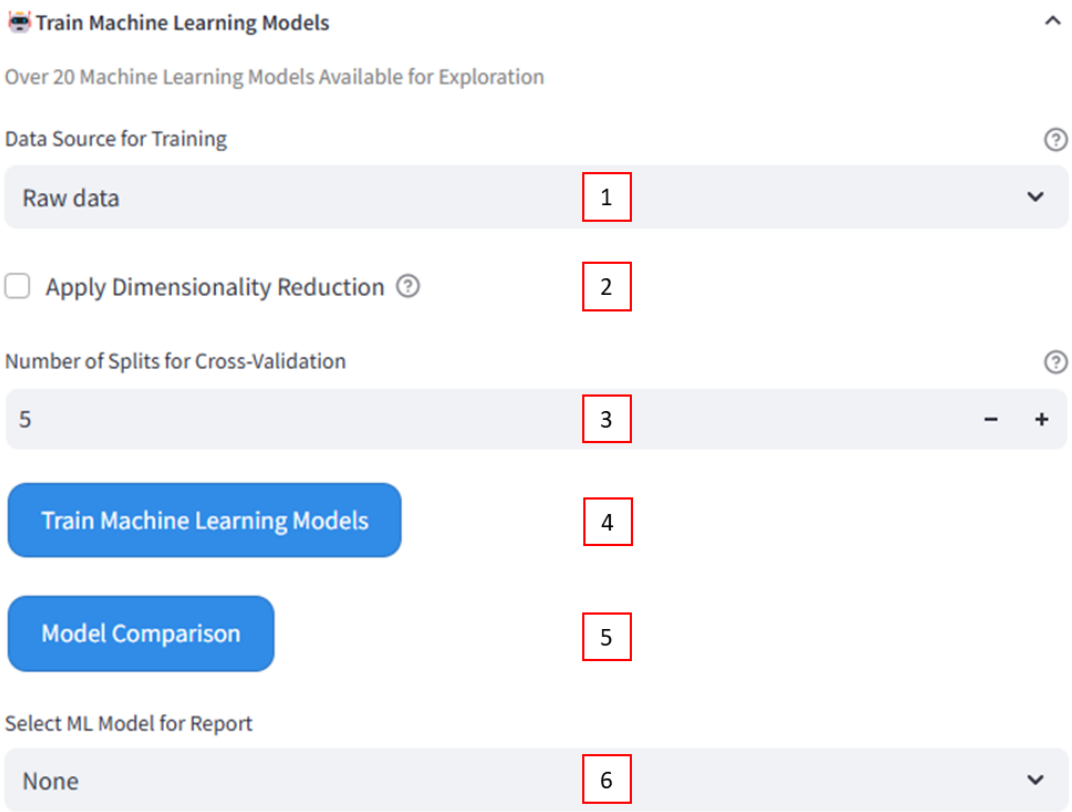
Finally, choose whether to visualize the clustering results in 2D or 3D (**6**).

Supervised Learning

This section focuses on supervised learning to train and save classification models by using either machine learning or deep learning algorithms.

Train Machine Learning Models

This first expander aims to train classification models by using machine learning. Indeed, over 23 algorithms, including linear models, tree-based models, and ensemble methods are trained and compared.



⚠ **Note:** As for all analysis, don't forget to choose the desired data source (**1**).

If needed, a dimensionality reduction can be applied to the data before the training (**2**). Indeed, reducing feature dimensions, by using PCA, UMAP, or t-SNE, faster training and prevent overfitting in case of features>>>sample.

The models are evaluated by using a k-fold cross-validation. Knowing that the number of splits is chosen in section (**3**). Higher values give more reliable generalization estimates but increase

training time. For example, with 5 splits, the model trains on 80% of the data and validates on 20%, rotated across 5 cycles.

After clicking the « Train Machine Learning Models » button, a progress bar will appear, providing feedback on the training stage (**4**).

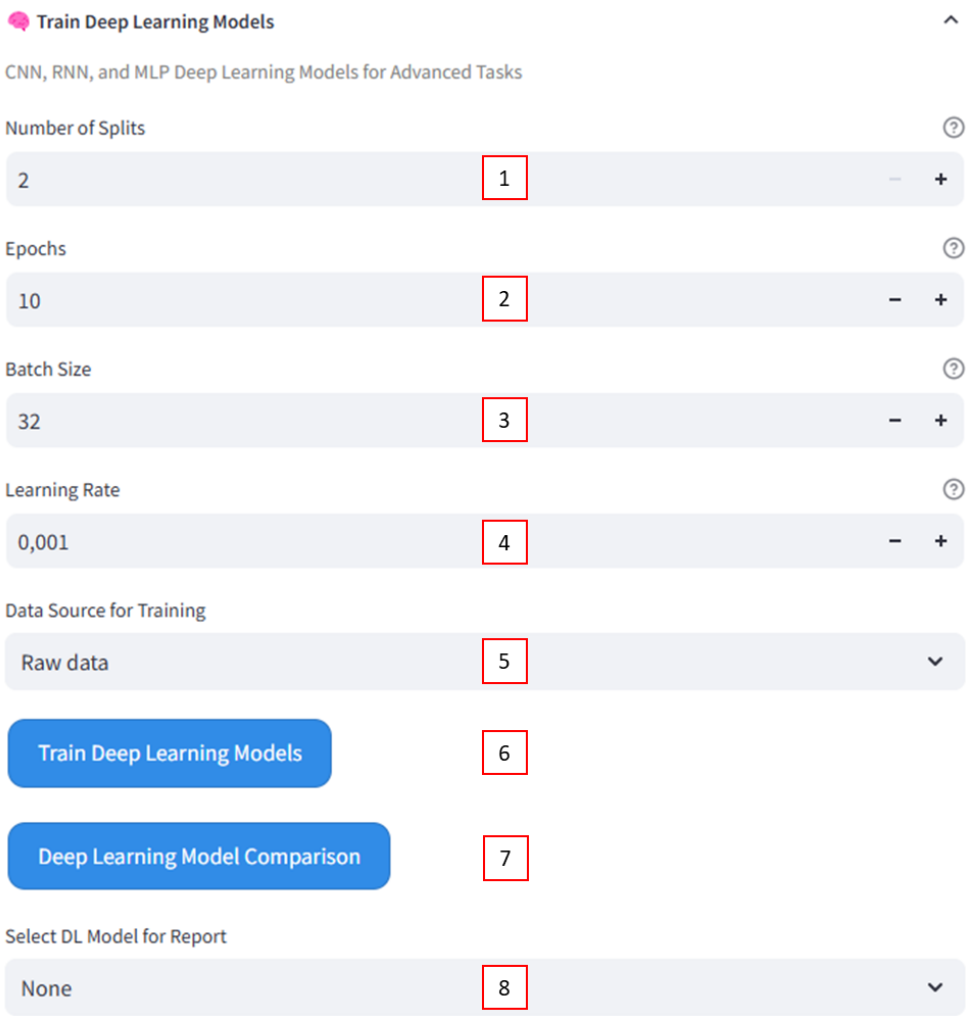
Once training is complete, click the “Model Comparison” button to view the performance of each algorithm (**5**). A bar chart will display the cross-validated accuracy and F1 score for all models, helping to identify the most effective one.

Additionally, the top three models, ranked by F1 score, are presented with detailed metrics: F1 score, accuracy, sensitivity, and specificity.

