nature communications

Review article

https://doi.org/10.1038/s41467-025-63751-1

Best practices and tools in R and Python for statistical processing and visualization of lipidomics and metabolomics data

Received: 27 July 2024

Accepted: 25 August 2025

Published online: 30 September 2025

Check for updates

Jakub Idkowiak[1,2], Jonas Dehairs[2], Jana Schwarzerová[3,4,5], Dominika Olešová[6,7], Jacob X. M. Truong[8,9], Aleš Kvasnička[10,11], Marios Eftychiou[12,13,14,15], Ruben Cools[12,13,14], Xander Spotbeen[2], Robert Jirásko[1], Vullnet Veseli[1], Marco Giampà[16,17], Vincent de Laat[2], Lisa M. Butler[8,9], Wolfram Weckwerth[4,18], David Friedecký[10], Jonas Demeulemeester[12,13,14], Karel Hron[19], Johannes V. Swinnen[2] & Michal Holčapek[1]

Mass spectrometry-based lipidomics and metabolomics generate extensive data sets that, along with metadata such as clinical parameters, require specific data exploration skills to identify and visualize statistically significant trends and biologically relevant differences. Besides tailored methods developed by individual labs, a solid core of freely accessible tools exists for exploratory data analysis and visualization, which we have compiled here, including preparation of descriptive statistics, annotated box plots, hypothesis testing, volcano plots, lipid maps and fatty acyl chain plots, unsupervised and supervised dimensionality reduction, dendrograms, and heat maps. This review is intended for those who would like to develop their skills in data analysis and visualization using freely available R or Python solutions. Beginners are guided through a selection of R and Python libraries for producing publication-ready graphics without being overwhelmed by the code complexity. This manuscript, along with associated GitBook code repository containing step-by-step instructions, offers readers a comprehensive guide, encouraging the application of R and Python for robust and reproducible chemometric analysis of omics data.

Advances in mass spectrometry and chromatography have boosted the fields of biomedical and clinical lipidomics and metabolomics, with vast volumes of data being generated each day. The primary focus of biomedical/clinical lipidomics and metabolomics studies is to investigate the biological variation reflected by different lipid or metabolite levels between analyzed groups[1,2]. However, lipid or metabolite concentrations in biological materials can be influenced by other factors, including culture and growing conditions[3,4], age[5,6], sex[5–7], dietary habits[8,9], smoking and drinking status[6,10], medications[11,12], circadian rhythm[1,13–15], or other comorbidities[6,12,16–22]. Data also exhibit unwanted variation that can arise at multiple steps during the experiment[23–25]. Extracting the biological variation without applying the right procedures is complicated, leading to ambiguous or erroneous results. Therefore, investigators collaborating within the International Lipidomics Society and the Metabolomics Society have created guidelines for performing lipidomics and metabolomics experiments to improve the quality of quantitative omics data, unifying experimental protocols, and standardizing data reporting[26–32]. The variability of lipidomics and metabolomics data can be reduced by following these recommendations.

A full list of affiliations appears at the end of the paper. e-mail: Michal.Holcapek@upce.cz

A typical output of quantitative measurements is a table filled with lipid or metabolite concentrations measured across samples (observations). Usually, the number of variables/features (lipids or metabolites) quantified in biological materials exceeds the number of samples measured[33]. Lipidomics and metabolomics tables often contain missing values[34–38] and outliers[39]. As a result, the concentration distributions for biological groups often deviate from a symmetric Gaussian distribution, exhibiting left- or right-skewed patterns, with the latter being usually more common. Lipidomics and metabolomics data are also characterized by heteroscedasticity, which means that the spread of variable values within examined biological groups may not be comparable. Their concentrations can differ by orders of magnitude even within the same biological class of compounds. However, more abundant molecules may not necessarily be more important from the biological point of view. The magnitude of alterations in metabolite and lipid levels may also differ. Molecules involved in the tightly controlled central metabolism are less prone to changes than those in the secondary metabolism[40]. Concentrations of molecules from the same subclass, class, or closely related metabolic pathways are likely to be correlated[41]. Computed concentration values are affected by batch effects resulting from fluctuations in the instrument's response during the sample sequence. To address this, the standardized datasets contain additional quality control (QC) samples[2]. QCs can be obtained simply by pooling small aliquots of all biological samples[42] or purchased, e.g., National Institute of Standards and Technology (NIST) standard reference material (SRM) 1950[43] for metabolomics/lipidomics of plasma samples. Using QCs and blanks allows for evaluating the quality of the obtained data, provides insight into technical variability[42], and is instrumental for normalization (e.g., removal of batch effects)[2,44].

Analyzing these complex data, scientists must acquire statistical, computational, and data visualization skills to gain insights into statistically significant trends and relevant relationships hidden in their datasets, being aware of their specific properties. Advancing knowledge in statistics and programming is a demanding task with many hurdles, particularly when transitioning from a graphical user interface (GUI) to a text editor. Therefore, web-based, user-friendly tools have been developed to facilitate data exploration, e.g., the MetaboAnalyst platform[45], LipidSig[46], LipidSuite[47], LipidMaps Statistical Analysis Tool[48], LipidomicsR [http://www.lipidomicsr.top/], or COVAIN[49]. When using these platforms, the user is guided through a simple chemometric pipeline, from uploading datasets to extracting and visualizing the most significant information. User decisions are translated into code that ultimately triggers mathematical operations, simplifying the data mining. The novel Shiny app ADViSELipidomics has also been introduced, covering preprocessing, analyzing, and visualizing lipidomics data[50]. Although these solutions suit novices in statistics and chemometrics, more experienced users demand more flexibility, particularly in visualization. Complex lipidomics and metabolomics datasets can be visualized in various ways. Lipidomic data can be grouped based on common characteristics, such as lipid subclass, fatty acid composition, saturation, or a number of aggregate carbons. Generating informative figures can be facilitated by at least basic R or Python scripting skills.

This manuscript is structured in three parts: (i) data preparation for statistical analysis, (ii) an overview and critical review of key statistical methods and visualizations applied in lipidomics and metabolomics – to build a solid understanding of the analyses, (iii) a beginner's guide dedicated to those who want to use R and Python for statistical analysis and visualization of clinical lipidomics and metabolomics data. We also provide a GitBook code repository containing scripts and step-by-step notebooks to support the readers' first steps with R or Python.

Data preparation for statistical analysis

Missing values handling in lipidomics and metabolomics

Missing values (NA, NaN) occur commonly in lipidomics and metabolomics datasets and often arise from issues with peak picking, integration, and alignment or analytical factors, including, for instance, matrix effects or lipid/metabolite abundance below the detection limit (LOD), etc.[25]. Although there are statistical methods robust to datasets with missing values, the general practice in lipidomics and metabolomics involves the imputation of missing entries. The best practice, however, involves investigating the reasons behind missing values[25] and remeasuring or reprocessing the datasets, if necessary, as randomly and frequently occurring missing entries can indicate potential issues with data acquisition and/or processing methods.

Missing values in -omics data can be divided into three categories: missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). This classification relies on two types of variables, i.e., observed variables (measurements present in a dataset) and unobserved variables (measurements absent in a dataset). In the case of MCAR, the missing values are independent of the observed or unobserved variables, i.e., the absence of values is unrelated to biological or technical factors[35,36,51,52], and it is the effect of a purely random event, e.g., vials containing a few random samples were broken during extraction. In the case of MAR, the missing values can be connected to the observed data[35,36,52,53], e.g., ion suppression of co-eluting signals of analytes[36,52]. MNAR are linked to unobserved values (so the values themselves), i.e., some lipid species or metabolites are below the limit of the detection (LOD) in biological material using a particular chromatography and/or mass spectrometry method settings (so-called left-censored values)[35,36,51–53].

