Engineering a metabolomics workflow to capture single-cell diversity in 3D models

Hetzel, L.A.

Chapter II

From Mythology to Methodology: Untargeted single-cell metabolomics in R with MeDUSA

Based on:

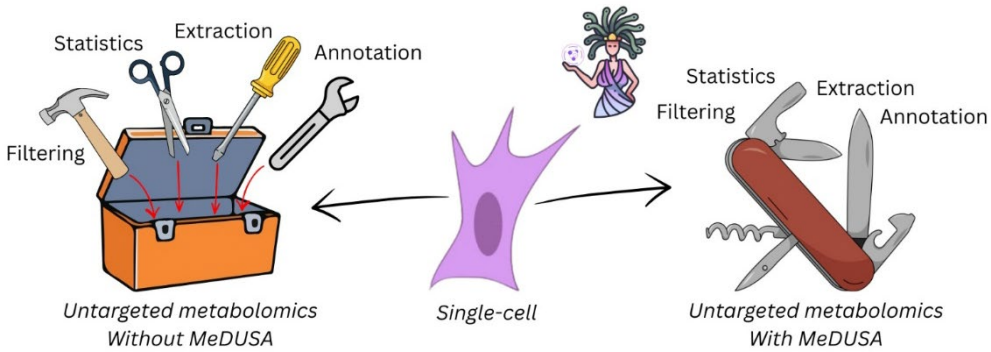
Laura Ann Castaneda, Eric Hetzel, Fercan Gül, Thomas Hankemeier, Ahmed Ali

From Mythology to Methodology: Untargeted single-cell metabolomics in R with MeDUSA

Journal of the American Society for Mass Spectrometry (2026)

ABSTRACT

Direct infusion-based single-cell metabolomics analysis has the potential to isolate the causes of drug resistance and cancer progression, however, processing and analyzing the data generated remains a challenge. While many packages exist for metabolomics analysis, they are not optimized for direct infusion-based single-cell measurements which do not rely on chromatographic separation and is typically noisier than traditional population-level methods. To address this gap, the MeDUSA (Metabolomics of direct-infusion untargeted single-cell analysis) R package was developed. MeDUSA was built especially for direct infusion-based single-cell metabolomics, with modularity, noise filtering, and user-customization in mind. In this work we introduce the package, how to use it, and implement it in a single-cell metabolomics experiment to identify the differences between two cell lines. MeDUSA compromises several functions that deal with file import, peak picking, spectral processing, statistical analysis and feature annotation. Each function is defined with the purpose, usage, parameters, default values, and output of an example dataset. MeDUSA was built to be a modular platform that aims to be a foundation to be built upon with additional modules for the single-cell metabolomics field.



Introduction

Untargeted single-cell metabolomics is the only technique capable of evaluating cellular heterogeneity among a population of cells which has an impact on cancer progression[1] and drug resistance. It has been used to identify previously unknown subtypes of cancer[2–4], and to study rare cells such as circulating tumor cells (CTCs).[5] However, despite their utility, untargeted single-cell metabolomic studies require extensive data processing and analysis before any meaningful data is obtained which limits its potential.

The workflow of untargeted single cell metabolomics begins with the measurement of a single cell with mass spectrometry after its isolation using techniques such as live single-cell sampling, cell sorting, or mass spectrometry imaging.[6] Once the measurements are complete, the data is processed for noise reduction, peak detection, and normalization before statistical analysis.[7,8,9] A significant number of resources are available and being developed for metabolomics in general. Unfortunately, aside from mass spectrometry imaging workflows, current vendor software such as Compound Discoverer, and SCIEX OS are not suitable for direct infusion-based single cell untargeted metabolomics. This is due to the fact that they are optimized for separation-based mass spectrometry techniques.[10] Furthermore, they rely on vendor specific formats, which fragments the metabolomics community and makes it difficult to establish community-wide data processing pipelines. Open access tools that are recently developed such as MS-DIAL [11]and XC-MS[12] are mostly optimized for liquid chromatography mass spectrometry (LC-MS) experiments. Therefore, the absence of meaningful retention time in untargeted single-cell direct infusion metabolomics experiments would heavily influence the results.[13] Additionally, most new tools focus on one aspect of the full pipeline, such as annotation or statistical analysis, instead of providing a complete workflow.[13] For example, MZmine[14] does not offer filtering that is suitable for the high missingness inherent to single-cell direct infusion measurements. While other software such as ALEX, is conducive to one aspect of the processing pipeline, such as annotation.[15] To address these limitations, pipelines such as MetaboAnalyst were developed. MetaboAnalyst is one of the most widely used data processing pipelines due to its complete workflow and visualization techniques, however, it is not

designed with untargeted single-cell experiments in mind. This is especially evident when it comes to the high degree of missingness and noise present in single-cell datasets that is incompatible with current techniques used in pipelines such as MetaboAnalyst. This in turn can influence the overall analysis and annotation, and consequently, the biological interpretation of single-cell experiments significantly.[16,17]

To address the forementioned limitations, we have developed an open-source package in R developed specifically for inhomogeneous direct infusion single-cell untargeted metabolomics (MeDUSA). It was created to be an end-to-end, modular solution with user defined parameters to provide a complete workflow while still allowing users to customize the analysis depending on their unique experimental needs. MeDUSA deals with single-cell metabolomics datasets from mzML files, ending with

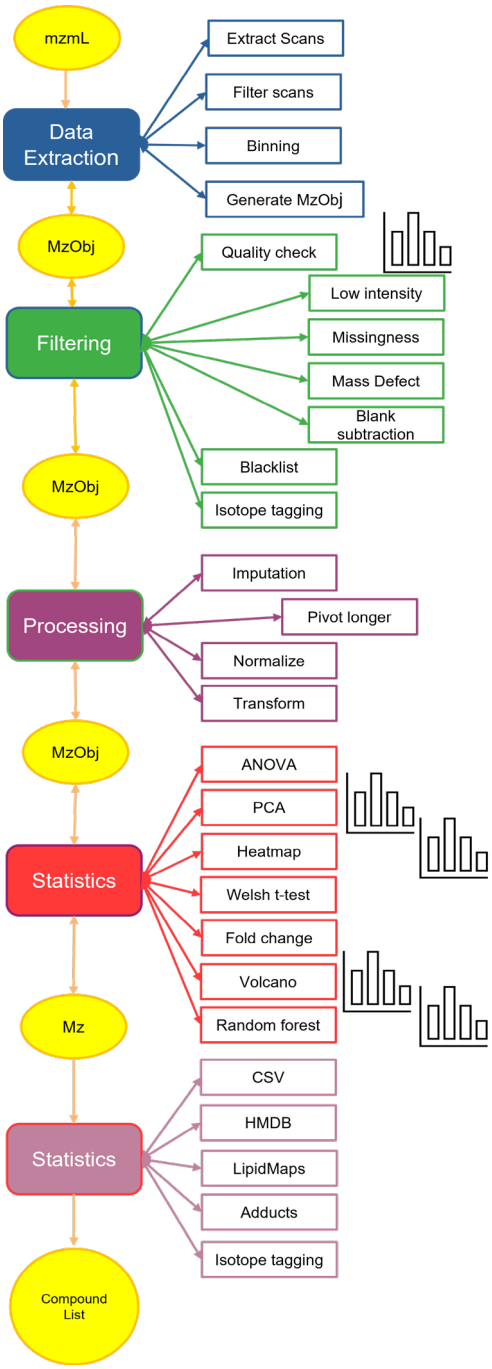


Figure 1. Overview of the Medusa data processing workflow.
Mass spectrometry files in .mzML format are first processed via data extraction, where scan information is binned, filtered, and converted into an MzObj. This object then undergoes filtering based on intensity thresholds, missingness, isotope tagging, and quality control metrics. Post-processing steps, including normalization, transformation, and imputation, prepare the data for statistical analyses. Statistical tools such as PCA, ANOVA, and random forest are applied to the processed MzObj. Identified *m/z* values are matched with compound databases using adduct rules and tagging strategies to assign compound annotations.

annotated feature lists after feature selection using machine learning or univariate tests. To accommodate inhomogeneous direct infusion data, the retention time information is not used in the processing and analysis pipeline. Furthermore, MeDUSA incorporates several
de-noising and filtering steps that are specifically tailored for untargeted single-cell experiments without the need for reference or quality control samples. All of the filtering is modular, so users may choose to bypass any of the filtering functions. Finally, the consistent and accessible data objects carried throughout the workflow allow the user to easily implement methods not included in the package. In this manuscript, the various functions of MeDUSA are highlighted and implemented on an untargeted single-cell metabolomic experiment of two cancer cell lines (HepG2 and HEK293T), measured as inhomogeneous cellular samples.

Materials and Methods

MeDUSA was developed and tested in a Windows environment, utilizing Docker to maintain dependencies. The package is publicly hosted on GitHub (https://github.com/laura-hetzel/MeDUSA).

Cell culture (validation data set)

HepG2 and HEK293T cells were individually cultured. The medium consisted of High Glucose DMEM (D6546, Sigma-Aldrich), supplemented with 10% Fetal Calf Serum (FCS), 2 mM L-Glutamine (25030081, Thermo Fisher Scientific) and
1% penicillin/streptomycin (15070063, Thermo Fisher Scientific) and was changed every three days during culture. To prepare the cells for live single-cell sampling (LCS), the cells were released from the plate with 1mM Trypsin; the Trypsin was deactivated with the medium. The cell suspension was then centrifuged for 4.5 min at 0.3 rcf. The supernatant was removed, leaving a cell pellet. The pellet was resuspended with 1ml of medium; 20µl of the cell suspension was added to a 35 mm petri dish. The cells were allowed to settle in incubation for at least four hours prior to sampling. Adherent cells are indicative of healthy cells for the cell lines used, and healthy cells are most representative of the cell line dynamics and resulting metabolome.

Cell sampling

For LSC, a Nikon microscope was used to view the cells. A micromanipulator (Narishige) was used to control the 5µm Humanix capillaries. After a cell was captured in the capillary, the capillary was stored

on dry ice until sampling was complete. All samples were then stored at -80°C until measurement. Freezing the cells ensured the metabolome would not change after the cells were removed from their controlled environment.

Mass spectrometry measurements

For mass spectrometry measurements, 80% Acetonitrile with 5mM ammonium formate and 20% deionized water was used as an ionization solvent. Just prior to the measurement, 4µl solvent was back filled into the capillary, which was then centrifuged at 1000 rpm for 1 second.

A Thermo Fisher Orbitrap Exploris 480 with a nano electrospray ionization source was used to analyze the single-cell samples. Acquisition was performed in a mass range of 50-1000 *m/z* in both positive and negative modes, with both small (30) and large (500) mass windows. For example, 500-1000 *m/z* and 50-80 *m/z* are both mass windows in the measurement method, with cycling through the smaller windows with SIM stitching to cover the full range comprising a majority of the measurement. Only MS1 data was acquired. The acquisition time was ten minutes and generated 1000-1300 scans per measurement. The voltage applied ranged was static per polarity during the measurement and was set at the beginning of the measurement to establish a stable spray. The voltage ranged from 900-1200V in positive mode and 1000-1300V in negative mode. The RF lens was 50%, the resolution was set to 120000, and the ion transfer tube temperature was 400° C.

The application of MeDUSA was evaluated with 180 measurements, including quality control (QC), cellular, and blank samples.

Results

Explanation of package

Setup

MeDUSA was developed to work with inhomogeneous direct infusion MS1 data that is converted to mzML files. All dependencies in the R package are version controlled within a Docker image, which must be running when using the package. Instructions on pulling and running the latest Docker image are available within the "Docker file" on the GitHub repository. MeDUSA also requires a file with metadata noting the sample type and phenotype for each sample. An example of the metadata with expected headers and logical inputs (TRUE/FALSE) is available in Table S1.

MeDUSA was created to be a modular package with standard objects (MzObj) being passed through the functions. Figure 1 provides a simplified

overview of the conceptual functions and the suggested order. A notable feature of MeDUSA is the inclusion of "magic functions" which incorporate multiple functions into one, which is detailed in Figure 2.