To view the confusion matrix and classification report for a specific model, simply select it (**6**) and click the corresponding button.

Train Deep Learning Models

As the previous expander, this one aims to train classification models by using, this time, deep learning algorithms. Indeed, 3 algorithms, including CNN, RNN and MLP are trained and compared.



⚠ **Note**: As for all analysis, don't forget to choose the desired data source (**5**).

The models are evaluated by using a k-fold cross-validation. Knowing that the number of splits is chosen in section (**1**). Higher values give more reliable generalization estimates but increase training time.

In contrary to machine learning models, different parameters need to be adjusting before the training, like batch normalization, epochs and learning rate.

The epochs (**2**) correspond to the number of complete passes through the training dataset. Too few may lead to underfitting, too many may cause overfitting. Start with 10–20 and adjust based on performance.

The batch size (**3**) is the number of samples used per gradient update. Smaller batches give noisier but more frequent updates. Larger batches are more stable but require more memory. A batch size of 32 is a common starting point.

For the learning rate (**4**), a lower value makes learning slower but more stable. Try 0.001 as a starting point.

After clicking the « Train Deep Learning Models » button, a progress bar will appear, providing feedback on the training stage (**6**).

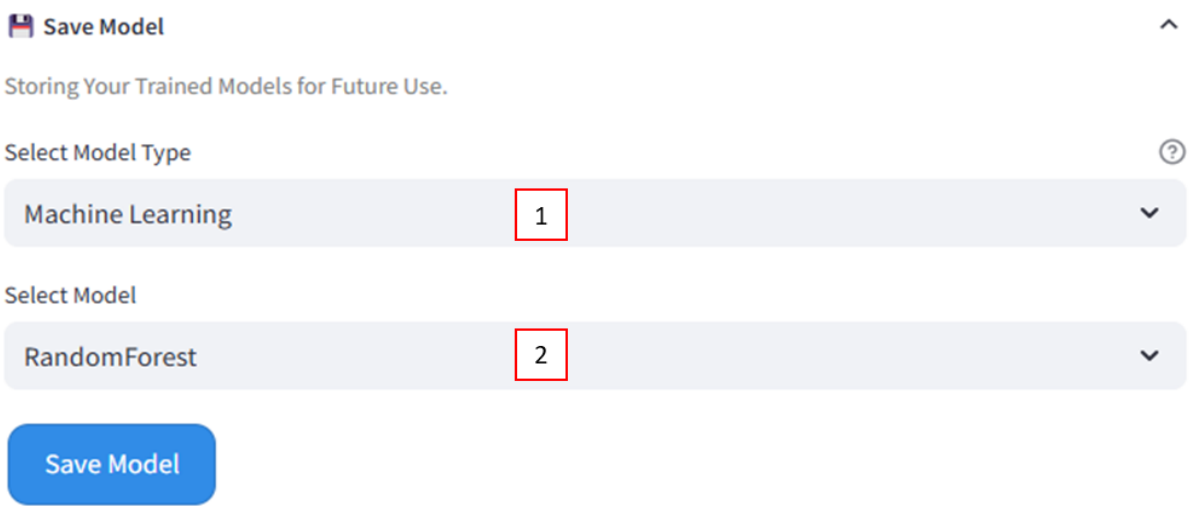
Once training is complete, click the "Deep Learning Model Comparison" button to view the performance of each algorithm (**7**). A bar chart will display the cross-validated accuracy and F1 score for all models, helping to identify the most effective one.

To view the confusion matrix, classification report, training/validation accuracy and loss functions for a specific model, simply select it (**8**) and click the corresponding button.

It will then be possible to retrieve the results of the model with or without cross-validation.

Save Model

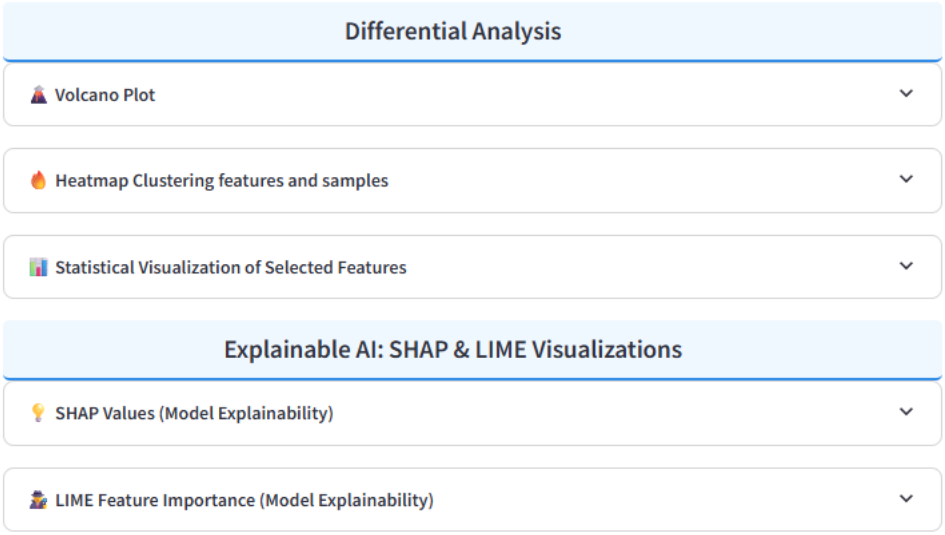
This last expander is designed to simply save trained models and associated feature-label data in .pkl format for future use or real-time applications.



⚠ **Note**: To save a model, simply select the model type, either a machine learning or deep learning model (**1**), and then choose the corresponding algorithm (**2**).

Biomarker Discovery

This fourth tab is divided into 2 sub-categories: statistical hypothesis tests and black box model analysis. This two aims to discover biomarkers, either dependent on the classification models, or independently.



Differential Analysis

This first sub-category includes three expanders, each offering a method for biomarker discovery independent of classification models: volcano plot analysis, heatmap clustering, and statistical testing.