Different strategies have been examined to address MCAR, MAR, and MNAR in -omics data[25,34–38,51–53]. Among the most often listed are imputation by a constant value (e.g., a percentage of the lowest concentration measured, a mean or median value for a lipid/metabolite, etc.) (i), imputation by k-nearest neighbors (kNN) and its variants (ii), or random forest (iii)[25,35–38,51–53]. Replacing NA values should not lead to substantial changes to the information stored in the data. Due to the varying nature of missing values (MCAR, MAR, and MNAR), a single imputation method might often be insufficient, especially when MCAR (or MAR) and MNAR coexist in a dataset, leading to better imputation for one type over the other[35,51]. Additionally, MNAR is usually more challenging to impute[35,38,51], as relying on observed values to replace the unobserved ones can introduce additional bias[51]. Columns with lipids/metabolites with predominantly missing values are usually filtered out before statistical analysis with a defined threshold (e.g., >35% of concentrations missing)[25]. A recent study by Frölich et al. showed that kNN-based methods can be used to impute both MCAR and MNAR in shotgun lipidomics data, performing better than random forest-based imputation for MNAR[35]. Earlier, Do et al. and Armitage et al. also recommended kNN-based imputation for replacing NA values in metabolomics data[36,37]. Furthermore, Kokla et al. demonstrated that random forest was the most effective imputation method for LC/MS metabolomics data, with kNN-based imputation ranking just after it in performance[38]. Wei et al. demonstrated that the random forest method performed best for MCAR/MAR and quantile regression imputation of left-censored data (QRILC) for MNAR in the case of metabolomics data[34]. In lipidomics, MNAR data are often imputed using a percentage of the lowest concentration for a lipid, which Frölich et al. also identified as an optimal method when testing the half-minimum (hm) imputation approach[35].

More information and strategies for dealing with missing values can also be found in the GitBook, including substitution by the percentage of the lowest concentration, a constant value, mean, median, kNN, and random forest model.

Data normalization and preprocessing

Generally, in all -omics sciences, data normalization aims to alleviate all the unwanted sources of variation, so the spotlight in the final dataset is only on the biological information of interest. Thus, data normalization and removing unwanted variation are often used interchangeably[25,54]. However, as noticed earlier by Olshansky et al. in modern lipidomics and metabolomics heavily relying on data standardization, data normalization usually addresses analytical variation, i.e., batch corrections and signal intensities (areas) recalculation against specific analytical standards into concentrations[25]. For sample amount normalization, pre-acquisition methods are preferred in metabolomics and lipidomics. Typically, sample aliquots are normalized based on volume, mass, cell count, protein amount, DNA amount, or metabolite concentration (e.g., creatinine in the case of urine)[55,56]. Further, several statistical post-acquisition normalizations exist, e.g., sum, median, probabilistic quotient normalization (PQN), maximal density fold change (MDFC), quantile, class-specific quantile, etc.[57]. The resulting lipid/metabolite concentrations are often further preprocessed based on the assumptions of the statistical or machine-learning methods used for data analysis. This primarily involves data transformation (to stabilize the mean or improve data interpretability) and scaling (finding a uniform scale for all variables)[40]. Although analytical variation is typically well controlled, and specific post-acquisition normalization methods can mitigate or correct datasets affected by improper or missing pre-acquisition normalization, pre-analytical variation is often difficult to account for. For example, in most lipidomics and metabolomics studies, no control features are in place to account for unwanted variation introduced during sample collection, storage, and handling before the sample preparation step and the addition of standards. Any unwanted alterations resulting from ongoing metabolism can influence various lipid and metabolite classes differently, complicating the control feature selection process[25]. Finally, one should also be aware of different sources of biological variation. Some may interfere with the variation of interest and should be accounted for before comparing lipidomes/metabolomes[54].

Normalizations to standards in lipidomics and metabolomics, as well as pre- and post-acquisition sample normalizations, have been the subject of extensive reviews, comments, recommendations, and research articles, e.g.[26,29,31,55–60]. Here, we briefly focus on batch corrections and data preprocessing through transformation and scaling.

Batch correction in metabolomics and lipidomics. Correcting batch effects is critical to minimize unwanted variability that could interfere with the biological differences. While batch effects are briefly mentioned here, a more in-depth discussion can be found in our GitBook, where QC-based algorithms such as LOESS (Locally Estimated Scatterplot Smoothing) and SERRF (Systematic Error Removal using Random Forest)[61] are presented. These algorithms leverage QC sample data to correct for systematic measurement biases, including time-dependent drifts caused by changes in chromatography or mass spectrometry conditions. Additionally, randomizing sample run orders is recommended to minimize the impact of batch effects. Readers are encouraged to refer to the GitBook for detailed guidance on implementing these approaches, including practical considerations for quality control and batch correction.

Data transformation and scaling. Data transformation and scaling are an essential omics data analysis step, and the researcher should be aware of their purpose and impact. As an integral part of data preprocessing, one should examine the effect of data preprocessing on both the data structure - such as by comparing histograms or density plots before and after preprocessing - and the interpretability, i.e., if one still can derive meaningful insights after the transformation. Typically, it is not recommended to apply more than one data transformation or scaling method to a dataset.

Data transformation usually occurs before scaling. The need for transformation depends on the structure of the data and the goals of the analysis. Transformation often addresses skewness, heteroscedasticity (variance differences between groups or conditions), extreme outliers, and non-linear relationships. Transformation helps conform data more closely to the assumptions of certain algorithms, taking into account factors like distribution, variance, and linearity. Software and online platforms for analyzing -omics data often offer logarithmic, square root, and cube root transformations. The log and square root transformations can minimize the frequently occurring right-skewness in lipidomics and metabolomics. Log transformation also converts multiplicative relationships into additive ones, simplifying the interpretation of results, e.g., when comparing lipid abundances among different groups (fold change analysis). The cube root transformation, used for negative predictor values, is rarely employed in lipidomics and metabolomics. Figure 1 – A illustrates the selected effects of applying the log transformation to the data. However, other transformation methods that aim to do the same job exist, some of which can also cope with different size volume and/or sample concentration (i.e., the size effect), such as the family of log-ratio transformations[62] or probabilistic quotient normalization[63].

After transformation, the data are rescaled to ensure that all values are on the same scale (as shown in Fig. 1B). Auto- or Pareto-scaling methods are often used in lipidomics and metabolomics. A key step of both is mean-centering, which involves subtracting the column mean value from each entry. This process streamlines data interpretation, shifting the focus towards differences[40], i.e., a mean-centered score close to 0 indicates outcomes near the average, a score above 0 represents an above average level, and a score below 0 signifies a below average level. Mean centering does not influence standard deviation or variance. Hence, in the next step, every outcome is related to the standard deviation (Auto-scaling) or the square root of the standard deviation (Pareto-scaling). Frequently selected Auto-scaling adjusts each lipid or metabolite column so that the standard deviation (or variance) is 1 and the mean is 0. The resulting score for every entry indicates how many standard deviations a particular value is above or below the mean, with 0 representing the average, providing a clearer view of the relative spread. Such scaling is crucial for statistical methods sensitive to the variance of variables (lipids or metabolites with larger variances can dominate the analysis). A good example is principal component analysis (PCA), discussed in the following section.

Given the importance of data transformation and scaling, we provide more detailed information in the first section of the GitBook.

Overview of key statistical methods and visualizations applied in lipidomics and metabolomics

Methods for data exploration

Methods for data exploration can be characterized as univariate (considering one variable at a time) or multivariate (examining multiple variables simultaneously)[64]. Data exploration begins with the preparation of descriptive statistics. Although univariate methods can also be used in this step, it is important to keep in mind that omics data are, in essence, multivariate.