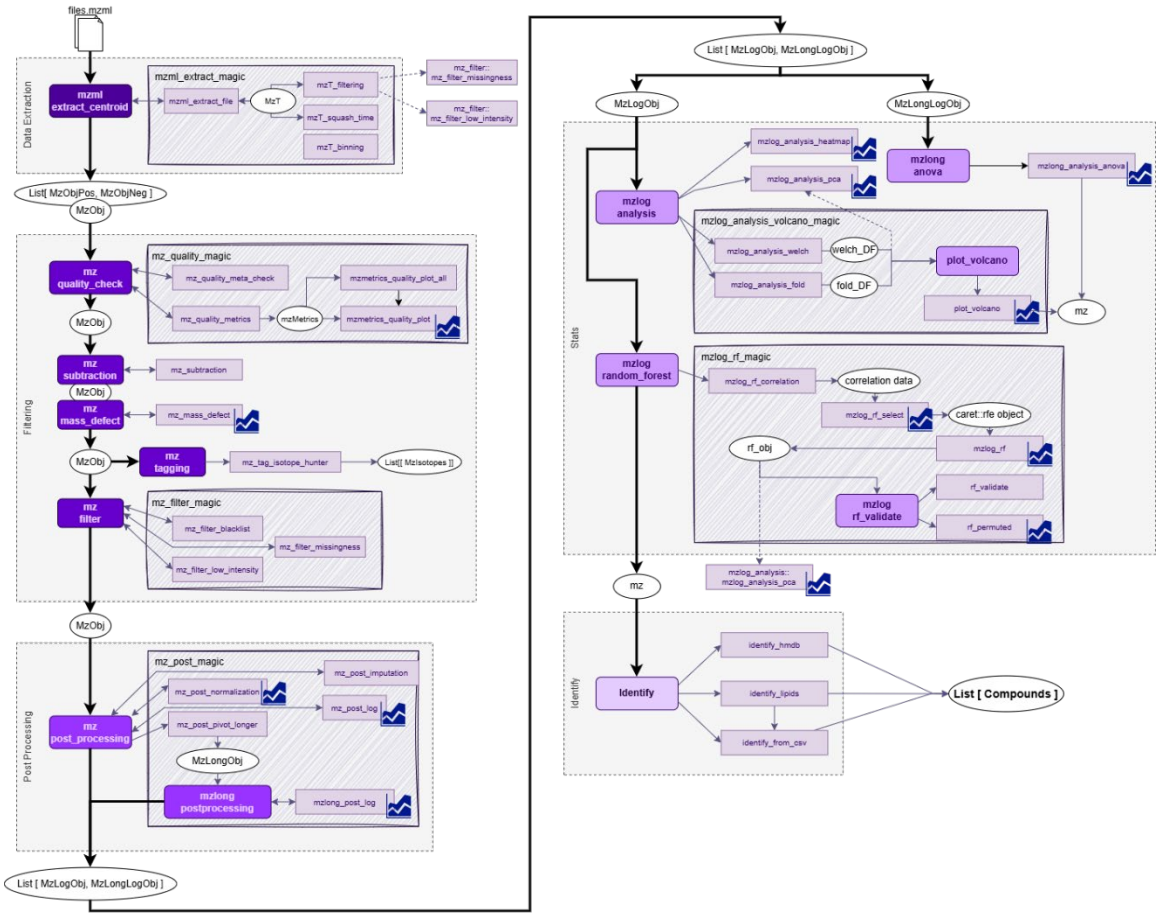


Figure 2. Overview of functions of MeDUSA. The actual functions that are called are placed into the recommended workflow. The magic functions are highlighted and the functions contributing to the magic functions are shown.

Data Extraction

The first step in MeDUSA is the parsing of the mzML files and formatting the data. The *mzml_extract_file* function extracts the spectral information, creating a time-based MzObj, which consists of the *m*/*z* values and their respective intensity at a specific time (scan). Spectral data that is in a profile

format is first smoothed using Savgol method[18] then centroided. The function is summarized in Figure S1. There can be two time-based MzObjs created per file if the measurement involved both positive and negative polarities. Each polarity should be extracted separately with the input parameter *polarity*, where extraction of both is the default value. It should be noted that this function is intended for only one file, however, the user can write a modification to apply it to multiple files.

Once the data was extracted from the file, it is in a format that can be passed into the *mzT_filtering* function. This function is summarized in Figure S2 and applies a missingness filter, low intensity threshold, and alignment. The missingness filter, applied to reduce the effect of artefacts, works by filtering out signals that appear in less than a specified number of scans. It defaults to 0.1 indicating that the feature should appear in at least 10% of scans to be preserved, otherwise, it is removed. The function allows the user to input a whole number indicating how many scans the identified peak must occur in, or any value between 0 and 1 indicating the fraction of scans that must contain the identified peak. The low intensity threshold is implemented to reduce the effect of noise; the default value is 1000. Data sets can be large with direct infusion data, so it is recommended to filter the data before alignment to reduce the computational load. The *mzT_filtering* function aligns the data via a modified binning method based on the version implemented in MALDIquant.[19] The binning method in MALDIquant was modified to allow the user to implement other methods to choose the value of the binning *m*/*z* signal.

After alignment, the data must be condensed from multiple scans per measurement each containing an intensity per *m*/*z* to a data frame (MzObj) containing only one averaged intensity per *m*/*z* per measurement. The *mzT_squash_time* function calculates an appropriate intensity value with the user defined computational parameter; the default is *mean.*

The modularity of MeDUSA allows the user to extract data in each individual file of a dataset; however, this is impractical for full datasets. To address this, the *extract_magic* function includes the aforementioned data extraction functions and loops through all mzML files in the specified location. This function outputs a list containing two data frames, one for each polarity. For the example 180 files, the *extract_magic* function required approximately 90 minutes to extract all data.

The *mz_quality_metrics* function formulates a table containing the minimum, maximum and median *m/z* values as well as the total number of peaks, and number of peaks with an intensity greater than 1,000, 10,000, and 100,000. For visualization, the *mzmetrics_quality_plot_all* function generates plots from the table. These functions enable the user to assess the data for any outliers that may be attributed to a measurement error. No data is removed by MeDUSA; any outliers must be manually removed by the user. The *mz_quality_meta_check* function assesses the meta data file to ensure that columns used later in functions are present and properly named. It also compares the names of the samples in the data set to the names in the metadata to ensure they match. Any discrepancies result in an error.

Data Filtering

In order to isolate the features unique to the cellular components, the spectra of cellular samples can be compared to the spectra of the blanks. The *mz_subtraction* function, conceptually summarized in Figure S3, can be used for blank subtraction in this case. It identifies features with an intensity at least three times higher in the cellular measurements compared to the non-cellular measurements. This threshold has a default value of three, however, it can be adjusted by the user. The *m/z* values which do not have higher intensities in the cellular data compared to the non-cellular are removed from all samples.

The *mz_mass_defect* function was used to remove features identified as salt cluster artefacts based on McMillan et al.[20] MeDUSA creates a plot, Figure S4, to visualize the results.

The *mz_filter_blacklist* function was included in the package such that the user could include a list of *m/z* values that should be omitted. This tool is best used for eliminating known contaminants from the data set.

The *mz_filter_missingness* function was used to remove features that could be artefacts. Inhomogeneous direct infusion measurements contain a significant number of features that may be detected in only a few measurements. As with the previous missingness filter, the package allows the user to input a whole number indicating the number of samples the feature must occur in, or a number less than one indicating the percentage of samples. The default threshold is 10%, which was used in this study.

The *mz_filter_low_intensity* function was used to ensure that all features remaining in the dataset had an appropriate intensity before statistical analysis. The user can choose a threshold different than that in the previous

filtering step. After assessing the filtering and data set, the threshold was set relatively low at 500 for this study.

The *mz_filter_magic* function encompasses the blacklist, missingness, and low intensity filters. The default values were the same for the individual functions compared to the magic function and therefore yield the same end data set.

The *mz_tag_isotope_hunter* function looks for potential Carbon-13 (C_{13}) isotopes in the data. The function compares each *m/z* value to its surrounding *m/z* values to determine if a surrounding value is a likely isotope. The function generates a list of *m/z* values that are a potential isotope and the user can decide to remove the values from the data set manually. Only C_{13} is considered for the isotope.

Post filtering processing

Once the data was filtered to isolate the features free from background features, noise, and artefacts, the data set was prepared for statistical analysis. The *mz_post_imputation* function was used to remove any zero intensity values with imputation since some statistical functions cannot process a zero value. This function replaces the zero values with a random value between 10 and the designated noise level, which ensures the imputed values will not influence the statistical analysis. This method may introduce bimodal distributions, and if this is not suitable to users, the modularity of MeDUSA allows the user to use another method such as RF or regression-based imputation. However, due to the high degree of missingness in single-cell data, this should be done with extreme caution.[21]

It is likely that cells will have a slight variation in size, which can undoubtedly influence the intensity of the measured metabolites. To mitigate this, Dieterle et al proposed quotient normalization which chooses a reference sample from the measurements, compares to the measurement to be normalized and calculates a ratio.[22] The median of all ratios determines the dilution factor for the specific measurement. The *mz_post_normalization* function uses this technique as an option to overcome this variation .

Some of the mathematical functions used for statistics require the data to be in a format containing an "intensity" column. To achieve this, a pivot longer was implemented in the *mz_post_pivot_longer* function. The format can be seen in Figure S5. As an additional component of the function, MeDUSA also output plots to visualize the variance in the data after all filtering and normalization. The plots are available as a scatter plot (Figure S6) or a box plot.

There are two log transformation functions available in MeDUSA, one (*mzlong_post_log*) which applies Log(2) to the long format, and the other (*mz_post_log*) applies the log scale to the wide format. Both functions scale only the intensity values. For a streamlined, user-friendly experience, the *mz_post_magic* function implements the imputation, normalization, and scaling functions all in one function.

Statistics

There are multiple statistical options in the MeDUSA package, including PCA, Welch t-test, ANOVA, and random forest. With the modularity of the package, analyses can be performed in any order and with any combination of tests. The *mzlog_analysis_pca* function expects the data to be logged, which is why the post processing should be completed first.

The output of *mzlog_analysis_welch* is a data frame consisting of a p-value for each *m/z* value and a "True" or "False" Boolean for the p-value ≤0.05, ≤0.10, and ≤0.15, all of which can be sorted, as shown in Figure S7. The *mzlog_analysis_fold* function outputs a data frame of two columns, one for *m/z*, and the other for the fold change as shown in Figure S8. One method of visualization of both the fold change and Welch t-test is the volcano plot. The *mzlog_analysis_volcano_magic* accepts the data set for each phenotype and calculates both the p-value using Welch t-test and the fold change in one function, also outputting the volcano plot. Metabolomics studies often utilize a heatmap to visualize the data. The *mzlog_analysis_heatmap* creates a heatmap for the data set. Unlike some of the other statistical functions, the data set was passed into the function as a complete data set, not separated by phenotype.

Random forest is a modeling tool used to determine which features contribute to the prediction of phenotype and is summarized in Figure S9. There are three functions that were used to complete the model, all of which are included in the *mzlog_rf_magic* function. The *mzlog_rf_correlation* function was used to identify and remove highly correlated peaks. The data set generated in this function was passed into the *mzlog_rf_select* function which assesses the features and selects those that are important to the model. The output of this function was an *rf object*, which is then passed into the final function, *mzlog_rf* which split the data set to train and test the model.

The *identify_hmdb* function allows the user to input a list of *m/z* values and compares the list to the Human Metabolome Database (HMDB). The function also allows adducts to be passed in as a parameter for more accurate

annotation. Similarly, the *identify_lipids* function compared the features to LipidMaps for identification. It is possible for the user to include a targeted list using the *identify_from_csv* to create a semi-targeted analysis.

Application

Setup

Inhomogeneous single-cell direct infusion measurements containing 30 mass range windows spanning an *m/z* range of 50 to 1080 were used to assess the use case of MeDUSA. To prepare the files for processing with MeDUSA, all (Thermo) .raw files were converted to .mzML files using ProteoWizard MS Convert.[23] The data was processed using the MeDUSA version 3 of the image.