Volcano Plot

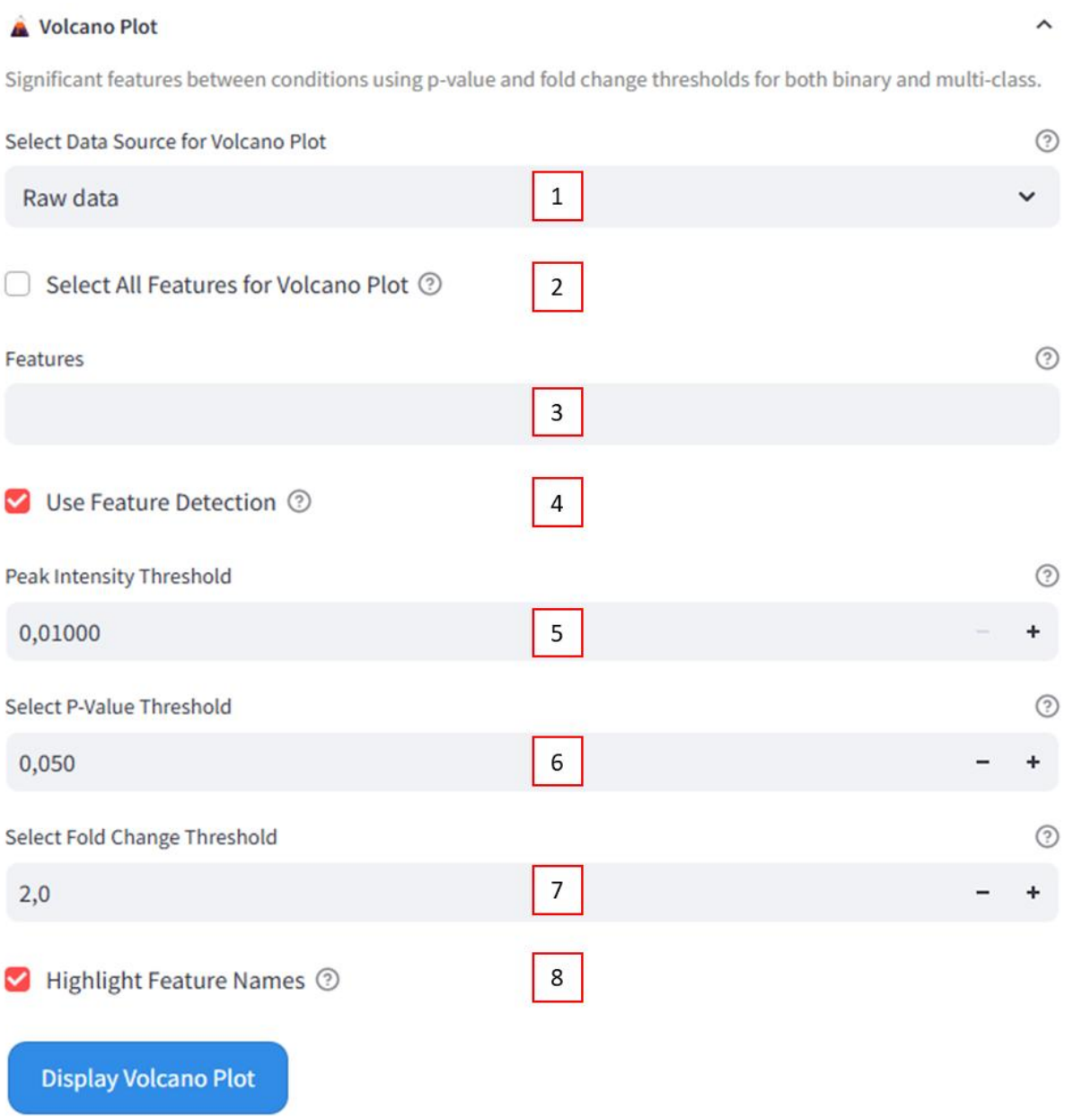
This first expander, using volcano plot, aims to discover significant features between conditions using p-value and fold change thresholds for either binary or multi class.

As for all analysis, the desired data source needs to be chosen (**1**).

The analysis can be done on all the features in the dataset (**2**) or on selected features (**3**). It is possible to enable feature detection to automatically identify peaks of interest based on an intensity threshold (**4-5**). If wanted, the intensity threshold for peak detection needs to be set (**6**).

⚠ **Note**: Don't forget to set the p-value and fold change threshold to filter significant features (**7-8**).

Finally, check the last box to highlight feature names in the Volcano Plot (**8**).



Heatmap Clustering features and samples

The second expander, using heatmap clustering, aims to cluster feature and samples with statistical significance tested on intensities.

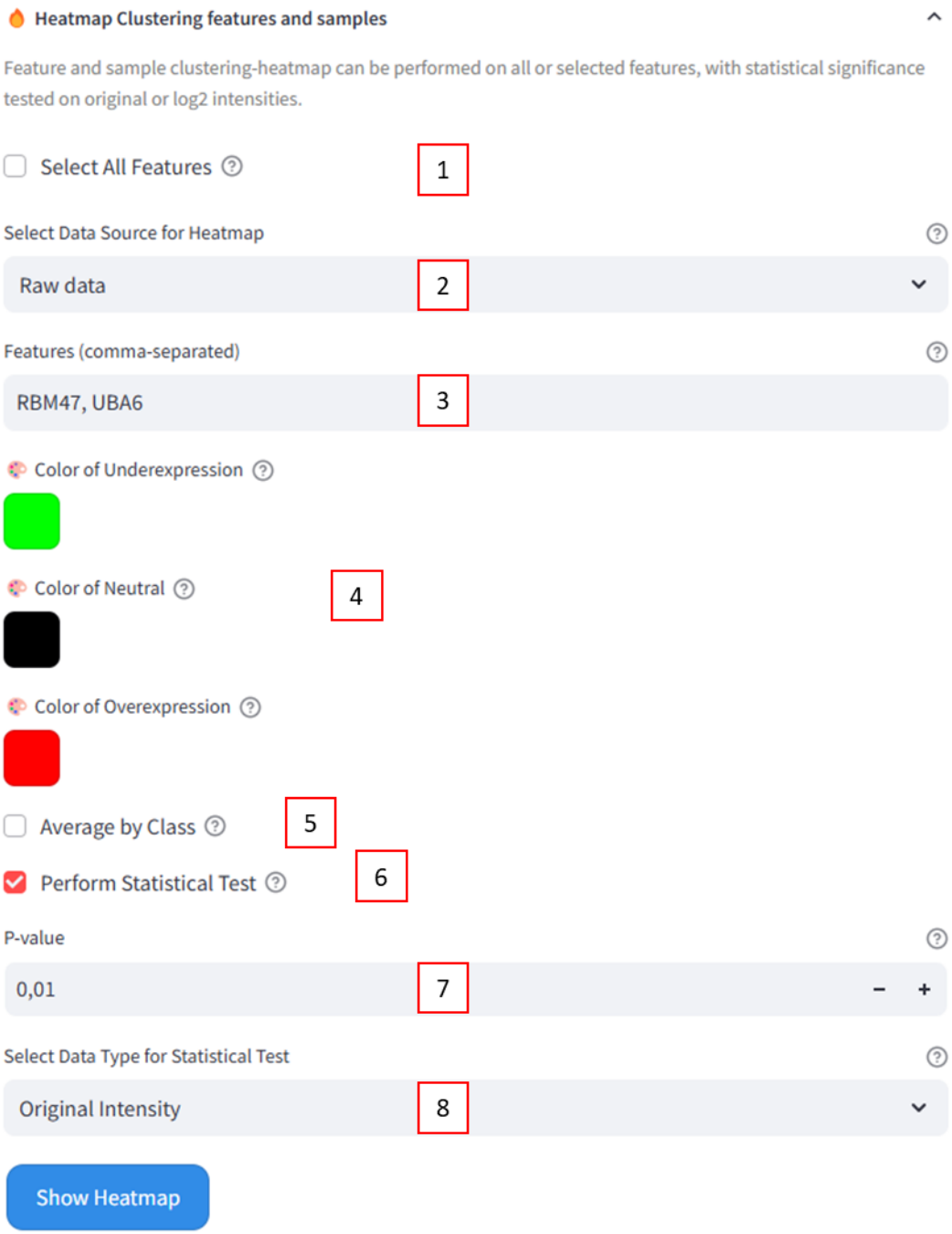
As for all analysis, the desired data source needs to be chosen (**2**).

The analysis can be done on all the features in the dataset (**1**) or on selected features (**3**).

The colors used for under expression, neutral expression and overexpression can be changed if needed (**4**).

Check the “Average by class” box to average the feature values by class in the Heatmap (**5**).