Descriptive statistics. Descriptive/Summary statistics summarize the basic properties of the dataset. At this step, measures of central tendency are estimated, so-called location parameters, which refer to a typical lipid or metabolite concentration value for each biological group, a center of each distribution. Depending on the shape of a distribution, the most typical value within a biological group can be reflected by the mean (symmetric distributions only) or the median and the mode (better for

https://doi.org/10.1038/s41467-025-63751-1

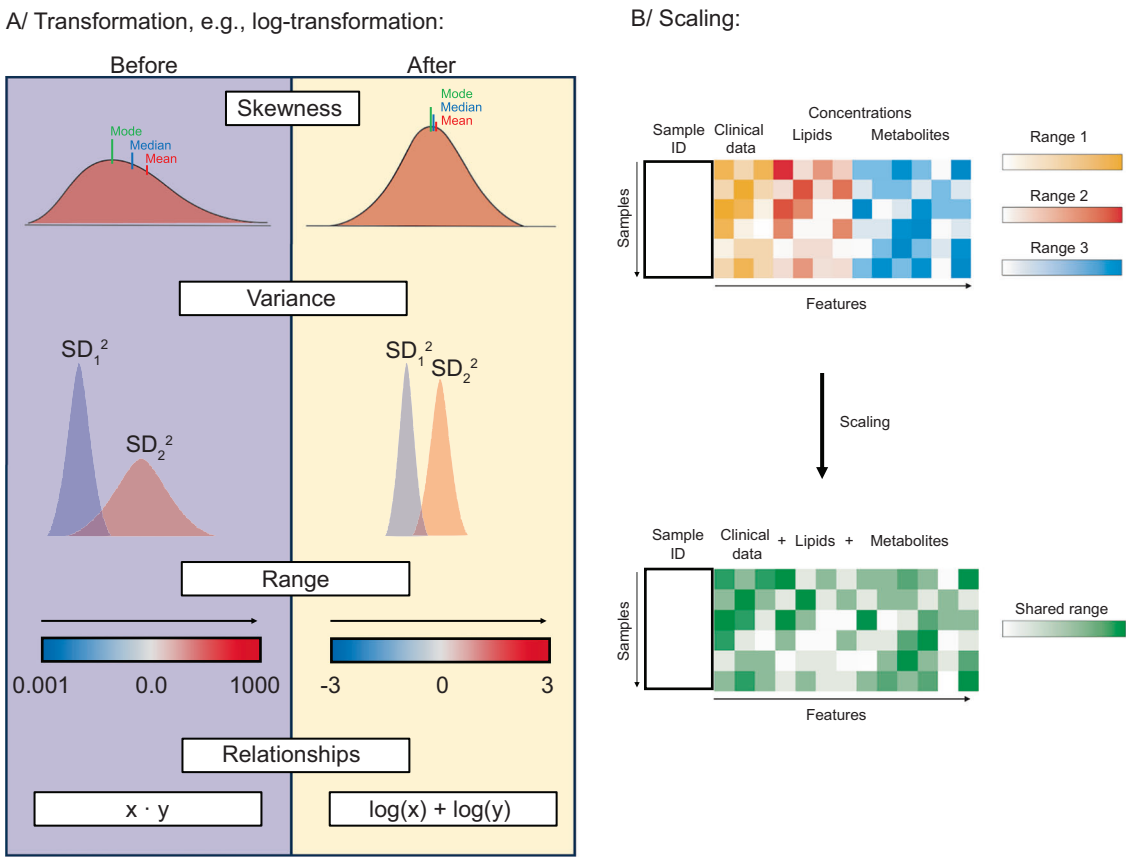


Fig. 1 | Data transformation and scaling. A Possible consequences of the data transformation based on the log transformation (for more information, see GitBook). **B** Simplified visualization of the effect of scaling on the dataset range.

skewed distributions) (Fig. 2)[65–67]. The typical value representing a biological group is usually presented together with a measure of dispersion, for example, standard deviation (SD), variance, range, interquartile ranges (IQR)[65,67], or the coefficient of variation (SD relative to mean)[67,68]. Quartiles (25% percentiles) split the data into four equal parts (*Q1-4*) after listing them in ascending order. Deciles (10% percentiles - *p10*, *p20*, *p30*, etc.) are computed similarly to quartiles and split data into ten equal parts[67]. Summary statistics can also contain information on contingency tables (for presence-absence analysis) and parameters characterizing the distribution shape, like skewness and kurtosis[67]. This analysis allows the detection of potential outliers and implies sample distribution properties. The initial investigation of relationships among variable concentrations is also a part of summary statistics. Here, covariance can be applied to indicate how and in what direction concentrations of two lipids or metabolites change together. Correlation analysis is often performed in lipidomics and metabolomics, which measures the direction and strength of a relationship. Correlation ranges between −1 and 1, indicating a strong negative or positive linear relationship, respectively, while a correlation close to 0 indicates no linear relationship exists between two concentrations[67,69]. The Pearson correlation can be calculated for normally distributed samples of populations, but it is sensitive to outliers. Instead, Spearman's rank correlation should be used, also for skewed distributions[67,70].

Graphical representation of descriptive statistics using box plots. Box plots co-plotted with dot plots thoroughly and unambiguously depict individual lipid or metabolite distributions, while other types of plots can be used for accompanying or extending descriptive statistics (see GitBook). A typical Tukey box plot is shown in Fig. 2A. This plot presents outlying values above fences (defined as $Q3 + 1.5 \cdot IQR$ and $Q1 - 1.5 \cdot IQR$ for upper and lower, respectively), minimum and maximum values (highest and lowest values before the upper and lower fence - corresponding to whiskers' length), first and third quartiles, and median[71,72]. The classic Tukey box plot accurately depicts samples characterized by symmetric, normal-like distributions. However, skewed distributions are better captured using adjusted box plots (Fig. 2B), which are not commonly used in lipidomics and metabolomics, even though they are available in R packages like *robustbase*[71] or *litteR*[73]. Box plots for skewed distributions redefine the lengths of whiskers (upper and lower fences) to use a robust statistic that measures skewness, known as the medcouple (*MC*) – a scaled median difference between the values of the left and right half of distribution. For $MC \geq 0$ (positive skewness), the box plot model is defined as $Q3 + 1.5e^{3MC}IQR$ (upper fence) and $Q1 - 1.5e^{-4MC}IQR$ (lower fence), while for $MC < 0$ (negative skewness) as $Q3 + 1.5e^{4MC}IQR$ (upper fence) and $Q1 - 1.5e^{-3MC}IQR$ (lower fence) (Fig. 2C, D). For data following the symmetric distribution (e.g., Gaussian), $MC = 0$, and the adjusted boxplot reverts to the standard Tukey box plot (Fig. 2D)[71]. Box plots are frequently co-plotted with dots corresponding to every data point to give better insight into the sample distribution. To keep the figure informative and reflective of the actual data, transparency, color, or fill color settings can be adjusted, and horizontal jitter added to avoid overplotting (Fig. 2D). In the case of paired samples (e.g., the same subjects measured twice – see Fig. 2E), data points corresponding to the same patient can be connected by lines for clarity. An inventive solution is a hybrid of a density plot with a box plot, known as a violin plot[74], which preserves even more of the data structure (Fig. 2E, plot on the right). The accompanying GitBook presents a variety of box plots along with the corresponding R/Python scripts.