Data Extraction

For validation purposes, the *mzml_extract_file* function was used on two quality control (QC) samples. Table S2 summarizes the number of scans and features identified in each file, per polarity.

The *mzT_filtering* function was applied and the reduction in the number of features per file is summarized in Table S3.

The *mzT_squash_time* function was used to reduce the data frame to one intensity per *m/z* value, eliminating the *per scan* information.

After the validation of the individual functions, the full data set was extracted with the *extract_magic* function. Each data frame was manually assigned to a separate variable to be processed independently based on polarity. The *extract_magic* function is computationally heavy and is one of the longest functions to run. On a MAC laptop, the validation data set of 180 measurements took approximately 40 minutes per polarity to extract.

To validate the methods incorporated in MeDUSA for peak identification and alignment, the measurements of standards were used to evaluate the ppm error across the processing steps. The MeDUSA extracted and aligned peaks of 13 standard compounds were compared with the measured raw values (Figure 3), and the theoretical *m/z* value (Table 1). All error values were below 3 ppm.

Since the error is not completely computational, the measured *m/z* was also compared to the theoretical *m/z* (Figure S10).

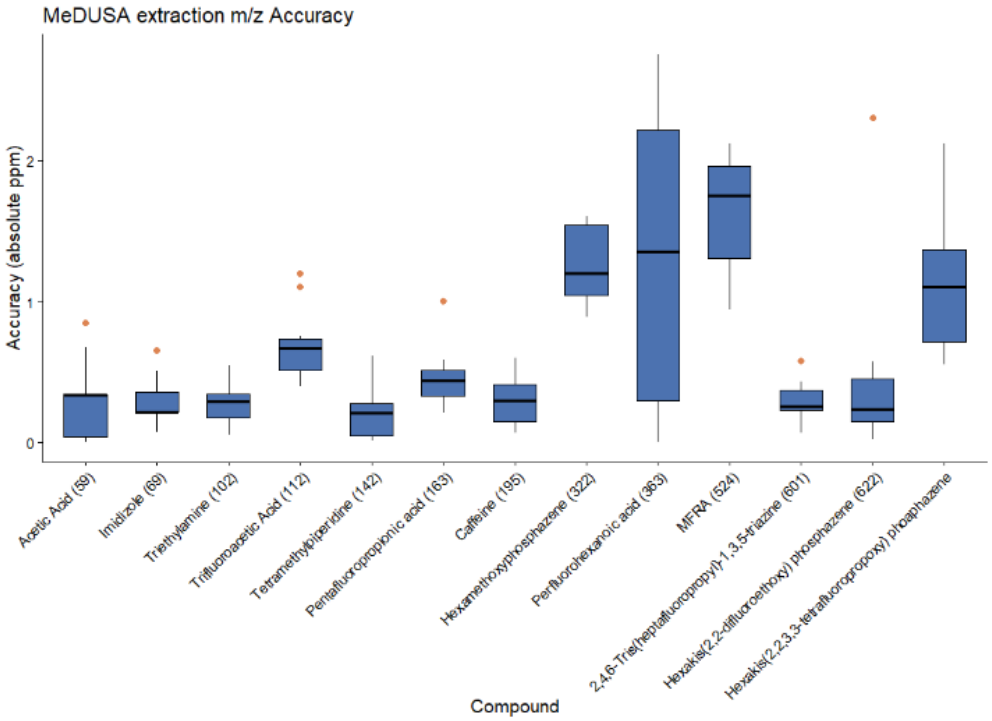


Figure 3: Absolute error in ppm of 13 compounds when comparing measured *m/z* values to *m/z* values after processing with MeDUSA.

Table 1: Error in ppm of calculated m/z values calculated of 13 compounds and the extracted and aligned m/z values of the same compounds after processing with MeDUSA

Theoretical m/z	MeDUSA output m/z	Error (ppm)
59.0139	59.01391	-0.169451604
69.0447	69.04467486	0.364111945
102.1277	102.1275743	1.230812013
112.9856	112.9857553	-1.374511442
142.159	142.1589378	0.437538249
162.9824	162.9823438	0.344597411
195.0876	195.0875971	0.014865117
322.0481	322.0478731	0.70455314
362.9696	362.9682683	3.668985788
524.2649	524.2637756	2.144717298
601.9779	601.9789	-1.661190552
622.029	622.0291943	-0.312364858
922.0098	922.0111338	-1.446622368

Once the data frames were prepared, the quality of each measurement was assessed with the *mz_quality_metrics* function. For visualization, the *mzmetrics_quality_plot_all* function was used. If a sample had an abnormally low maximum *m*/*z*, or minimum *m*/*z* value, the sample may be an outlier due to an incomplete or unsucessful measurement. Additionally, if a sample had relatively low intensities, it may be an outlier. Potential outliers from each plot were identified and if a sample was flagged in multiple plots, the raw spectra was evaluated and the sample removed from the data set if it was determined that the measurement was unreliable. The plots generated are shown in Figures S11 through 17. The *mz_quality_meta_check* function was used to ensure the metadata was prepared properly.

Data Filtering

The dataset was split into two variables, the cellular measurements and the non-cellular measurements using the *mztools_filter* function, and the *mz_subtraction* function was used to filter the background signal.

The data was filtered with default values in *mz_mass_defect*, *mz_filter_missingness*, and *mz_filter_low_intensity*. The number of features removed with each filter are summarized in Table S4. The *mz_filter_magic* function was confirmed to produce the same results as the individual functions.

Post filtering processing

The data was processed with *mz_post_imputation* and *mz_post_normalization*, generating the plot shown in Figure S17. The data was transformed with the *mz_post_log* function to prepare for statistical analysis.

Statistics

A PCA plot was generated with the *mzlog_analysis_pca* function, shown in Figure S19. Additionally, the metadata file was passed in as a parameter to group based on phenotype, which is the default grouping.

For the Welch t-test, the two different phenotypes were assigned to unique variables using the same tools as were used for the blank subtraction. The two data sets were then passed into the *mzlog_analysis_welch*.

The *mzlog_analysis_fold* was used to calculate the fold change of one phenotype compared to the other at each *m*/*z* value. The *mzlog_analysis_volcano_magic* function was used to generate the volcano

plots shown in Figure S20. The *mzlog_analysis_heatmap* function was used to generate the heatmap shown in Figure S21.

The random forest functions were used to identify the features used to predict the phenotype (Figure S9). While the functions have multiple outputs, the multidimensional scaling (MDS) plot was used to visually represent the distinct clustering of the phenotypes despite the complex data. The mean decrease accuracy identified the *m/z* features that most contributed to the differentiation. Finally, the accuracy of the model was also evaluated with a confusion matrix and ROC plot (Figure S22). Random forest was also computationally heavy and took approximately 40 minutes to execute.

The generated list of significant features was then passed into the *identify_hmdb* function for comparison to the HMDB and annotation.

Conclusion

In summary, MeDUSA was created to lay the groundwork for current and future inhomogeneous direct infusion untargeted single-cell metabolomics workflows. The functions in the pipeline were designed to be modular and interchangeable with user-generated functions while giving the user full control of the parameters to be used in each step of their data processing and analysis pipeline. Furthermore, MeDUSA incorporated several filtering options that are optimized for single-cell metabolomic datasets and the relatively high level of noise they possess.

Despite these advantages, there are several improvements that can be made to both the pipeline and the package itself. Currently, MeDUSA implements various packages, which creates dependencies and a need for the dependencies to be managed to avoid errors. Additionally, the binning process is very memory intensive and there is the possibility of crashing the container if the computer used does not have enough memory. Finally, the docker image can take more than one hour to build as it contains the package and two databases, and the image should be rebuilt if there are major changes to the package. From start to finish, the validation took approximately three hours to process (approximately 1 min per file), after the Docker image was built. Future areas of improvement to the pipeline itself involve adding more filtering options, and additional peak alignment and clustering tools. Furthermore, it is worth noting that results analyzed using MeDUSA, as well as other tools share a large limitation when it comes to their interpretation. Mainly, since the data is direct infusion-based

annotation is typically done only on the *m*/*z* value alone, which reduces confidence and the translation potential of the results. From the computational side, a potential way to increase confidence in the annotations is to develop tools that check for potential isotopic patterns and other parameters to narrow down the number of possible annotations per *m*/*z*. From the lab side, incorporating techniques that provide orthogonal structural information such as ion mobility without sacrificing throughput is also a possibility to improve annotation confidence.

Supporting Information:

Table S1: metadata format example. Tables S2-S4: Extraction and filtering overview. Figure S1: Extraction process of an mzML file. Figures S2-22 output plots from MeDUSA.

Data availability

The code and example data to validate MeDUSA are available at https://github.com/laura-hetzel/MeDUSA

Funding Sources

This research was funded by the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement number 861196 , NWO grant number OCENW.XS23.1.070 and the ZonMw project VOILA, project number: 457001001.

Author Contributions

L.A. Castaneda: Methodology, validation, format analysis, investigation, writing. E. Hetzel: Software, Resources. F. Gül: Visualization, data curation. T. Hankemeier: Funding acquisition, supervision. A. Ali: Conceptualization, methodology, writing, supervision. All authors have given approval to the final version of the manuscript.

References

(1) Wei, D.; Xu, M.; Wang, Z.; Tong, J. The Development of Single-Cell Metabolism and Its Role in Studying Cancer Emergent Properties. *Front Oncol* **2022**, *11*, 814085. https://doi.org/10.3389/fonc.2021.814085.

(2) Chen, X.; Sun, M.; Yang, Z. Single Cell Mass Spectrometry Analysis of Drug-Resistant Cancer Cells: Metabolomics Studies of Synergetic Effect of Combinational Treatment. *Analytica Chimica Acta* **2022**, *1201*, 339621. https://doi.org/10.1016/j.aca.2022.339621.

(3) Wang, R.; Zhao, H.; Zhang, X.; Zhao, X.; Song, Z.; Ouyang, J. Metabolic Discrimination of Breast Cancer Subtypes at the Single-Cell Level by Multiple Microextraction Coupled with Mass Spectrometry. *Anal. Chem.* **2019**, *91* (5), 3667–3674. https://doi.org/10.1021/acs.analchem.8b05739.

(4) Tian, J.; Fu, G.; Xu, Z.; Chen, X.; Sun, J.; Jin, B. Urinary Exfoliated Tumor Single-Cell Metabolomics Technology for Establishing a Drug Resistance Monitoring System for Bladder Cancer with Intravesical Chemotherapy. *Medical Hypotheses* **2020**, *143*, 110100. https://doi.org/10.1016/j.mehy.2020.110100.

(5) Abouleila, Y.; Onidani, K.; Ali, A.; Shoji, H.; Kawai, T.; Lim, C. T.; Kumar, V.; Okaya, S.; Kato, K.; Hiyama, E.; Yanagida, T.; Masujima, T.; Shimizu, Y.; Honda, K. Live Single Cell Mass Spectrometry Reveals Cancer-specific Metabolic Profiles of Circulating Tumor Cells. *Cancer Science* **2019**, *110* (2), 697–706. https://doi.org/10.1111/cas.13915.

(6) Shrestha, B. Single-Cell Metabolomics by Mass Spectrometry. In *Single Cell Metabolism*; Shrestha, B., Ed.; Methods in Molecular Biology; Springer New York: New York, NY, 2020; Vol. 2064, pp 1–8. https://doi.org/10.1007/978-1-4939-9831-9_1.

(7) Chen, Y.; Li, E.-M.; Xu, L.-Y. Guide to Metabolomics Analysis: A Bioinformatics Workflow. *Metabolites* **2022**, *12* (4), 357. https://doi.org/10.3390/metabo12040357.

(8) Yao, H.; Zhao, H.; Zhao, X.; Pan, X.; Feng, J.; Xu, F.; Zhang, S.; Zhang, X. Label-Free Mass Cytometry for Unveiling Cellular Metabolic Heterogeneity. *Anal. Chem.* **2019**, *91* (15), 9777–9783. https://doi.org/10.1021/acs.analchem.9b01419.