It is possible to perform a statistical test on the selected features (**6**). If wanted, the p-value threshold for statistical test needs to be set (**7**). In addition, choose the data type for the statistical test (original or transformed log2 (fold-change)) (**8**).



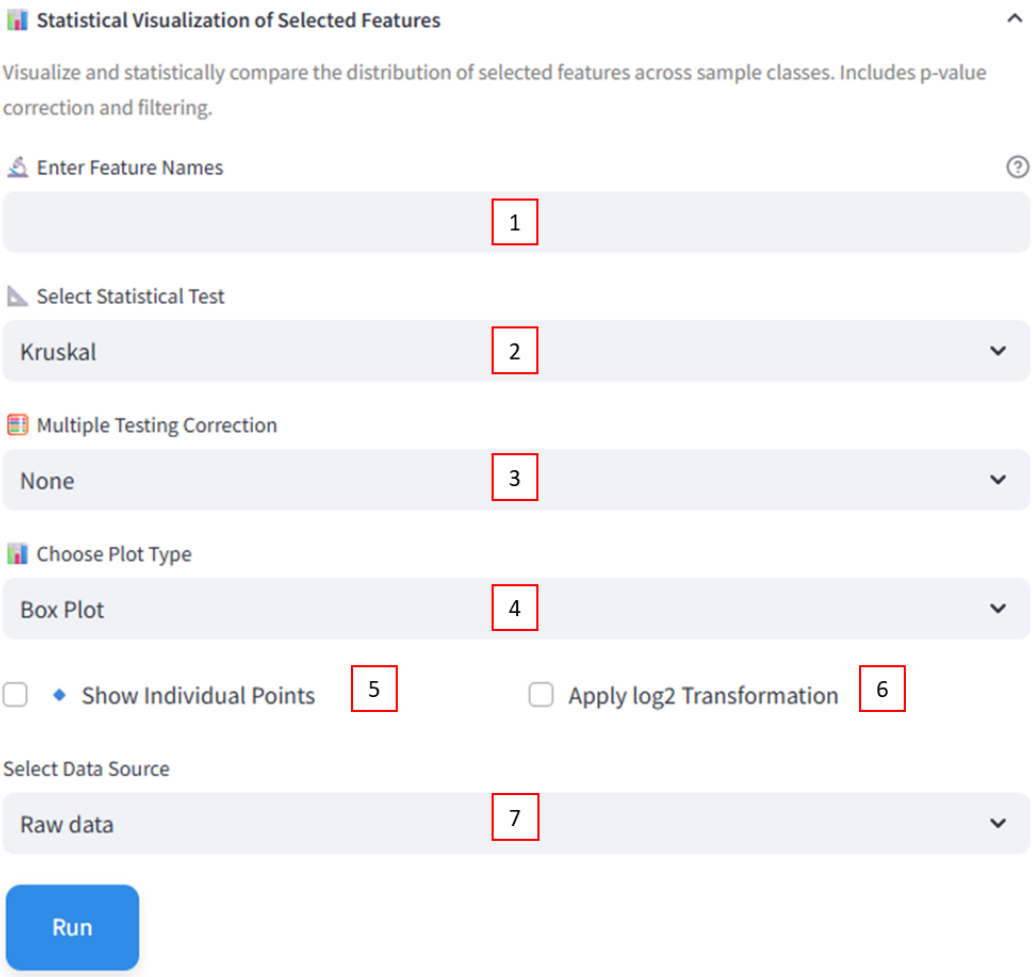
Statistical Visualization of Selected Features

This third expander allows to display boxplots, violin plots, or bar plots for selected features, to assess their statistical significance.

To use it, simply enter a comma-separated list of features to visualize (**1**), and choose the appropriate statistical test to apply (**2**): Kruskal-Wallis, Mann-Whitney, independent t-test, or ANOVA.

Note : The choice of statistical test depends on the number of groups being compared and the distribution of the data. Profiler suggests appropriate tests based on these factors.
In general :
- **Mann-Whitney test** : Used for comparing two groups when the data are not normally distributed.
- **Independent t-test** : Used for comparing two groups when the data are normally distributed and have equal variances.
- **Kruskal-Wallis test** : Suitable for comparing three or more groups when the data are not normally distributed.
- **ANOVA** : Used for comparing three or more groups when the data are normally distributed.
When the normality of the data is uncertain, non-parametric tests such as Mann-Whitney test or Kruskal-Wallis tests are safer choices, though they tend to be less powerful than their parametric counterparts.

When dealing with multi-class rather than binary data, it is recommended to apply a multiple testing correction, such as Bonferroni or False Discovery Rate (Benjamini-Hochberg), depending on the characteristics of the dataset (**3**).



You can optionally check the box (**5**) to overlay a scatter plot on the boxplot or violin plot. In addition, select the desired plot type: boxplot, violin plot, or bar plot (**4**).

Optionally, check the box (**6**) to use log2 transformed values for the plots.

⚠ **Note**: As for all analysis, don't forget to choose the desired data source (**7**).

Black Box Model Analysis

This second sub-category includes two expanders, each offering a method for biomarker discovery dependent of classification models: SHAP values and LIME feature importance.

SHAP Values

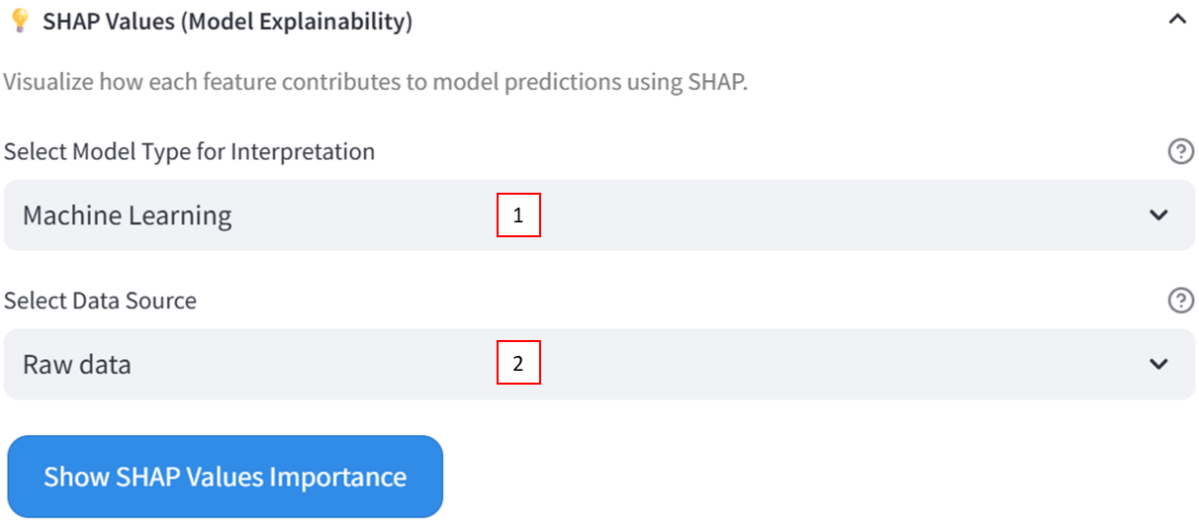
In this first expander, the contribution of each dataset feature to the model's predictions is visualized using SHAP.

The type of model to interpret, machine learning or deep learning, must first be selected (**1**).

⚠ **Note**: For now, biomarker discovery, using SHAP and LIME, is only available for machine learning models.