https://doi.org/10.1038/s41467-025-63751-1

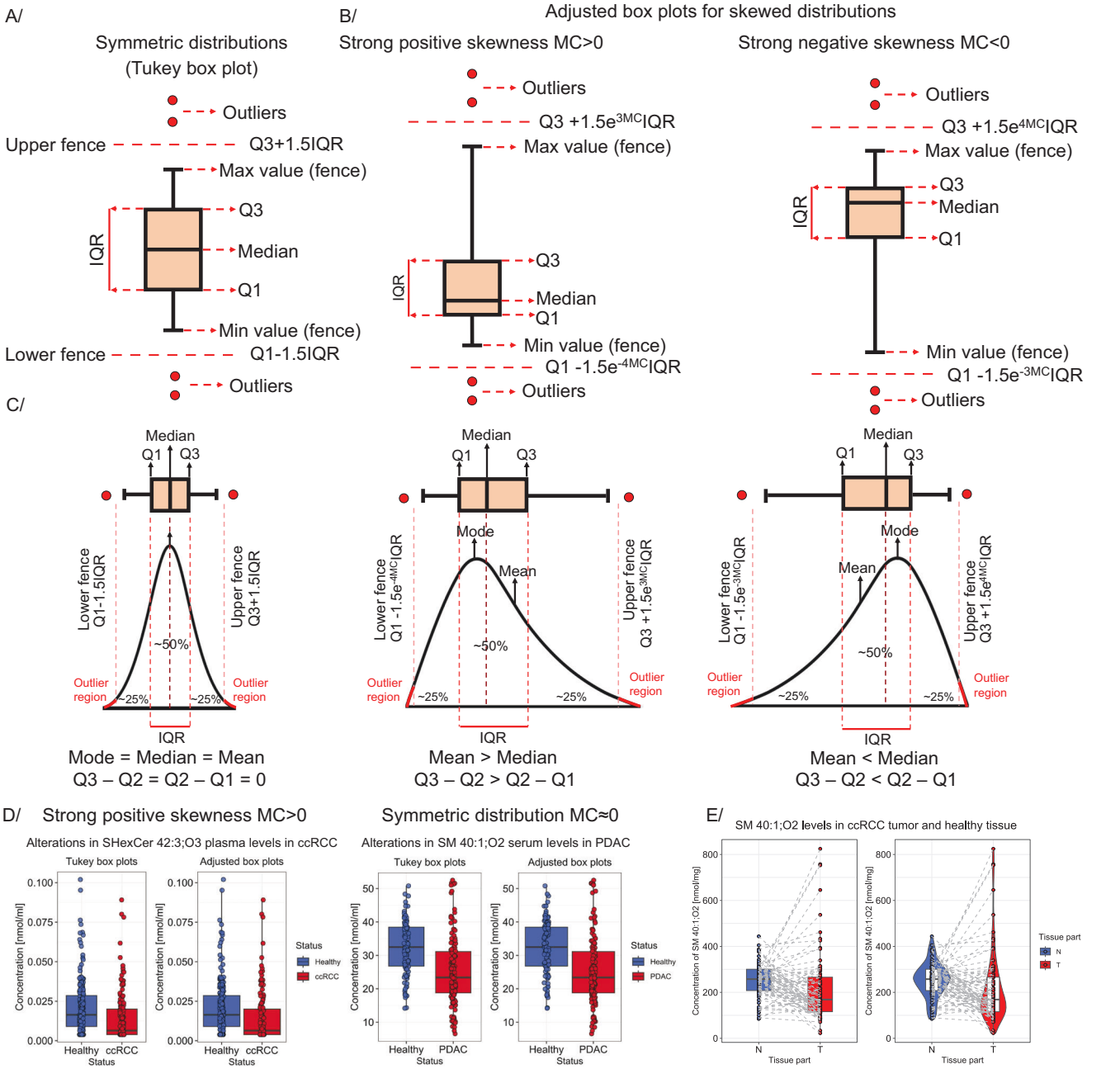


Fig. 2 | Components and construction of box plots. Construction of (**A**) the classic Tukey box plot for symmetric distributions and (**B**) adjusted box plots for skewed distributions. Traditionally, box plot whiskers extend to minimum and maximum values before the fence border. Points above fences are considered outliers (global minimum/maximum values representing the complete concentration range). **C** Relationship between sample distribution and box plot shape. **D** Comparison of the number of outliers as visualized in Tukey and adjusted box plots for data with positive skewness (on the left side – ccRCC study[125]) and symmetric distributions (based on healthy controls – on the right side – PDAC study[126]). **E** Box plots for paired observations – with lines connecting paired samples. On the right side - a violin box plot is presented.

Univariate statistical methods

Fold change. After measures of central tendency are calculated for each group, a fold change can be computed. Fold change is the ratio of two mean concentrations (usually presented as $\log_2$ or $\log_{10}$ value) of lipids/metabolites in a condition related to the control condition. The ratios of medians or modes can be used for skewed distributions.

Statistical tests for comparing two biological groups. Statistical tests are broadly applied to compare outcomes between biological groups, e.g., differences in mean concentrations of lipids/metabolites in the control and disease groups. The tests can be parametric or non-parametric. The former covers all tests that make assumptions about the distribution from which the sample data is drawn. The latter can be considered distribution-free tests and are not restricted by assumptions on the nature of the sampled population[42,69].

The *t*-test is a parametric test to compare the location (i.e., mean) of two random samples of continuous variables. If two samples are independent, the unpaired *t*-test is used. In contrast, if measurements involve, e.g., subjects pre- and post-intervention, a paired *t*-test is applied[42,69,75]. While the latter assumes that differences between pairs of values are approximately normally distributed, the former requires