(9) Cahill, J. F.; Riba, J.; Kertesz, V. Rapid, Untargeted Chemical Profiling of Single Cells in Their Native Environment. *Anal. Chem.* **2019**, *91* (9), 6118–6126. https://doi.org/10.1021/acs.analchem.9b00680.

(10) Yu, H.; Low, B.; Zhang, Z.; Guo, J.; Huan, T. Quantitative Challenges and Their Bioinformatic Solutions in Mass Spectrometry-Based Metabolomics. *TrAC Trends in Analytical Chemistry* **2023**, *161*, 117009. https://doi.org/10.1016/j.trac.2023.117009.

(11) Tsugawa, H.; Ikeda, K.; Takahashi, M.; Satoh, A.; Mori, Y.; Uchino, H.; Okahashi, N.; Yamada, Y.; Tada, I.; Bonini, P.; Higashi, Y.; Okazaki, Y.; Zhou, Z.; Zhu, Z.-J.; Koelmel, J.; Cajka, T.; Fiehn, O.; Saito, K.; Arita, M.; Arita, M. A Lipidome Atlas in MS-DIAL 4. *Nat Biotechnol* **2020**, *38* (10), 1159–1163. https://doi.org/10.1038/s41587-020-0531-2.

(12) Smith, C. A.; Want, E. J.; O'Maille, G.; Abagyan, R.; Siuzdak, G. XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification. *Anal. Chem.* **2006**, *78* (3), 779–787. https://doi.org/10.1021/ac051437y.

(13) Misra, B. B. New Software Tools, Databases, and Resources in Metabolomics: Updates from 2020. *Metabolomics* **2021**, *17* (5), 49. https://doi.org/10.1007/s11306-021-01796-1.

(14) Schmid, R.; Heuckeroth, S.; Korf, A.; Smirnov, A.; Myers, O.; Dyrlund, T. S.; Bushuiev, R.; Murray, K. J.; Hoffmann, N.; Lu, M.; Sarvepalli, A.; Zhang, Z.; Fleischauer, M.; Dührkop, K.; Wesner, M.; Hoogstra, S. J.; Rudt, E.; Mokshyna, O.; Brungs, C.; Ponomarov, K.;

Mutabdžija, L.; Damiani, T.; Pudney, C. J.; Earll, M.; Helmer, P. O.; Fallon, T. R.; Schulze, T.; Rivas-Ubach, A.; Bilbao, A.; Richter, H.; Nothias, L.-F.; Wang, M.; Orešič, M.; Weng, J.-K.; Böcker, S.; Jeibmann, A.; Hayen, H.; Karst, U.; Dorrestein, P. C.; Petras, D.; Du, X.; Pluskal, T. Integrative Analysis of Multimodal Mass Spectrometry Data in MZmine 3. *Nat Biotechnol* **2023**, *41* (4), 447–449. https://doi.org/10.1038/s41587-023-01690-2.

(15) Husen, P.; Tarasov, K.; Katafiasz, M.; Sokol, E.; Vogt, J.; Baumgart, J.; Nitsch, R.; Ekroos, K.; Ejsing, C. S. Analysis of Lipid Experiments (ALEX): A Software Framework for Analysis of High-Resolution Shotgun Lipidomics Data. *PLoS ONE* **2013**, *8* (11), e79736. https://doi.org/10.1371/journal.pone.0079736.

(16) Nguyen, Q.-H.; Nguyen, H.; Oh, E. C.; Nguyen, T. Current Approaches and Outstanding Challenges of Functional Annotation of Metabolites: A Comprehensive Review. *Briefings in Bioinformatics* **2024**, *25* (6), bbae498. https://doi.org/10.1093/bib/bbae498.

(17) Ross, D. H.; Guo, J.; Bilbao, A.; Huan, T.; Smith, R. D.; Zheng, X. Evaluating Software Tools for Lipid Identification from Ion Mobility Spectrometry–Mass Spectrometry Lipidomics Data. *Molecules* **2023**, *28* (8), 3483. https://doi.org/10.3390/molecules28083483.

(18) Savitzky, Abraham.; Golay, M. J. E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.* **1964**, *36* (8), 1627–1639. https://doi.org/10.1021/ac60214a047.

(19) Gibb, S.; Strimmer, K. MALDIquant: A Versatile R Package for the Analysis of Mass Spectrometry Data. *Bioinformatics* **2012**, *28* (17), 2270–2271. https://doi.org/10.1093/bioinformatics/bts447.

(20) McMillan, A.; Renaud, J. B.; Gloor, G. B.; Reid, G.; Sumarah, M. W. Post-Acquisition Filtering of Salt Cluster Artefacts for LC-MS Based Human Metabolomic Studies. *J Cheminform* **2016**, *8* (1), 44. https://doi.org/10.1186/s13321-016-0156-0.

(21) Krutkin, D. D.; Thomas, S.; Zuffa, S.; Rajkumar, P.; Knight, R.; Dorrestein, P. C.; Kelley, S. T. To Impute or Not To Impute in Untargeted Metabolomics─That Is the Compositional Question. *J. Am. Soc. Mass Spectrom.* **2025**, *36* (4), 742–759. https://doi.org/10.1021/jasms.4c00434.

(22) Dieterle, F.; Ross, A.; Schlotterbeck, G.; Senn, H. Probabilistic Quotient Normalization as Robust Method to Account for Dilution of Complex Biological Mixtures. Application in 1H NMR Metabonomics. *Anal Chem* **2006**, *78* (13), 4281–4290. https://doi.org/10.1021/ac051632c.

(23) Chambers, M. C.; Maclean, B.; Burke, R.; Amodei, D.; Ruderman, D. L.; Neumann, S.; Gatto, L.; Fischer, B.; Pratt, B.; Egertson, J.; Hoff, K.; Kessner, D.; Tasman, N.; Shulman, N.; Frewen, B.; Baker, T. A.; Brusniak, M.-Y.; Paulse, C.; Creasy, D.; Flashner, L.; Kani, K.; Moulding, C.; Seymour, S. L.; Nuwaysir, L. M.; Lefebvre, B.; Kuhlmann, F.; Roark, J.; Rainer, P.; Detlev, S.; Hemenway, T.; Huhmer, A.; Langridge, J.; Connolly, B.; Chadick, T.; Holly, K.; Eckels, J.; Deutsch, E. W.; Moritz, R. L.; Katz, J. E.; Agus, D. B.; MacCoss, M.; Tabb, D. L.; Mallick, P. A Cross-Platform Toolkit for Mass Spectrometry and Proteomics. *Nat Biotechnol* **2012**, *30* (10), 918–920. https://doi.org/10.1038/nbt.2377.

Supporting information

Table S1. Metadata example

Example subset of the metadata, specifically the column names, that are expected in MeDUSA. Note that the "FALSE" values in the filtered out column are logical values, not characters.

measurement	filename	type	phenotype	filtered_out
1	sumr_blank_001	IS_Blank	na	FALSE
2	sumr_media_002	media	na	FALSE
3	sumr_media_003	media	na	FALSE
4	sumr_qc_004	QC	na	FALSE
5	sumr_hek_005	cell	HEK293T	FALSE
6	sumr_hep_006	cell	HepG2	FALSE
7	sumr_hek_007	cell	HEK293T	FALSE
8	sumr_hek_008	cell	HEK293T	FALSE
9	sumr_hek_009	cell	HEK293T	FALSE
10	sumr_hep_010	cell	HepG2	FALSE

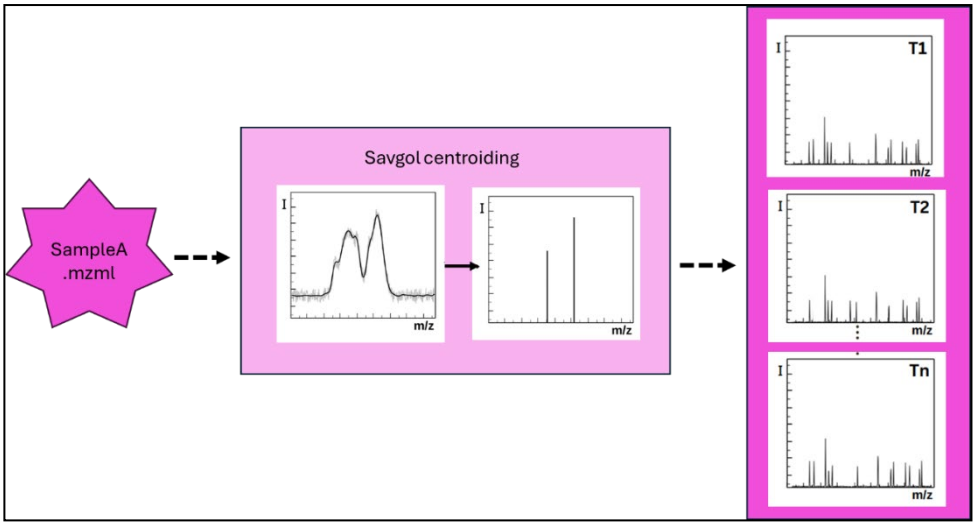


Figure S1. Extraction process of an mzML file.
Conceptual figure showcasing the function *mzml_extract_file*. The mzML file is smoothed with the Savgol method and then centroided.

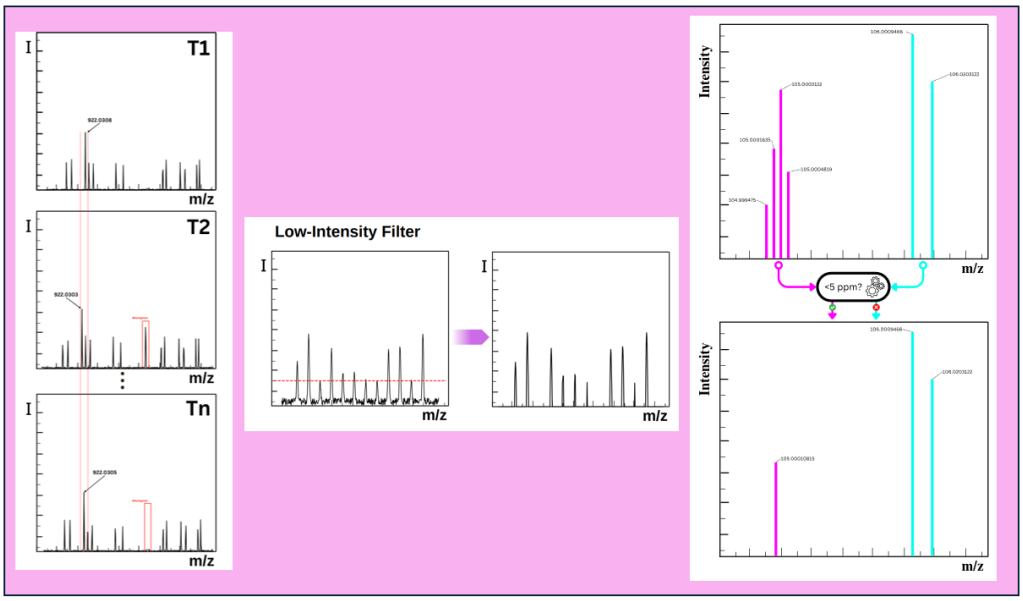


Figure S2. Overview of mzT_filtering.
Conceptual overview of the *mzT_filtering* function, which applies a missingness filter (left), low intensity threshold (center), and alignment (right) to reduce artefacts, noise, and redundant features before downstream analysis. The alignment filter assesses surrounding peaks within 5ppm (user defined) to determine if peaks are unique or are the same peak with a mass shift. Any nonunique peaks are condensed into one peak (pink) and unique peaks remain unchanged (blue).