The interpretation will be based on the model previously selected in the 'Train machine learning *models'* section.

⚠ **Note**: As with all analyses, the appropriate data source must be selected (**2**). It must be the same as the one used during model training.



However, SHAP cannot explain the predictions of all algorithms.

The table below summarizes which algorithms are supported and which are not.

Yes (13)	No (10)
Decision Tree	AdaBoost
Extra Tree	Bagging
Extra Trees	Dummy
Gradient Boosting	KNeighbors
Hist Gradient Boosting	Linear SVC
Linear Discriminant Analysis	NaiveBayes_Gaussian
LGBM	NaivesBayes_Bernoulli
Logistic Regression	Nearest Centroid
Passive Aggressive	Quadratic Discriminant Analysis

Perceptron Random Forest Ridge Classifier SGD	SVC

LIME Feature Importance

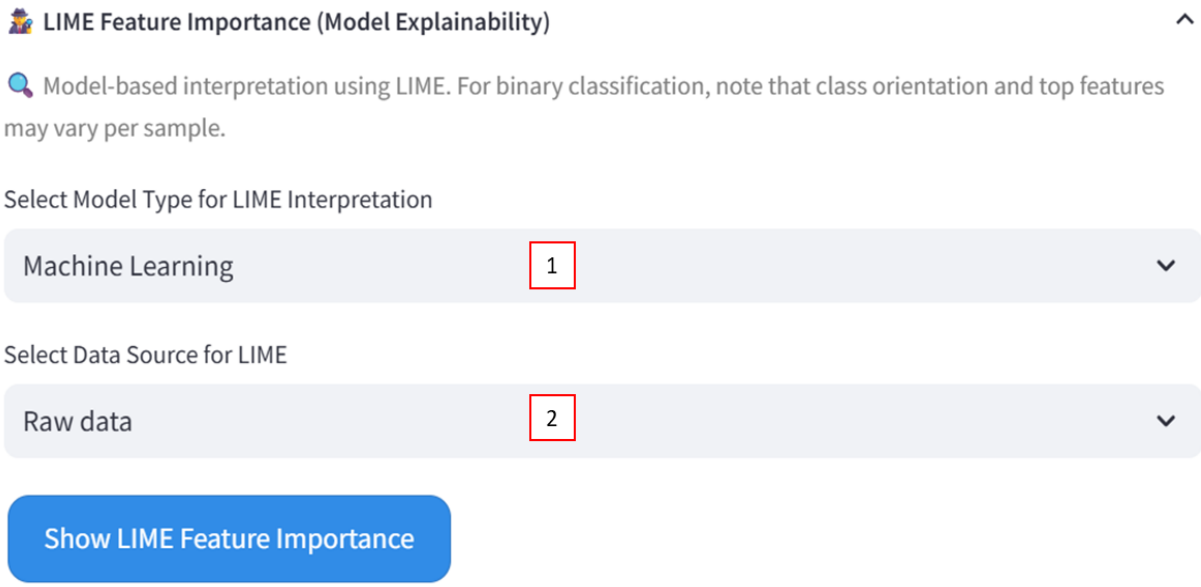
This second expander provides a LIME-based interpretation of how each feature in the dataset contributes to the model's predictions.

Begin by selecting the type of model to interpret, either machine learning or deep learning (**1**).

⚠ **Note**: For now, biomarker discovery, using SHAP and LIME, is only available for machine learning models.

The interpretation will rely on the model previously chosen in the 'Train machine learning models' section.

⚠ **Note**: As for all analyses, be sure to select the correct data source (**2**). It must correspond to the one used during model training.



Keep in mind that LIME does not support all algorithms.

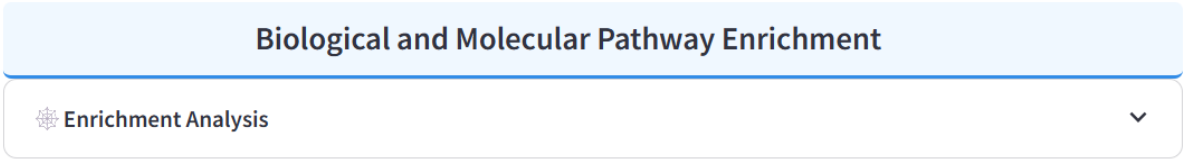
The table below outlines which algorithms are compatible with LIME and which are not.

Yes (14)	No (9)
AdaBoost	Bagging
Decision Tree	Dummy
Extra Tree	Hist Gradient Boosting
Extra Trees	KNeighbors
Gradient Boosting	Linear Discriminant Analysis
Linear SVC	Quadratic Discriminant Analysis
LGBM	NaiveBayes_Gaussian
Logistic Regression	NaivesBayes_Bernoulli

Passive Aggressive Perceptron Random Forest Ridge Classifier SGD SVC	Nearest Centroid

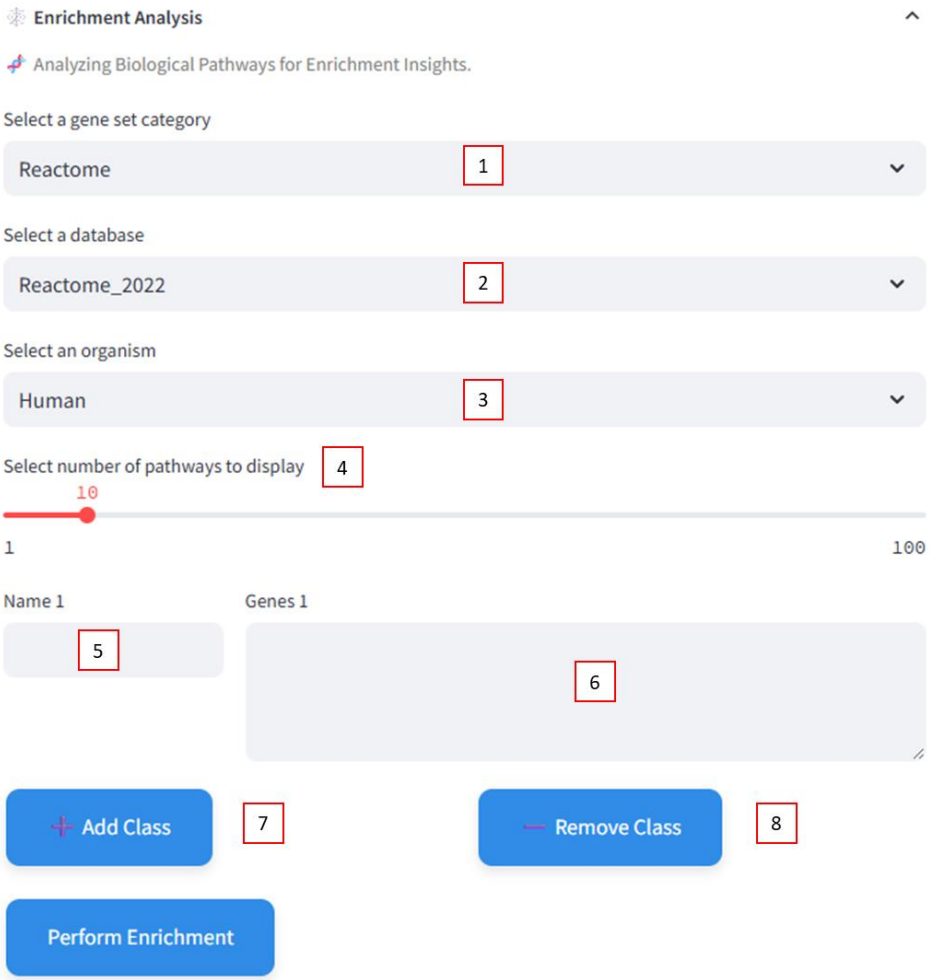
Enrichment

This fifth tab is only composed of one expander for the enrichment analysis.