https://doi.org/10.1038/s41467-025-63751-1

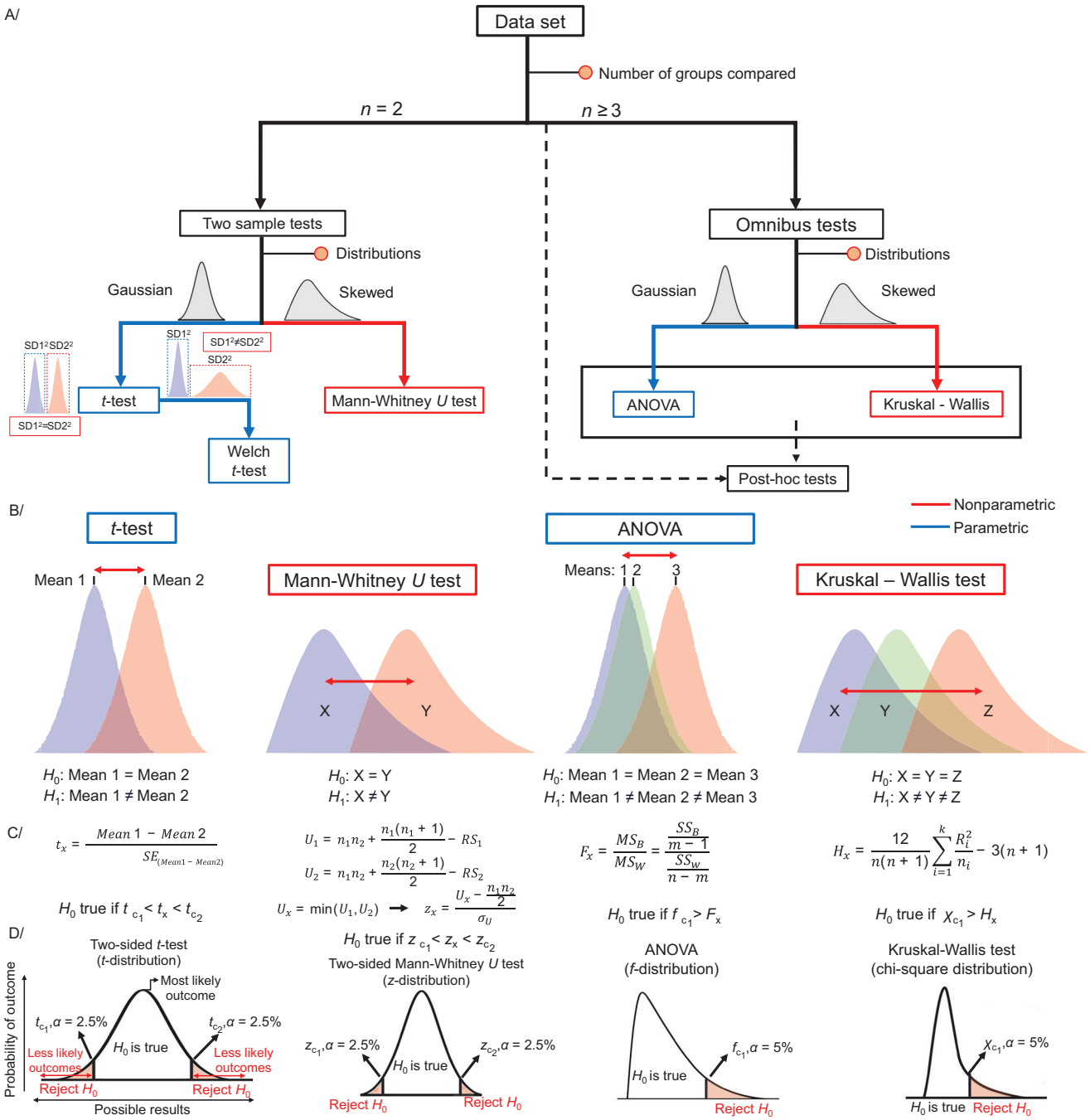


Fig. 3 | Statistical tests commonly used in lipidomics and metabolomics. **A** Roadmap for selecting basic common statistical tests and B their null and alternative hypotheses (H_0 and H_1). **C**, **D** Definition of the test statistics and conditions for rejecting H_0. Figure prepared based on information from B. Rosner's Fundamentals of Biostatistics[69].

the sampled concentration of both variables to come from the normal distribution with the same spread[69,75] (variance, testable using an *F*-test, for example) (Fig. 3A). Welch's *t*-test is used if the variances of the two groups are different (Fig. 3A)[70,75]. The *t*-statistic determines the test outcome, i.e., whether to reject or not the null hypothesis. This statistic is a scaled difference between the sample-estimated means, which, under the null hypothesis, follows a Student's *t*-distribution (akin to the normal distribution but with heavier tails; Fig. 3C, D – example 1). The null hypothesis can be that one mean is greater (or smaller) than the other (one-sided test – only the probability mass in one tail of the *t*-distribution is assessed) or test whether either is true (i.e., the means are not equal; two-sided test – both tails are considered) (Fig. 3D – example 1). Importantly, at the same significance level α, the one-sided test is more sensitive[69]. However, as the direction of differences is often unknown, the two-sided *t*-test is generally more suitable for metabolomics and lipidomics.

The appropriateness of the *t*-test for lipidomics and metabolomics data is debatable. In general, if its assumptions are met, it is more powerful than its non-parametric counterparts[76]. However, its results may no longer be robust if assumptions are violated, typically in the form of heteroscedasticity (presence of outliers or skewing) or unequal, small sample sizes[42] (e.g., <30). Although the use of non-parametric tests in medical studies has increased, they are not always necessary. When sample sizes are large (e.g., ≥200 observations per

group), the *t*-test is robust even for highly skewed data[76]. In turn, the application of non-parametric tests is almost always reasonable. Although non-parametric tests are less powerful to pick up an effect, the null hypothesis is rarely falsely rejected.

A distribution-free test, such as the Mann–Whitney *U* test (also known as Wilcoxon rank-sum test), is typically a superior option in the presence of outliers[42] (Fig. 3A). The Mann–Whitney *U* test assesses the null hypothesis that two collected samples come from the same distribution (Fig. 3B). In the first steps of the Mann–Whitney *U* test, all observations are pooled and sorted, and ranks are assigned. Ties (e.g., identical concentrations) are resolved by re-assigning the average of the initially assigned ranks to tied values[67]. Then, the sums of ranks are calculated (RS_1, RS_2) and used to compute the *U*-statistic for both groups. The smallest *U* value is used for the two-sided Mann–Whitney test (U_x) (Fig. 3C). Similarly to the *t*-test, the two-sided test is more suitable for lipidomics and metabolomics data, hence, only this case is discussed further. The *p*-value can be obtained based on *Z*-statistics (Fig. 3C, D), as *U* can be approximated to the normal distribution. *Z*-statistics can be computed based on U_x, the expected value of *U* ($\frac{n_1 \cdot n_2}{2}$), and the standard error of *U* (σ_U or σ_{Ucorr} for ties) (Fig. 3C, D). The null hypothesis is accepted or rejected based on the critical *z*-values (z_c)[67].

Statistical tests for comparing three or more groups. If the study involves multiple groups, e.g., healthy volunteers and patients split into different disease stages, the omnibus test is computed for comparing all groups simultaneously. A single omnibus test allows for maintaining type I error (i.e., incorrect rejection of the null hypothesis) at the significance level $\alpha = 0.05$, in contrast to performing repeated *t*-tests.

The parametric extension of the classical *t*-test for comparing multiple groups is the one-way analysis of variance (ANOVA). The one-way ANOVA tests the influence of one categorical variable (e.g., health status) on one continuous variable (e.g., lipid concentration). Similarly to the *t*-test, ANOVA assumes that all samples are drawn from populations with at least a symmetric distribution and approximately similar variances and that samples in each group are independent[69,77]. ANOVA relies on estimation of the so-called *F*-ratio. Assume *m* is the total number of groups and *n* the total number of data points in the dataset. Initially, means for all groups and the overall (grand) mean are computed. The variability between groups is then calculated as the sum of squared differences (SS_B) between the group mean and the grand mean, weighted by the number of observations in each group. The variability within groups is the sum of squared differences (SS_W) between every data point and its respective group mean. The total variability (SS_T) can then be described as the sum of SS_B and SS_W. The mean sum of squares between groups (MS_B), i.e., variance, is simply SS_B divided by the degrees of freedom for groups, i.e., $m - 1$. SS_W divided by the difference of the total number of data points and the total number of groups ($n - m$) is the mean sum of squares within groups (MS_W). Finally, F_x is the ratio of the mean sum of squares between groups (MS_B) vs. within groups (MS_W) or variance between groups to variance within groups (Fig. 3C, example − 3)[67,69,78]. Under the null hypothesis (no difference in means), the *F*-ratio follows *F*-distribution (Fig. 3D, example − 3)[69]. If the obtained F_x is higher than a critical value at $\alpha = 0.05$ (f_c), the variation between groups is higher than the variation within the groups, and the null hypothesis is rejected, indicating that at least one mean differs significantly from the others. Next, post hoc tests can be performed to find which means differ[77,79]. Many post hoc tests are similar to classic *t*-tests in their mathematical assumptions, computations, and null hypothesis (no difference in means). However, applying multiple tests (one comparison per pair of groups) again increases the chance of type I error occurrence. Therefore, most post hoc tests contain a correction, e.g., a more conservative cut-off compared to the *t*-test, which allows for maintaining the initial level of significance (e.g., 0.05)[79]. Tukey's Honest Significant Difference (HSD) post hoc test is frequently used and efficiently controls type I errors even without ANOVA. Hence, if a research question concerns multiple comparisons solely, post hoc tests can also be applied directly.