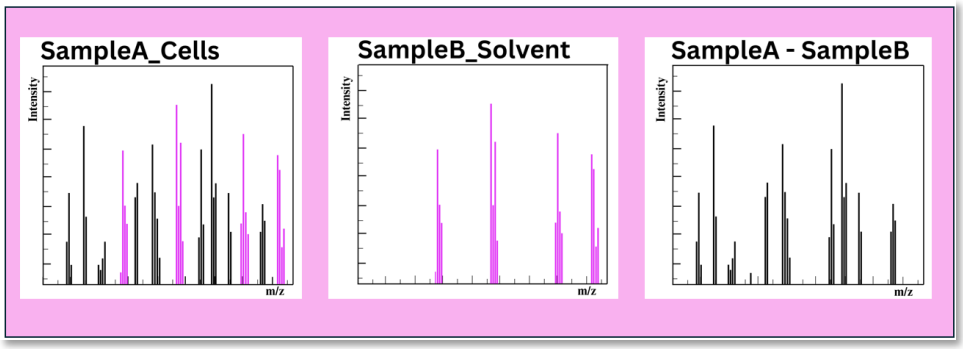


Figure S3. Example of feature-based background subtraction.
Conceptual showcase of the function *mz_subtraction*. Example spectra from SampleA (cells) and SampleB (solvent) are shown. Pink peaks indicate *m/z* values corresponding to solvent data in both spectra. Subtraction of SampleB signals from SampleA yields a cleaned spectrum highlighting cell derived features.

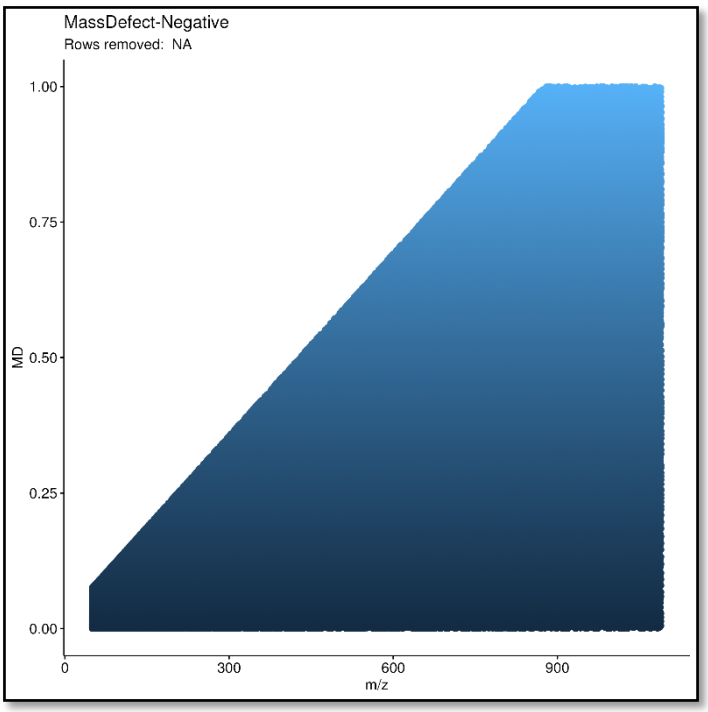


Figure S4. Mass defect plot.
Data generated output of the function *mz_mass_defect*. Each point represents a detected feature plotted by its *m*/*z* and corresponding mass defect. The triangular distribution reflects expected MD boundaries.

mz	sample_name	intensity
50.00367	neg_sumr_hek_005	550.25085
50.00367	neg_sumr_hek_007	228.18421
50.00367	neg_sumr_hek_008	59.50877
50.00367	neg_sumr_hek_009	59041.84726
50.00367	neg_sumr_hek_014	347.06897
50.00367	neg_sumr_hek_018	109989.04549
50.00367	neg_sumr_hek_020	110383.58471
50.00367	neg_sumr_hek_023	261.43974
50.00367	neg_sumr_hek_025	101601.51688

Figure S5. Output format of the mz_post_pivot_longer function.
Data generated output of the function *mz_post_pivot_longer*. This function does not apply filtering but reshapes the dataset into a long format with three columns: *m/z*, sample name, and intensity. Each *m/z* value appears once per sample, preparing the data for downstream statistical analyses and visualizations.

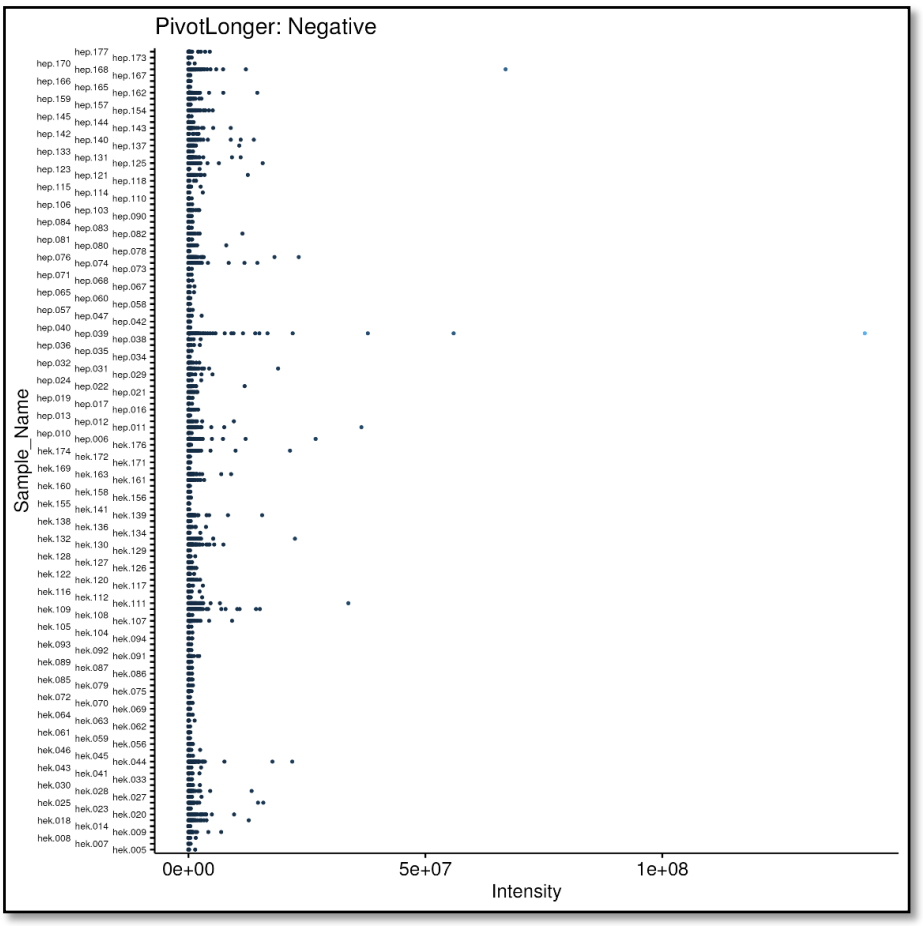


Figure S6. Scatter plot of sample intensities.
Data generated output of the function *mz_post_pivot_longer*. This visualization provides an overview of intensity distribution across all samples without applying any filtering or normalization.

mz	p	p_05	p_10	p_15
201.04067	0.05013174	FALSE	TRUE	TRUE
171.03466	0.05026531	FALSE	TRUE	TRUE
157.14420	0.05027937	FALSE	TRUE	TRUE
504.22769	0.05032580	FALSE	TRUE	TRUE
471.07619	0.05032631	FALSE	TRUE	TRUE
60.06742	0.05037599	FALSE	TRUE	TRUE
128.03399	0.05044969	FALSE	TRUE	TRUE
671.47343	0.05046230	FALSE	TRUE	TRUE

Figure S7. Welch's t-test comparison between groups.
Data generated output of the function *mzlog_analysis_welch*. No features passed the adjusted p-value threshold of 0.05; however, several features were significant at less stringent thresholds of 0.10 and 0.15**.**

mz	fold
50.00367	1.0348940
52.01932	1.0105436
55.03023	0.9869654
56.00170	0.9006314
56.01425	0.9882243
57.00951	0.9779229
58.02991	1.0382888

Figure S8. Fold change output table.
Data generated output of the function *mzlog_analysis_fold*. Displayed are *m/z* values alongside their corresponding fold change values.

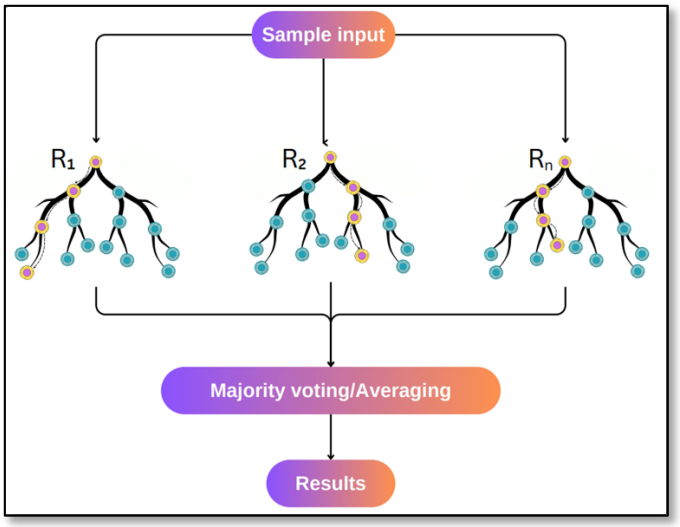


Figure S9. Random forest.
A. Concept figure showcasing how random forest assesses sample data to train a model to predict a phenotype. By going through multiple decision trees, it reaches a more accurate prediction or result than assessing only one decision point, in this case *m*/*z*. Despite multiple trees, there is final prediction of the phenotype.

Table S2. QC File Extraction Overview.
Showcase of the function *mzml_extract_file*. Number of scans and total extracted *m/z* features for each QC file, separated by ionization polarity.

QC file	Polarity	# scans	# features
004	Positive	584	385,173
004	Negative	569	8,562
181	Positive	573	78,815
181	Negative	544	354,948

Table S3. Overview of filtered *m/z* features per QC file.
Showcase of the function *mzT_filtering*. The number of features removed due to missingness and low intensity are shown separately for each polarity per file.

QC file	Polarity	Missingness filter	Low intensity filter
004	Positive	325,366	2,143
004	Negative	6,047	0
181	Positive	0	18,528
181	Negative	322,056	12,270

Table S4. Feature reduction by filtering step.
Showcase of the function *mz_filter_magic*. Total number of *m/z* features removed after applying the filtering criteria, mass defect, missingness and low intensity simultaneously, shown separately for positive and negative modes.