Enrichment analysis

The goal, here, is to analyze the biological pathways for enrichment insights.



Indeed, just choose a category of gene sets to analyze (KEGG, GO, Reactome, ARCHS4, Drug, MSigDB and Other) (**1**).

From the selected category, choose a specific gene set database (mostly the year of the database) (**2**).

Additionally, select the organism for which the enrichment analysis will be performed, either Human, Mouse, Rat, Yeast, Fly, Worm or Fish (**3**).

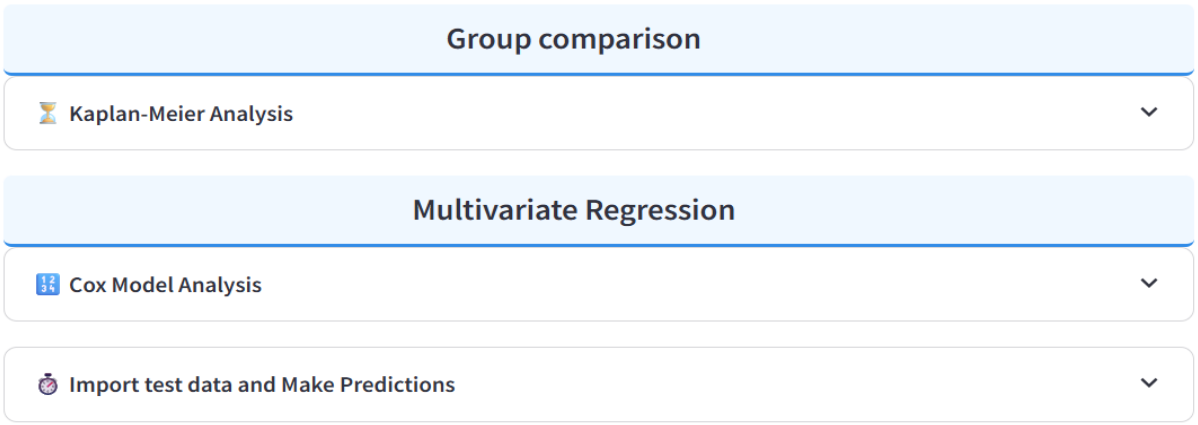
The number of pathways statistical enriched to display can be chosen in the section (**4**).

Finally, genes of interest (comma-separated) should be entered in section (**6**) and the respective conditions of interest in section (**5**).

⚠ **Note**: Enrichment analysis can be performed on multiple gene groups. To achieve this, classes can be added (**6**) or removed (**7**), and a name should be assigned to each group.

Survival Analysis

This sixth tab is divided into 2 sub-categories: group comparison and multivariate regression.



Group Comparison

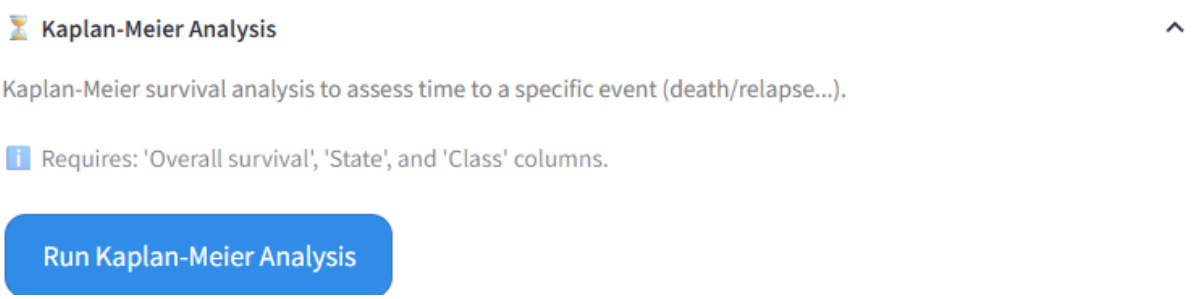
Kaplan-Meier Analysis

The Kaplan-Meier curve is a statistical method used to estimate the probability of survival over time, considering the time until an event occurs (such as death, relapse, or recovery).

To perform this analysis, the dataset must include a column for "Overall Survival", a "State" column (indicating whether the event of interest occurred) and a "Class" column (groups).

This method is particularly useful for comparing survival between two or more groups of patients (e.g., younger vs. older individuals, treated vs. untreated). It also allows for the estimation of key survival metrics, such as the median survival time.

To assess whether survival differences between groups are statistically significant, a log-rank test is typically performed.



Multivariate Regression

Cox Model Analysis

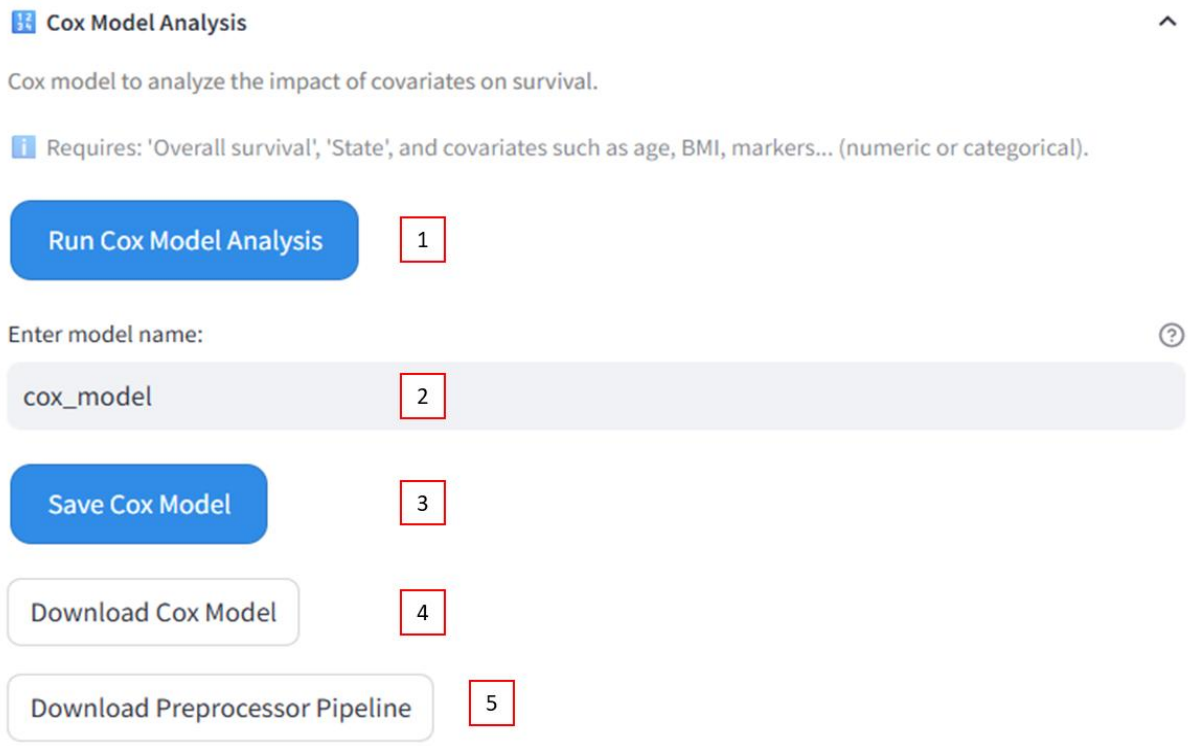
A Cox analysis (or Cox proportional hazards regression model) is used to evaluate the impact of multiple variables (called covariates) on survival time or the time to a specific event (such as death, relapse, or recovery).