A non-parametric Kruskal-Wallis test can be applied if the assumptions of ANOVA are not met, particularly regarding the distribution of samples[67,69]. As an extension of the Mann–Whitney *U* test, it requires three or more independent groups and a continuous dependent variable (the measured variable, e.g., concentration of a lipid or metabolite). The Kruskal-Wallis test examines if the rank sums of the observations differ between groups[69,80]. If the shape and scale of the sampled distributions are similar, the null hypothesis can be simplified to equality between group medians. In the Kruskal-Wallis test, ranks are assigned to all dependent variables ordered from the smallest to largest, ignoring groups. Then, a rank sum is calculated for each *i*-th group (R_i) along with the mean rank sum ($\frac{R_i}{n_i}$, with n_i the number of subjects in the group). Based on the mean rank sum, the total number of observations in all groups (*n*), and the total number of groups (*k*), the test statistic H_x is computed according to the formula in Fig. 3C, D (last example)[67,69,80]. For large sample sizes, it is possible to rely on the approximation of *H* to $\chi 2$ distribution (chi-squared distribution). The null hypothesis is rejected if the test statistic exceeds the critical $\chi 2$ value ($p < \alpha = 0.05$)[69]. Dunn, Nemenyi, or Conover posthoc tests can be used afterward[81].

Graphical representation of statistical tests. The resulting fold changes and test statistics for a set of molecules can be visualized using a volcano plot (see GitBook or Fig. S3A). This typically plots a log-transformed measure of effect size (e.g., log fold change) on the *x*-axis and a negative log-transformed measure of statistical significance on the *y*-axis (usually $-\log_{10}(p)$). This way, the most strongly dysregulated features between the two groups appear in the top-left and top-right corners of the plot. They are often labeled, allowing effective presentation of large omics data distributions and highlighting of essential results.

Additional box plots, bar charts, or dot plots are often generated for the most interesting variables with *p*-values indicated explicitly or implicitly via a variable number of symbols (e.g., asterisks) above bars, box plots, and dots (see GitBook or see below).

Lipid maps and fatty acyl-chain plots. Although lipidomic data are complex, it is possible to specifically visualize their biochemical and structural associations. The specificity of lipidomic data, when compared to other omics techniques, lies in the availability of structural information in the names of lipid molecules. Typically, the lipid name contains details about the specific lipid subclass and the composition of fatty acyl chains within its molecule. This information, when coupled with statistical analysis, can serve as the foundation for visualization approaches based on lipid structural information and classification. One option is to visualize systematic changes in the entire lipid classes using lipid networks (sometimes called lipid maps). It is then possible to further focus on structural aspects of lipids, such as carbon chain lengths and the number of double bonds in their fatty acyl chains.

As for lipid networks, these are usually constructed using Cytoscape[82] [https://cytoscape.org/], where each lipid is assigned a node and an edge identifier, where edges connect the nodes. Central nodes usually represent lipid classes and subclasses, which are connected to individual lipid nodes by edges. The nodes can be plotted to reflect different statistical variables, usually differences are color-encoded (effect size like fold change or Cohen's *D*), while another plotting parameter can reflect the magnitude (*p*-value of different statistical tests, area under the curve (AUC)). With this visualization, it is possible to observe systematic changes in entire lipid classes and subclasses, which are often more important than changes in a single

https://doi.org/10.1038/s41467-025-63751-1

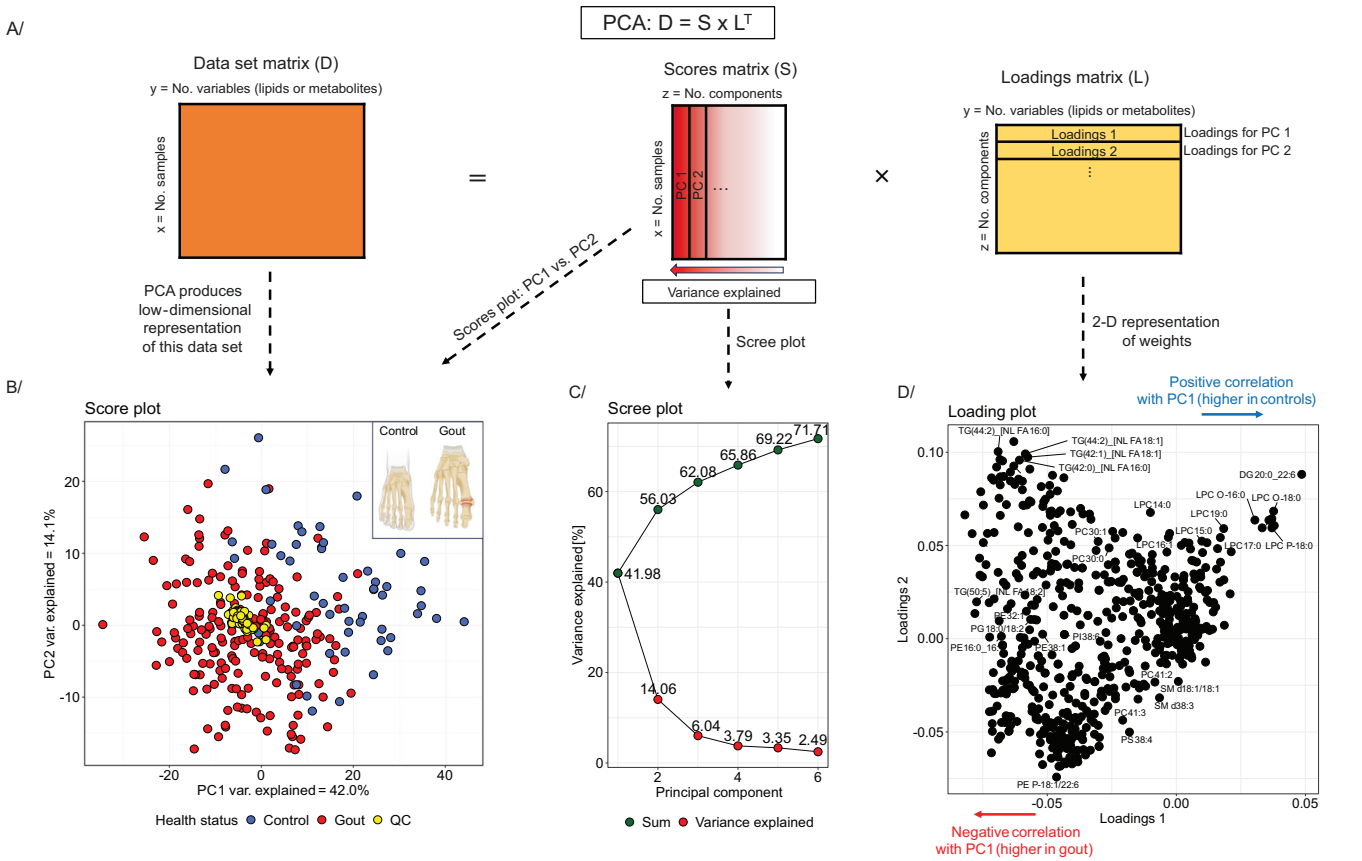


Fig. 4 | Conceptual overview of principal component analysis. A Principal Component Analysis – decomposition of a data matrix into a score matrix and a loading matrix. **B** Score plot presents an example PCA for a dataset composed of lipid concentrations analyzed via the liquid chromatography-mass spectrometry (LC-MS) in plasma samples from patients with gout vs. healthy controls (gout – red dots, healthy controls – blue dots, QC samples (data integrity) – yellow dots). Created in BioRender. Dehairs, J. (2025) https://BioRender.com/hcmf66r. **C** Scree plot representing the (cumulative) variance explained by each component. **D** Loading plot for the example dataset. The data sethas been published by Kvasnička et al. in their manuscript *Alterations in lipidome profiles distinguish early-onset hyperuricemia, gout, and the effect of urate-lowering treatment*[127].

individual lipid. Examples of this visualization can be found in Fig. S3B and in the GitBook, which also includes references to articles.