Filter	Reduction, positive	Reduction, negative
Mass Defect	**170,895**	**167,356**
Missingness	**145,873**	**138,124**
Low intensity	**0**	**0**

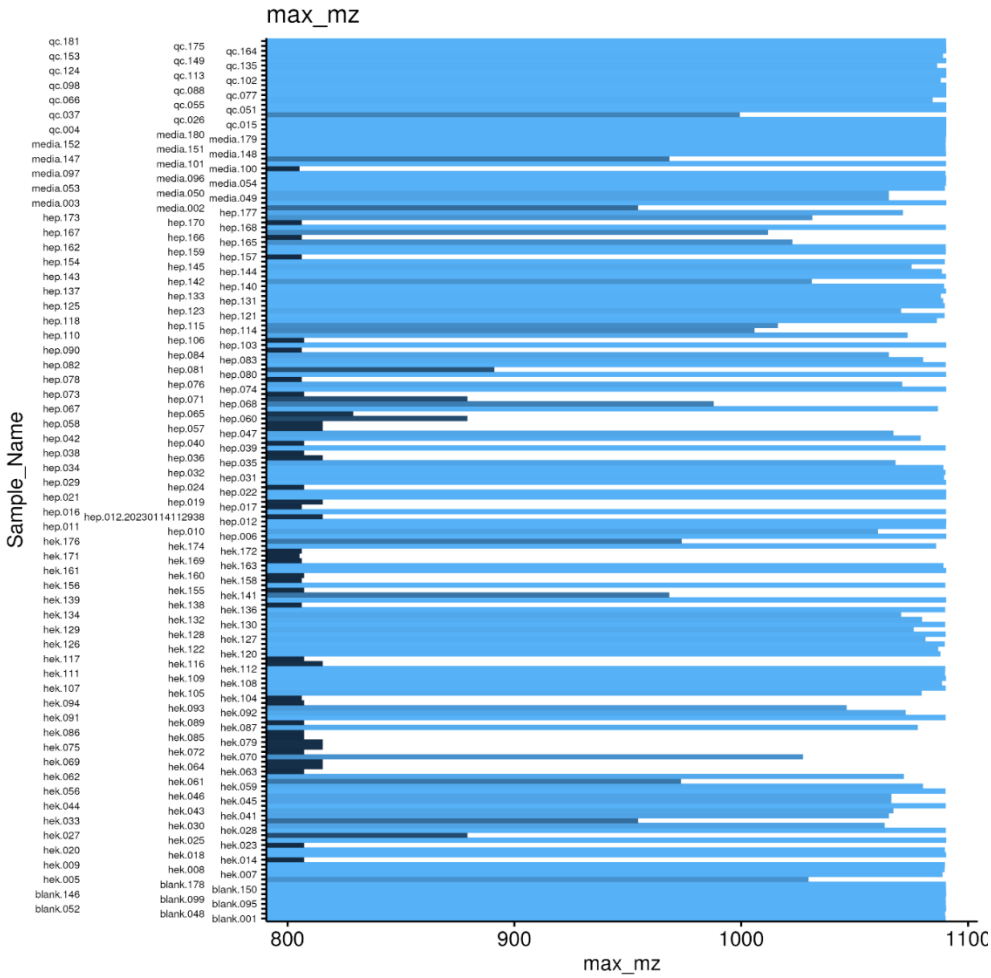


Figure S10. Maximum detected *m/z* per sample.
Data generated output of the function *mzmetrics_quality_plot_all*. Each horizontal bar represents the highest *m/z* value recorded in a given sample. Samples are grouped by type (e.g., QC, media, HepG2, HEK293T, blank) and sorted by sample name.

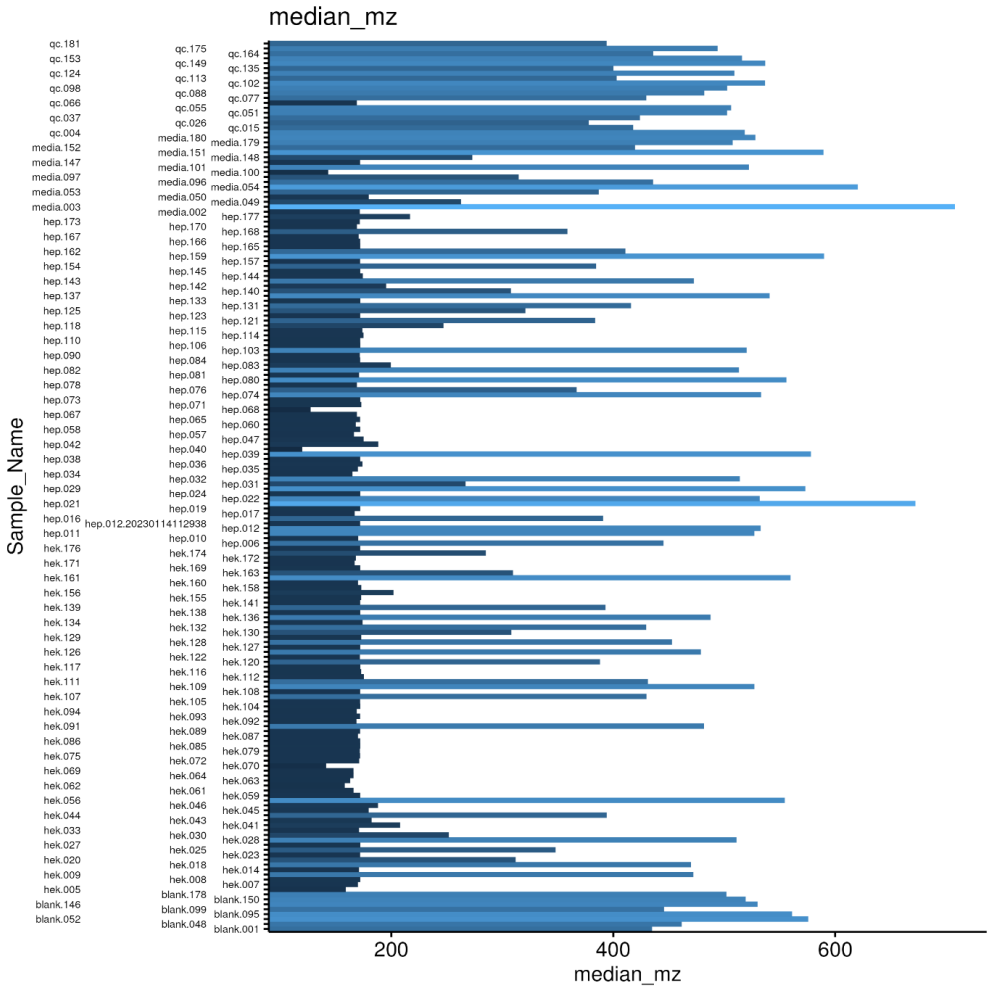


Figure S11. Median *m/z* per sample.
Data generated output of the function *mzmetrics_quality_plot_all*. Each horizontal bar represents the median *m/z* value recorded in a given sample. Samples are grouped by type (e.g., QC, media, HepG2, HEK293T, blank) and sorted by sample name.

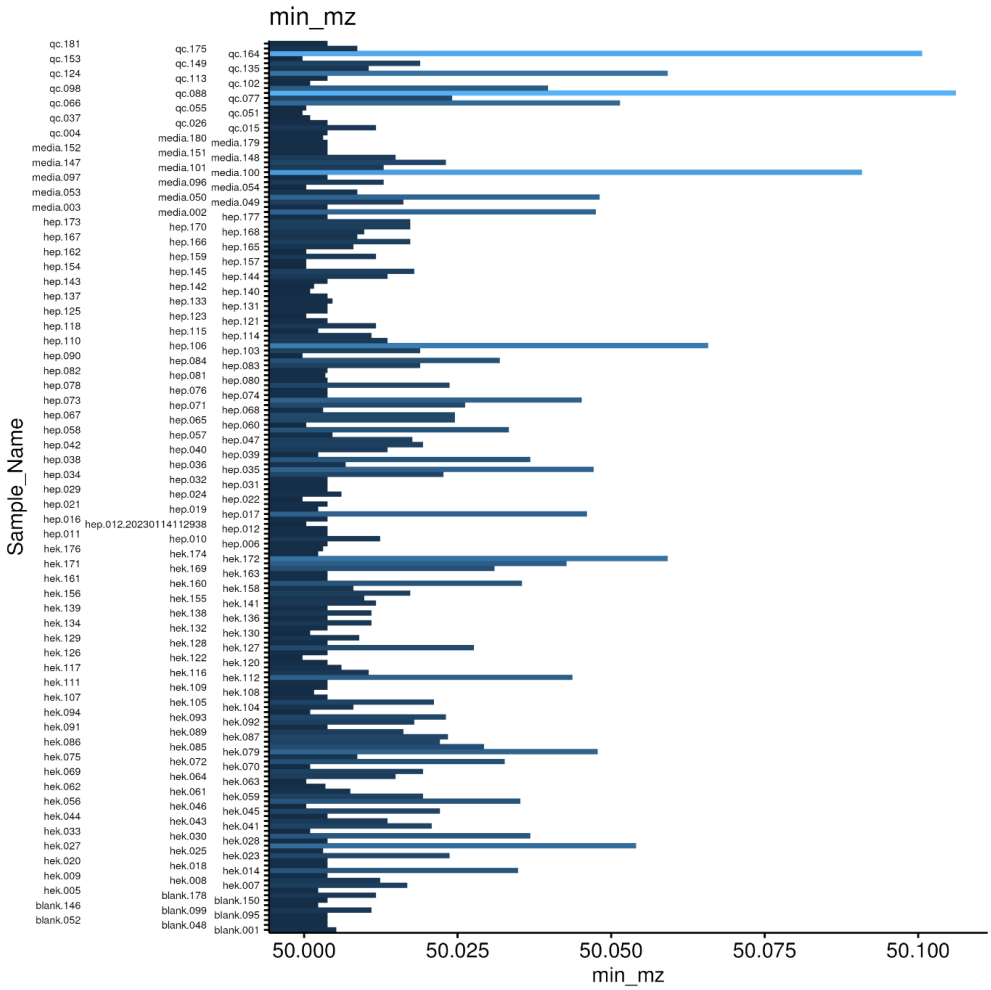


Figure S12. Minimum detected *m/z* per sample.
Data generated output of the function *mzmetrics_quality_plot_all*. Each horizontal bar represents the lowest *m/z* value recorded in a given sample. Samples are grouped by type (e.g., QC, media, HepG2, HEK293T, blank) and sorted by sample name.

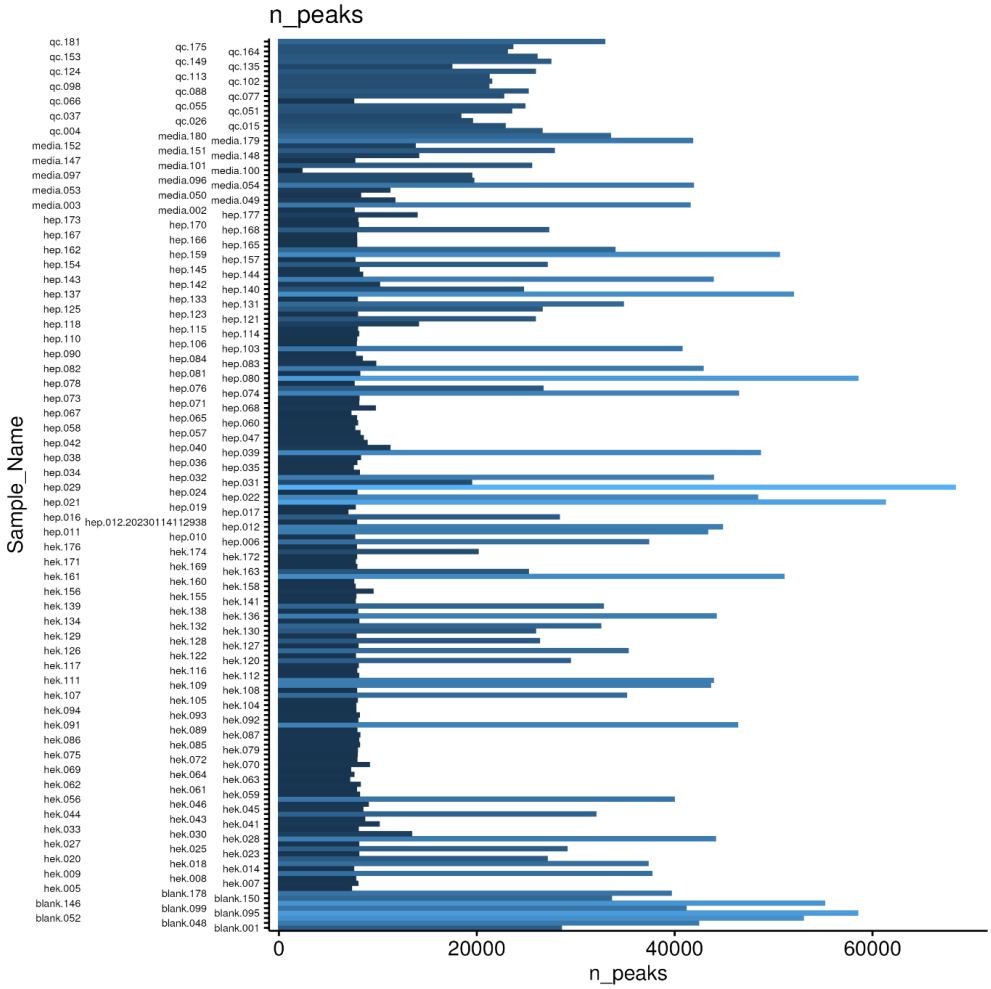


Figure S13. Number of detected peaks per sample.
Data generated output of the function *mzmetrics_quality_plot_all*. Each horizontal bar represents the number of peaks recorded in a given sample. Samples are grouped by type (e.g., QC, media, HepG2, HEK293T, blank) and sorted by sample name.