Unlike the Kaplan-Meier curve, which compares survival between groups based on a single variable, the Cox model allows for simultaneous adjustment for multiple factors (e.g., age, sex, comorbidities). This makes it possible to isolate the independent effect of each variable on the outcome.

The model helps to identify factors associated with survival and to quantify their impact. For each covariate, the Cox model estimates a hazard ratio (HR):

- An HR > 1 indicates an increased risk of the event.
- An HR < 1 indicates a reduced risk.
- For example, HR = 2 means the risk is twice as high for individuals with that characteristic, compared to the reference group.

⚠ **Note**: To perform this analysis (**1**), the dataset must include a column for "Overall Survival", a "State" column (indicating whether the event of interest occurred) and covariates columns such as age, BMI, markers (either numerical or categorical).



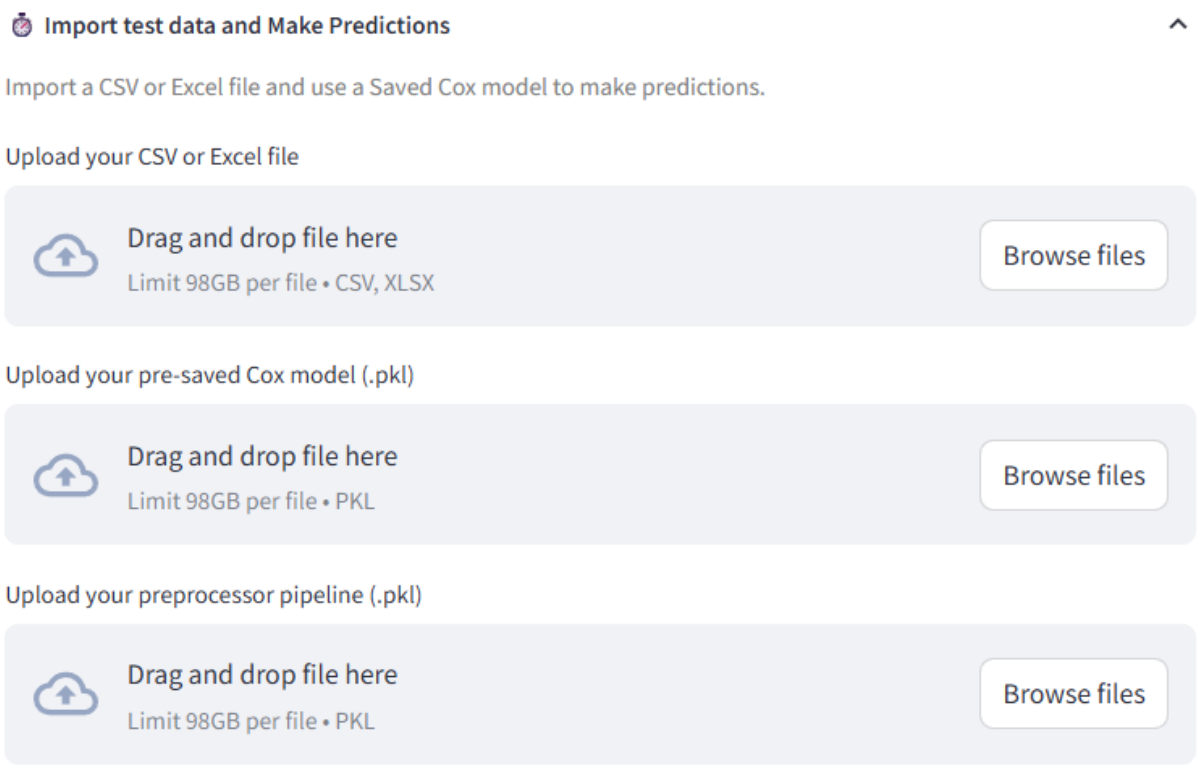
Finally, the Cox model can be saved by clicking the “Save Cox Model” button (**3**) after the desired model name is entered (**2**). Two buttons will then appear, allowing the Cox model and the corresponding preprocessing pipeline to be downloaded (**4**-**5**).

Import test data and Make Predictions

This last expander is designed to make predictions on a new dataset using a previously saved Cox model.

To do so, simply upload:

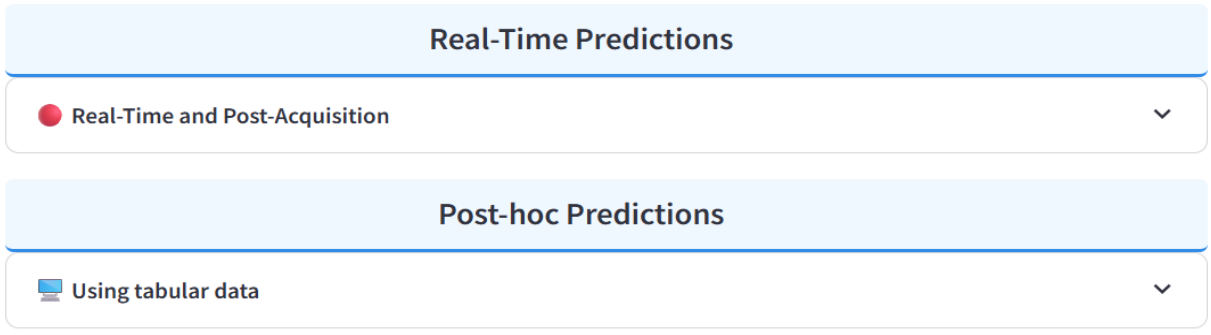
- a CSV or Excel file containing the new dataset,
- the previously saved Cox model (.pkl), and
- the corresponding preprocessing pipeline (.pkl), which was saved at the same time as the model.



The output consists of the overall survival prediction and descriptive statistics for the new patient, based on the previously trained Cox model.

Wizard

This seventh tab is divided into 2 sub-categories: Real-time predictions and post-hoc predictions.