Another type of visualization is a fatty acyl chain structure plot, where the *x*- and *y*-axes reflect, respectively, the number of carbons and double bonds in fatty acyl chains. These plots are used to visualize the structural composition of lipid classes, which is essential since lipids typically exhibit class-specific patterns in living organisms. Specifically, within a class, either the entire class adheres to a specific pattern, or there is a noticeable trend toward specific lengths/saturation of lipids. Given the potentially large number of lipids within a class, these patterns may not be apparent through the inspection of individual lipids alone or using lipid networks. In this context, the visualization of fatty acyl chain structure plots proves helpful, providing a rapid overview of the structure of the altered lipids and their trends. Examples of this visualization are in Fig. S3C and the GitBook.

Multivariate statistical methods

Multivariate statistics can be divided into unsupervised and supervised methods based on whether the approach requires labeled data during training[2,70]. We will focus on the most popular methods.

Unsupervised dimensionality reduction using principal component analysis. Lipidomics and metabolomics data are high-dimensional (i.e., many measured molecular species), hampering visualization and analysis. Concentrations for several molecules are often correlated. However, as a result, dimensionality reduction methods allow to summarize this apparent high-dimensional dataset (e.g., 800 lipids) in a low-dimensional space (e.g., 2D or 3D plots) using new uncorrelated variables (components)[83]. While reducing the number of variables, dimensionality reduction aims to keep as much of the meaningful patterns of the original data as possible[70,83].

PCA remains one of the most popular ways of visualizing lipidomics and metabolomics data[64,70]. In PCA, a matrix containing the centered and eventually also scaled lipidomics or metabolomics data (*D*) is decomposed into two orthogonal matrices: so-called score (*S*) and loading (*L*) matrices (Fig. 4A)[40,70,83]. PCA essentially performs a linear transformation of the data into a new coordinate system, where most of the variation of the data occurs along a few axes, called the principal components (PC). These PCs are linear combinations of the original variables, and each subsequent one explains a smaller portion of the total variance in the data than the preceding one (Fig. 4)[70,83]. Figure 4B shows an example 2D representation (low dimensional representation), known as a score or PCA plot, where PC1 (*x*-axis) is plotted against PC2 (*y*-axis). Single points represent samples (observations). In Fig. 4B, two biological groups are presented using blue and red colored points, corresponding to controls and patients with gout, and the yellow dots represent pooled plasma samples (quality control, QC). As PCA is a linear transformation, distances in PCA plots are meaningful because similar samples will cluster together, while divergent samples will separate. In Fig. 4B, looking at the score plot along PC1, the separation between gout patients and healthy volunteers is observed (red dot vs. blue dots). Hence, we summarize that the

main source of variance, accounting for 42.0% of the total variance, separates samples along PC1 according to health status (healthy controls vs. gout patients). Notably, removing the QC samples from the PCA analysis does not affect the variance explained by PC1 (41.96% vs. 41.98%). The percentage of the total variance explained by each PC is usually presented using a scree (or elbow) plot (Fig. 4C). A benefit of PCA is that the PCs can often be interpreted by looking at their loadings, which represent the relative contribution weights of the original variables to PC (Fig. 4D). The sign of a loading indicates a positive (negative) influence of the respective input variable on the PC[70]. The PCA analysis shown in Fig. 4 can be reproduced using the R script provided in the GitBook. An informative visualization based on PCA is the biplot. It combines the scores and loadings into one graph.

However, PCA is sensitive to differences in variance (scale) among variables in a dataset. Therefore, most lipidomics and metabolomics data require preprocessing, e.g., log transformation to stabilize the variance, centering (subtracting the mean), and scaling[40,84]. PCA can be used to visually inspect whether biological groups are separated in the data. Complex datasets visually represented using PCA can be checked for confounders by inspecting sample distribution patterns, e.g., biological, like age- or gender-related differences, or technical, including batch effects or differences between collection sites. PCA also serves as a tool for assessing analytical method stability, e.g., by including balanced pooled QCs (from all samples measured within the sequence) to observe if they cluster in the middle of the PCA score plot. The clustering of QC samples in Fig. 4B demonstrates the integrity of the selected dataset.

Unsupervised dimensionality reduction with non-linear techniques. Other dimensionality reduction techniques, primarily used for visualizing high-dimensional data, are t-SNE (t-Distributed Stochastic Neighbor Embedding)[85] and UMAP (Uniform Manifold Approximation and Projection). In contrast to PCA, these techniques are non-linear. Interest in applying t-SNE and UMAP has recently increased in the -omics field, especially for transcriptomics and genomics data.

The t-SNE algorithm has two stages. First, a high-dimensional dataset $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ is randomly embedded into a low-dimensional (2D or 3D) dataset $\mathbf{Y} = \{y_1, y_2, \ldots, y_n\}$ (Fig. 5B)[85]. For both the high- and low-dimensional spaces, the pairwise distances of the data points are converted into joint probabilities that represent similarities (Fig. 5A, B)[85]. For high-dimensional data points x_j and x_i, similarity is computed as the conditional probability $p_{j|i}$ under a Gaussian distribution centered around x_i (Fig. 5A). For data points that are close together in the high-dimensional space, $p_{j|i}$ is relatively high, whereas for data points that are more distant, $p_{j|i}$ will be extremely small[85]. In the low-dimensional space, the pairwise distance of data points y_j and y_i, is modeled using the conditional probability $q_{j|i}$ under a Student-t distribution with one degree of freedom (also known as Cauchy distribution; Fig. 5B)[85]. If the low-dimensional data points y_j and y_i correctly model the similarity between x_j and x_i, then the conditional probabilities $p_{j|i}$ and $q_{j|i}$ must be as close to each other as possible[85].

Based on this consideration, the second stage of the algorithm aims to find a low-dimensional data representation that minimizes the mismatch between the two joint probabilities: p_{ij} and q_{ij} (Fig. 5C). The latter is achieved by minimizing the Kullback–Leibler (KL) divergence between the joint probability distribution P, for the high-dimensional space, and the joint probability distribution Q, for low-dimensional space as follows: $KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$[85]. In essence, the low-dimensional data points are shifted to minimize the mismatch between the low- and high-dimensional spaces (Fig. 5C).

UMAP essentially is similar, it first constructs a high-dimensional graph representation of the data and then optimizes a low-dimensional graph to be as structurally similar to this as possible. Two key theoretical components of UMAP are local manifold approximations and simplicial sets/complexes. UMAP assumes the high-dimensional data approximately lie on a lower-dimensional manifold, i.e., a topological space that locally resembles Euclidean space near each point.

To construct a topological representation (essentially, a weighted graph) for the high-dimensional data, UMAP combines their local fuzzy simplicial set representations (Fig. 6B)[86]. These are topological spaces constructed by gluing together combinatorial building blocks called simplices, where the fuzziness represents a decreasing likelihood of connection. A k-dimensional simplex or k-simplex is formed by taking the convex hull of k + 1 independent points[87] (Fig. 6A). For the low-dimensional data, the local manifold approximation is simply $\mathbb{R}^d$ (d is Euclidean space dimension). Similar to the high-dimensional data, by combining fuzzy simplicial sets, an equivalent topological representation is constructed for the low-dimensional data (Fig. 6C)[86]. UMAP then optimizes the layout of the data representation in the low dimensional space, by minimizing its cross-entropy with the high-dimensional representation, to maximize structural similarity between the two topological representations (Fig. 6D)[86].

Comparison of unsupervised dimensionality reduction techniques. Unsupervised dimensionality reduction techniques vary in computational efficiency, scalability, and ability to preserve data structure. PCA is computationally efficient and scales well with large datasets. t-SNE, while powerful for visualization and revealing local clusters, can be computationally intensive and slow on large datasets. UMAP strikes a balance, offering speed and scalability while preserving more structure than PCA. In addition, compared to t-SNE, UMAP preserves more of the global structure of the data while still capturing local patterns. On the other hand, PCA results are easier to interpret because they come from a linear transformation. t-SNE and UMAP, being non-linear, produce results that are more complex to interpret but can reveal more intricate structures.