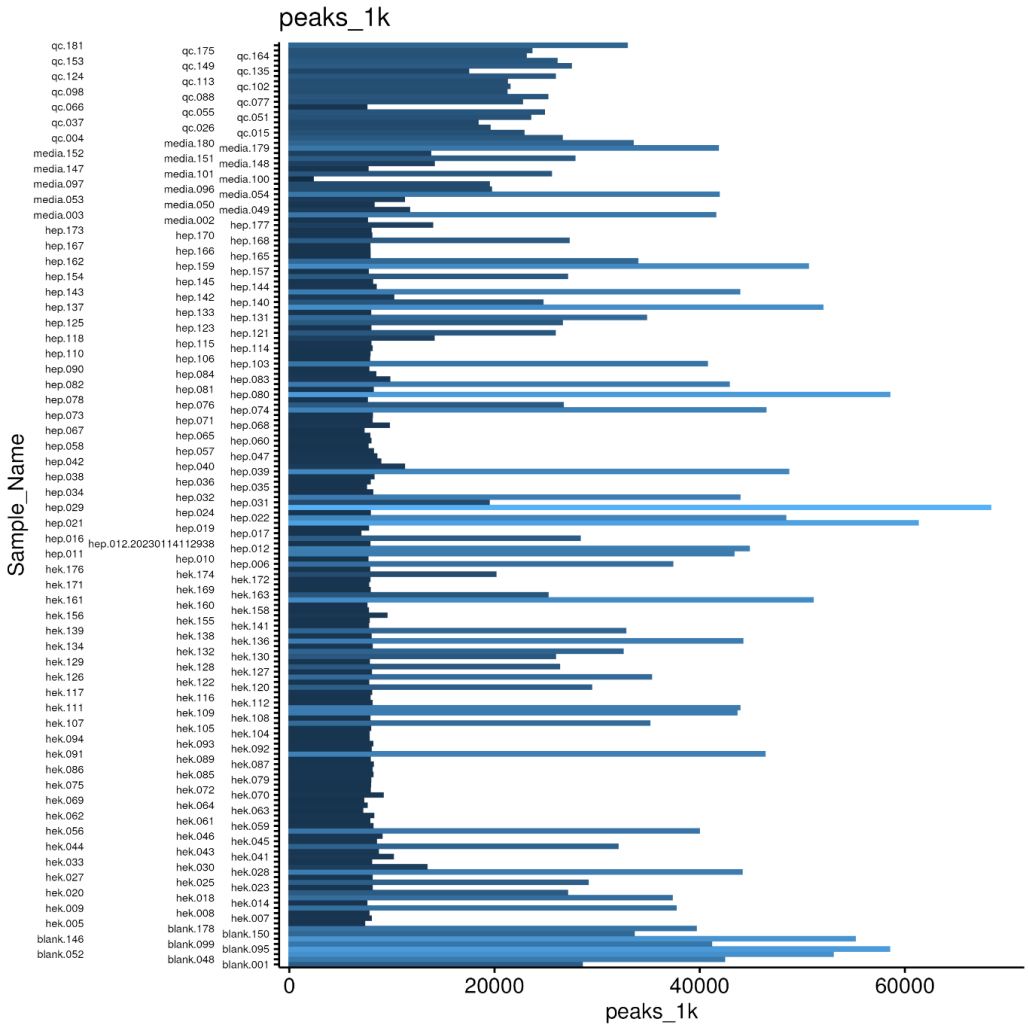


Figure S14. Number of detected peaks per sample (>1k).
Data generated output of the function *mzmetrics_quality_plot_all*. Each horizontal bar represents the number of peaks recorded in a given sample with an intensity great than 1000. Samples are grouped by type (e.g., QC, media, HepG2, HEK293T, blank) and sorted by sample name.

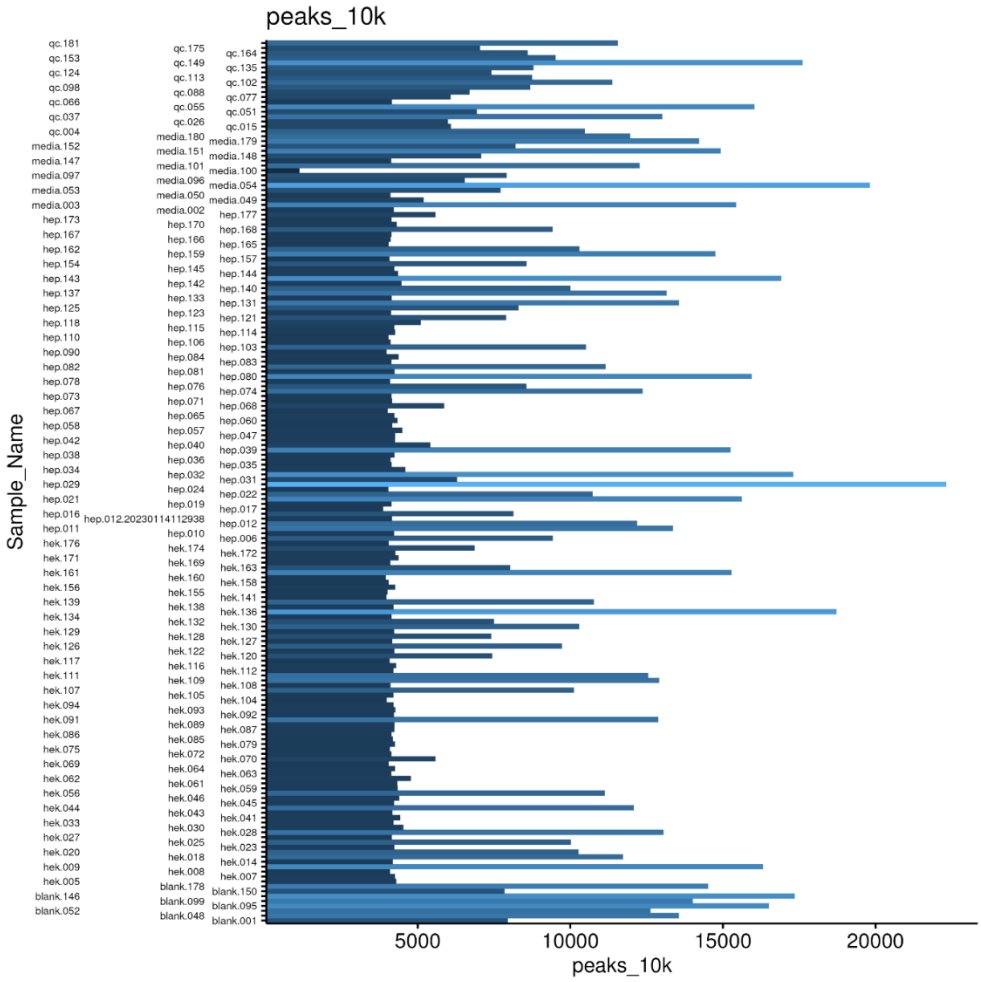


Figure S15. Number of detected peaks per sample (>10k).
Data generated output of the function *mzmetrics_quality_plot_all*. Each horizontal bar represents the number of peaks recorded in a given sample with an intensity great than 10000. Samples are grouped by type (e.g., QC, media, HepG2, HEK293T, blank) and sorted by sample name.

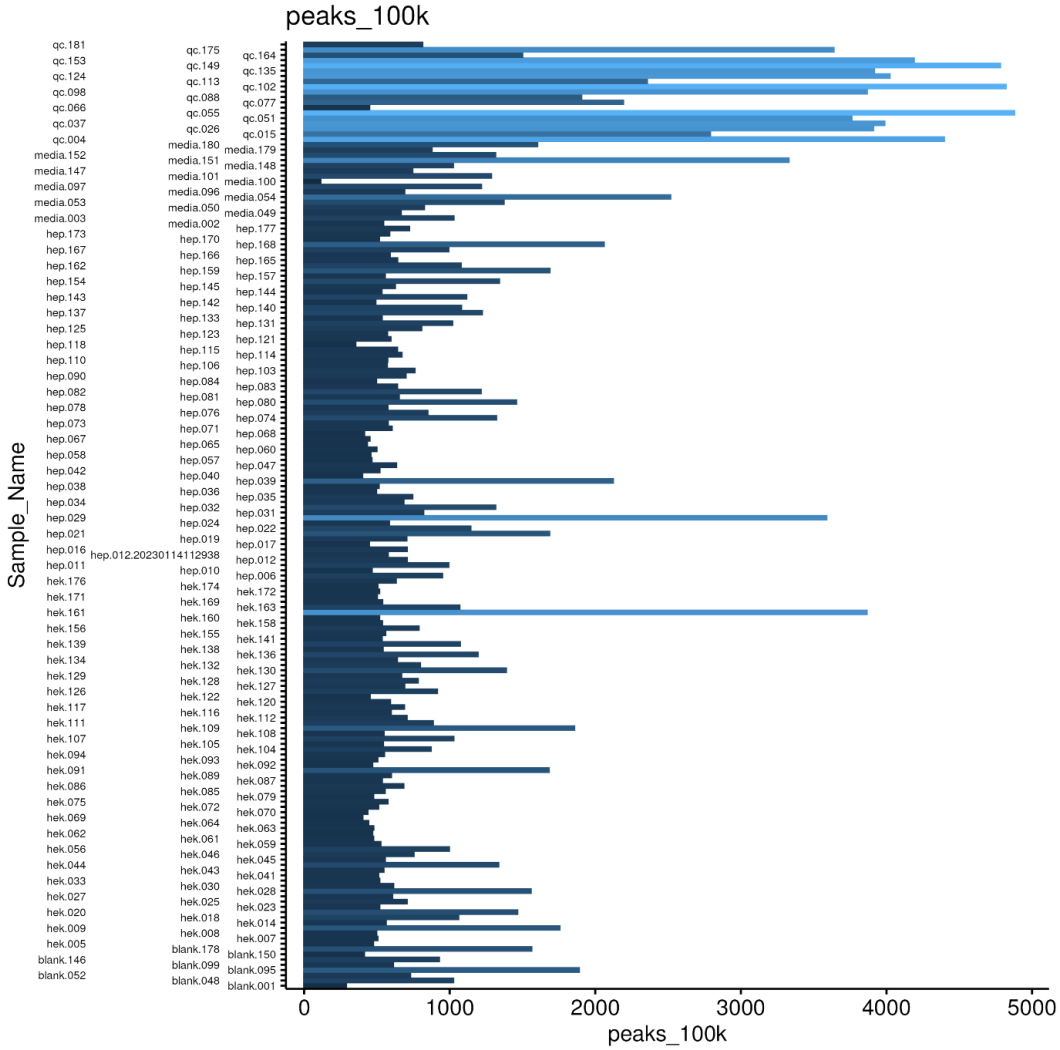


Figure S16. Number of detected peaks per sample (>100k).
Data generated output of the function *mzmetrics_quality_plot_all*. Each horizontal bar represents the number of peaks recorded in a given sample with an intensity great than 100000. Samples are grouped by type (e.g., QC, media, HepG2, HEK293T, blank) and sorted by sample name.