Real-Time Predictions

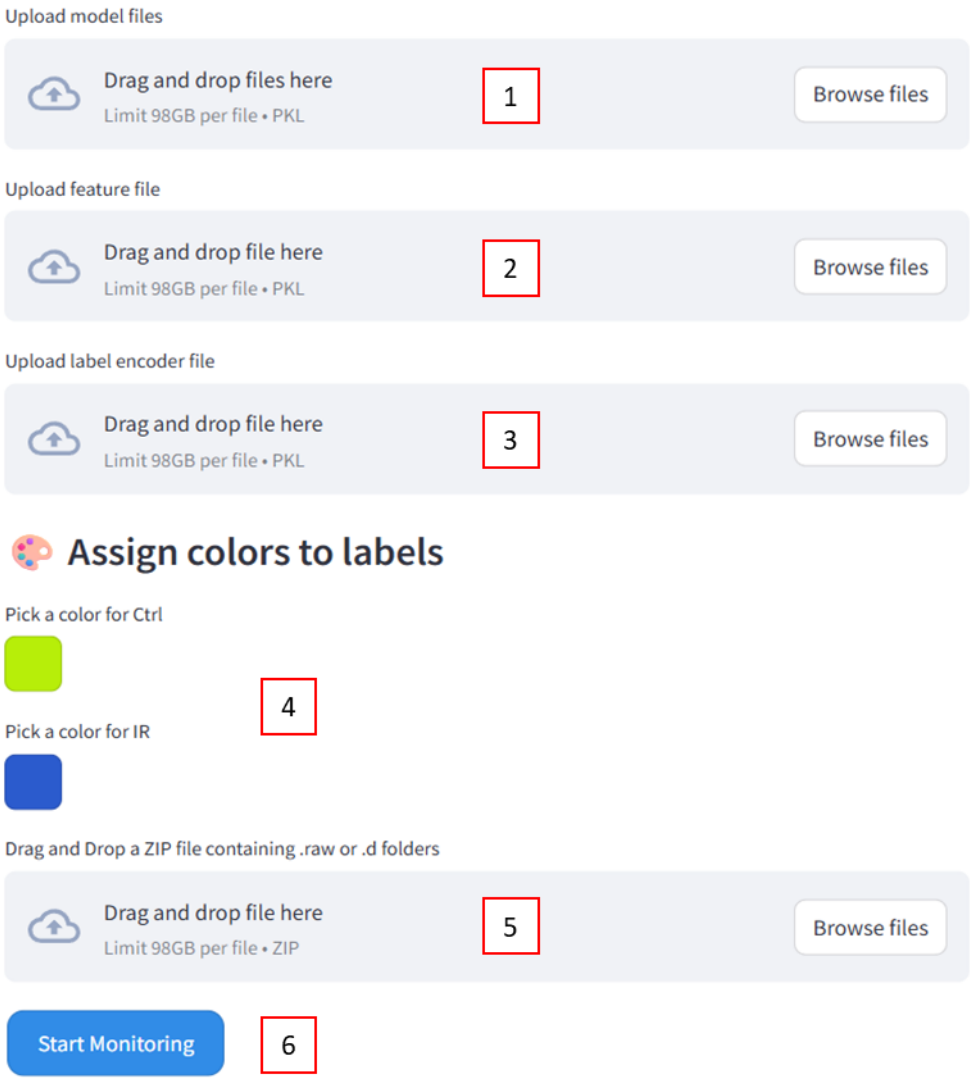
⚠ **Note**: When using Profiler via the web interface, real-time prediction directly from the instrument is not supported. However, it is possible to drag and drop a Raw file from Waters, Bruker, or Thermo to generate predictions. If real-time prediction is wanted, contact the author to obtain local version of Profiler.

Real-Time and Post-Acquisition

For post-acquisition predictions, zipped raw files are used (**5**). Prior to this, the following elements must be uploaded:

- the previously trained model (**1**),
- the corresponding feature set (**2**), and
- the associated label encoder (**3**), saved at the same time as the model.

Appropriate colors should be assigned to each label as desired (**4**).



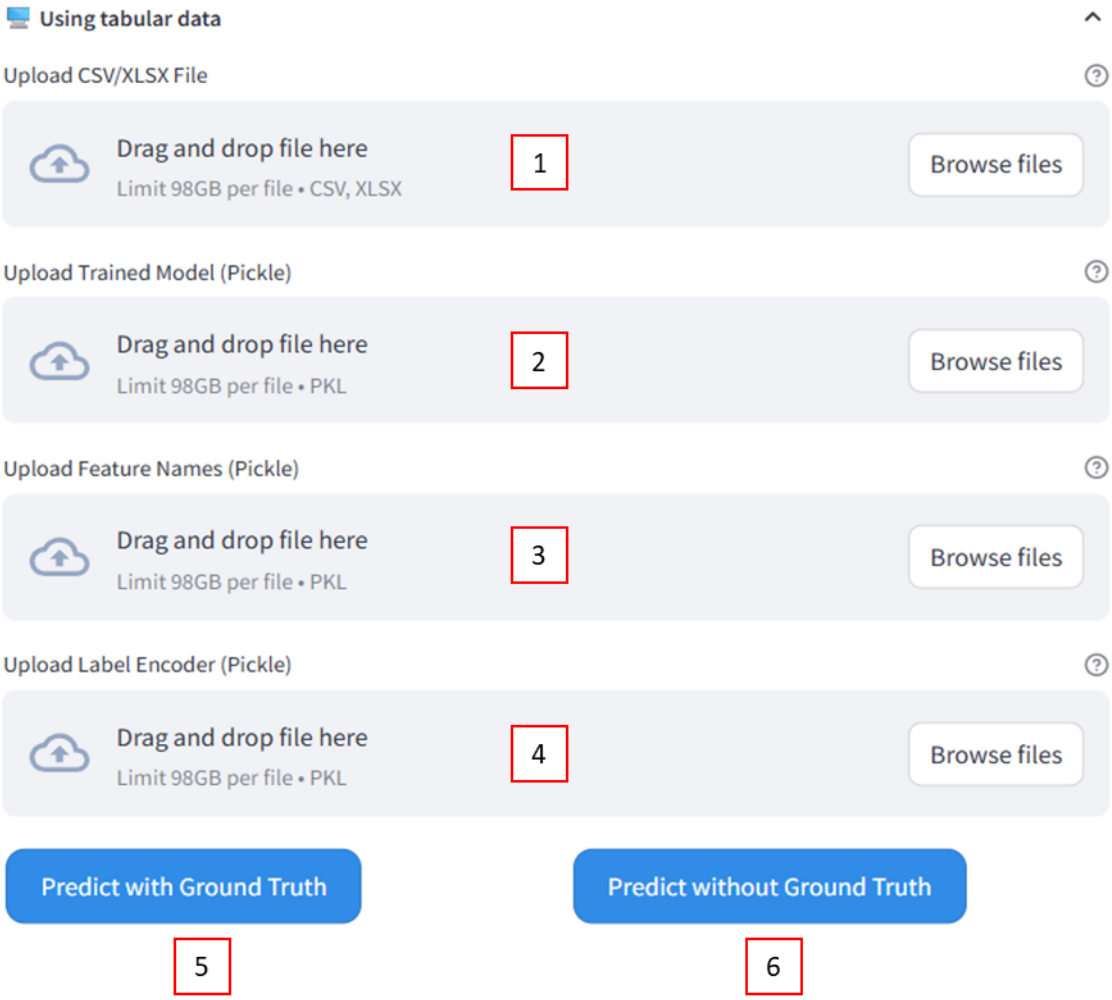
Finally, click the "Start Monitoring" button (**6**) to generate predictions on the new dataset using the previously trained model.

Post-hoc Predictions

Using tabular data

For post-hoc predictions, CSV/XLSX files are used (**1**). Prior to this, the following elements must be uploaded:

- the previously trained model (**2**),
- the corresponding feature set (**3**), and
- the associated label encoder (**4**), saved at the same time as the model.



Finally, two options are available to generate predictions on the new dataset using the previously trained model:

- Click “Predict with Ground Truth” (**5**) when the true outcomes are known and a comparison is desired. In addition to the prediction results displayed in the table, a confusion matrix and a classification report will be generated.

- Click “Predict without Ground Truth” (**6**) when the true outcomes are not available.