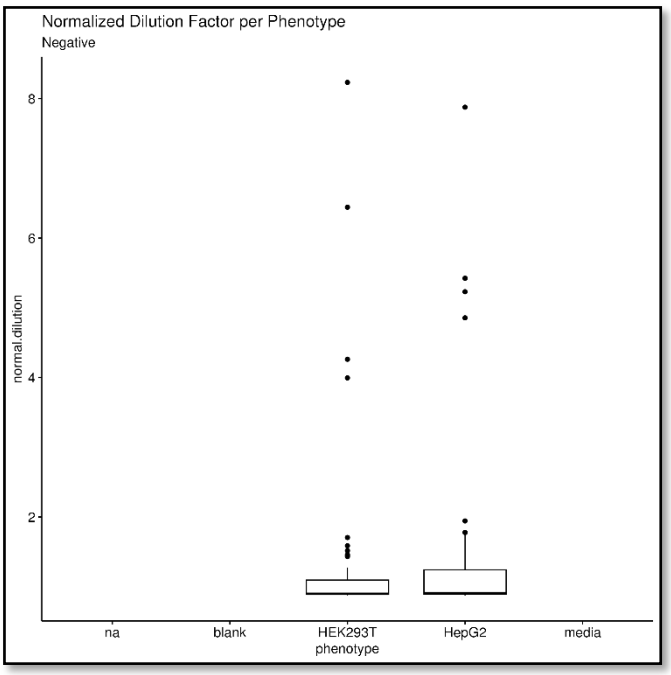


Figure S17. Normalized Dilution Factors by Phenotype.
Data generated output of the function *mz_post_normalization*. Normalized dilution values for each sample type in negative mode. The boxes show the middle range of the values, and the dots show individual samples.

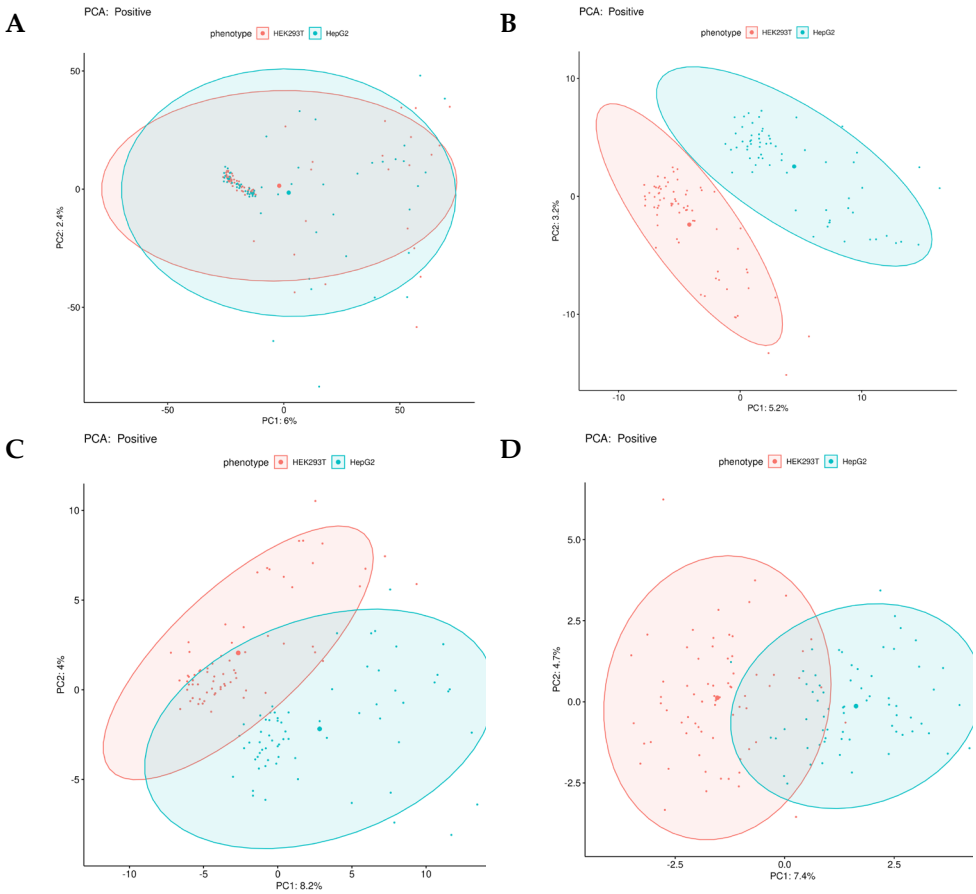


Figure S18. Principal component analysis (PCA) of HEK293T and HepG2 samples. Data generated output of the function *mzlog_analysis_pca*. A) PCA of the full data set. B) PCA of peaks identified as significant with Welch t-test. C) PCA of peaks identified as significant by the volcano plot (t-test and fold change). D) PCA of peaks identified as significant by random forest. The PCAs of the significant peaks reveal partial separation between the two phenotypes along PC1 and PC2 . Ellipses represent 95% confidence intervals for each group, with HepG2 consistently showing broader dispersion.

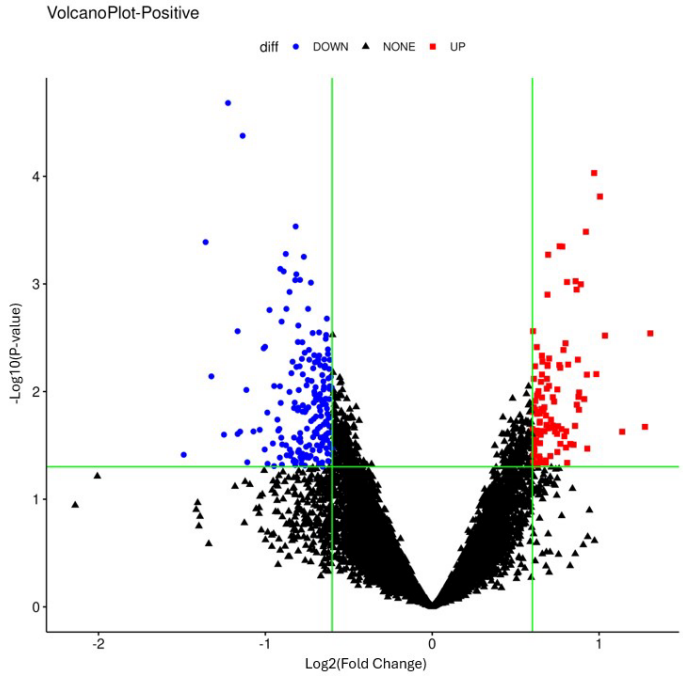


Figure S19. Volcano plot of fold change versus p-value in negative mode.
Data generated output of the function *mzlog_analysis_volcano_magic*. Each point represents a feature plotted by its log fold change and corresponding p-value. Vertical and horizontal green lines indicate fold and significance thresholds, respectively.

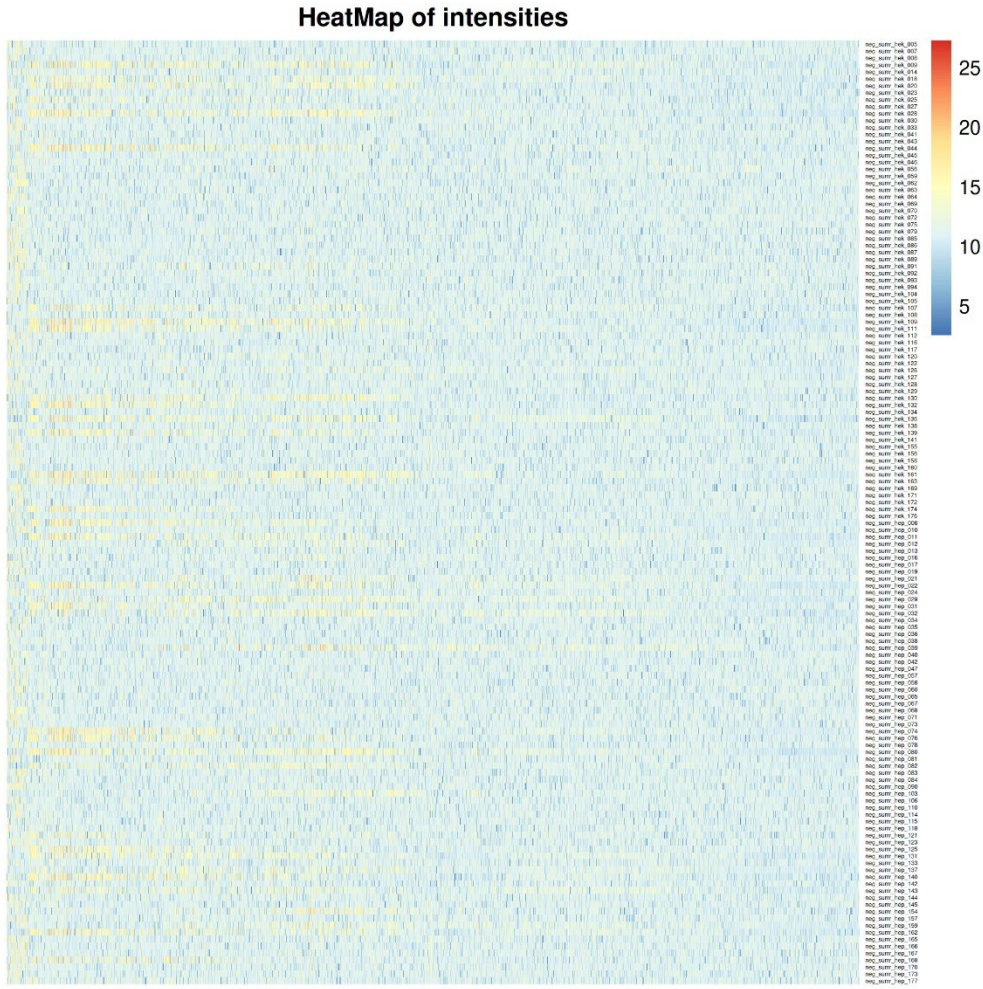


Figure S20. Heatmap of intensities across all samples.
Data generated output of the function *mzlog_analysis_heatmap*. Feature intensities are color-coded from low (blue) to high (red). Columns represent individual features, and rows represent samples, with sample names shown along the axes.

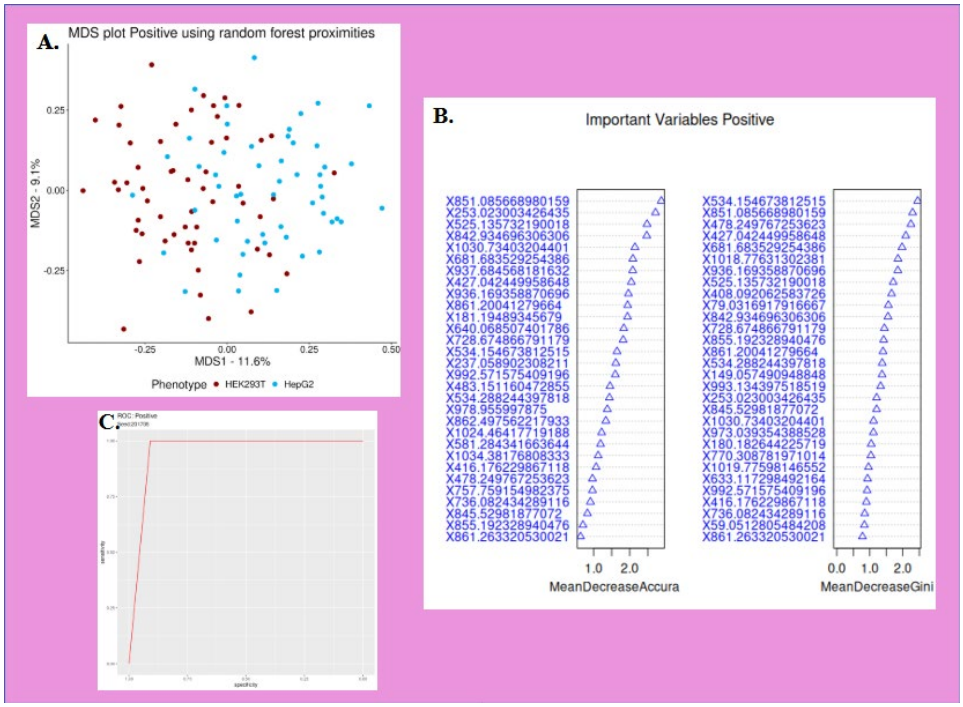


Figure S21. Phenotype-based sample distribution using multidimensional scaling
A. MDS data generated plot of random forest modelling. B. Data generated variables contributing to the difference between phenotypes, as determined by random forest. C. Data generated receiver operating characteristic (ROC) curve to assess the model's performance.