Supervised data exploratory methods based on Partial Least Squares analysis. Supervised dimensionality reduction methods like Partial Least Squares (PLS) and Orthogonal Partial Least Squares (OPLS) can be considered as linear regression with latent variables, here constructed to achieve maximal covariance between the covariates and the response. As a result, compared to regular linear regression, (O)PLS can deal with high-dimensional data containing many, possibly correlated variables and a limited number of samples. These models can be applied to regression tasks, feature selection, and classification (i.e., PLS-Discriminant Analysis, DA)[33,70,88], making them particularly useful for exploring lipidomics and metabolomics data.

PLS consists of five steps: (i) centering and eventually also standardization of variables, (ii) dimensionality reduction (computation of linear combinations of predictors and responses), (iii) fitting a linear regression model using PLS components, (iv) model parameter tuning, and (v) model application (Fig. 7). PLS starts from a data matrix of measured concentrations (*X*) and a vector containing sample class labels (the classification case), or continuous response variable (*Y*). Both should be centered and scaled. PLS iteratively finds the directions (latent variables, LV, marked as T in Fig. 7) in the *X* space that explain the maximum variance in *Y* (as opposed to PCA, which finds the axes of maximal variance within *X*). Models can be tuned using *k*-fold cross-validation or leave-one-out cross-validation (LOOCV) (see below), i.e., how many PLS components should be kept in the final model to achieve the best performance (Fig. 7)[33,70,88,89].

In 2002, OPLS was published[90,91], which uses so-called orthogonal signal correction to maximize the explained covariance within the first LV (T_p). In turn, the orthogonal LVs (T_o) cover the variance that is not correlated to the response variables (orthogonal variance) (Fig. 7). In effect, OPLS allows for separate modeling of variations of predictors in *X* correlated and uncorrelated to class labels stored in *Y*. This can improve the model interpretability[70,90,91].

https://doi.org/10.1038/s41467-025-63751-1

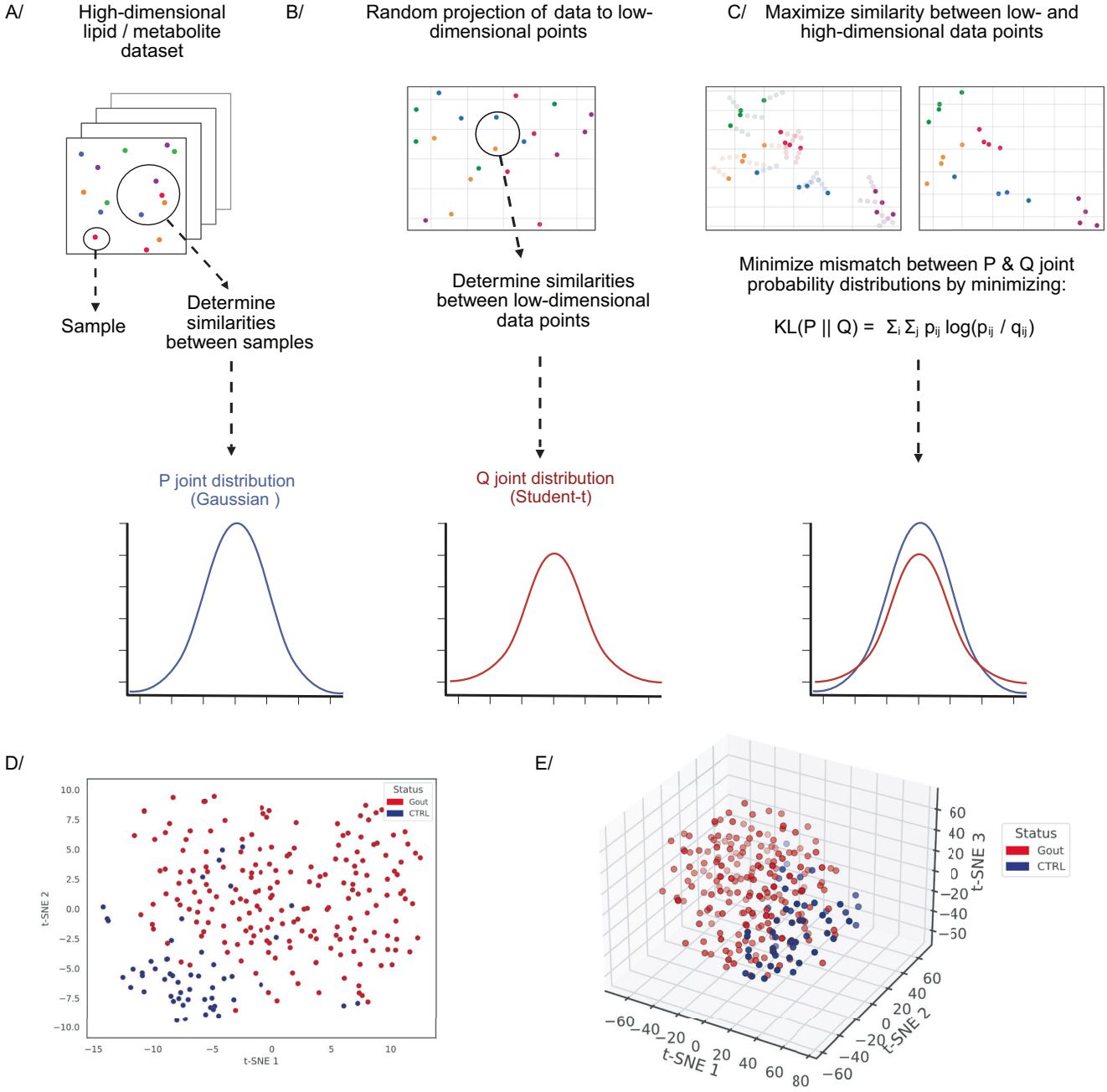


Fig. 5 | Conceptual overview of t-SNE. A–**C** t-Distributed Stochastic Neighbor Embedding algorithm. **A** The high-dimensional dataset needs to be projected into a low-dimensional representation. **A**, **B** Pairwise distances of high- and low-dimensional data points are converted into similarities using the joint probability distributions P (Gaussian) and Q (Student-t), respectively. **C** The low-dimensional representation that best resembles the high-dimensional dataset is found by minimizing the KL divergence between the two joint probability distributions. **D**, **E** 2D & 3D t-SNE scatter plots for a dataset composed of lipid concentrations analyzed via the LC-MS in plasma samples of patients with gout (red dots) and healthy subjects (blue dots). Created in BioRender. Demeulemeester, J. (2025) https://BioRender.com/sg7f248. The dataset has been published by Kvasnička et al. in their manuscript *Alterations in lipidome profiles distinguish early-onset hyperuricemia, gout, and the effect of urate-lowering treatment*[127].

Before building PLS models, data should be split into training, testing, and validation sets[88]. Usually, most data points are allocated to the training set. Several techniques are used for model training and tuning to avoid overfitting[70,88,92]. One of the most popular is *k*-fold cross-validation, which splits the data into *k* equal parts, where *k*–1 parts are used for training, and one part is used to test the model. The process is repeated *k*-times, and the final number of components in the PLS model is decided based on model performance calculated as the sum of squared differences of observed and predicted response values from *k*-fold cross-validation. For smaller datasets, LOOCV can be applied. Here, one observation is left out, and the rest is used for model training. The process is repeated until every single observation has been left out once for testing. The performance of the models can then finally be assessed using a validation set, which has not been used during the training-testing procedures[88,92]. In PLS-DA (or OPLS-DA), the group labels are predicted for every sample, and probability ranging between 0 and 1 is assigned[88]. In a two-class classification problem, 0 refers to controls, and 1 to conditions. A cut-off is selected, usually

https://doi.org/10.1038/s41467-025-63751-1

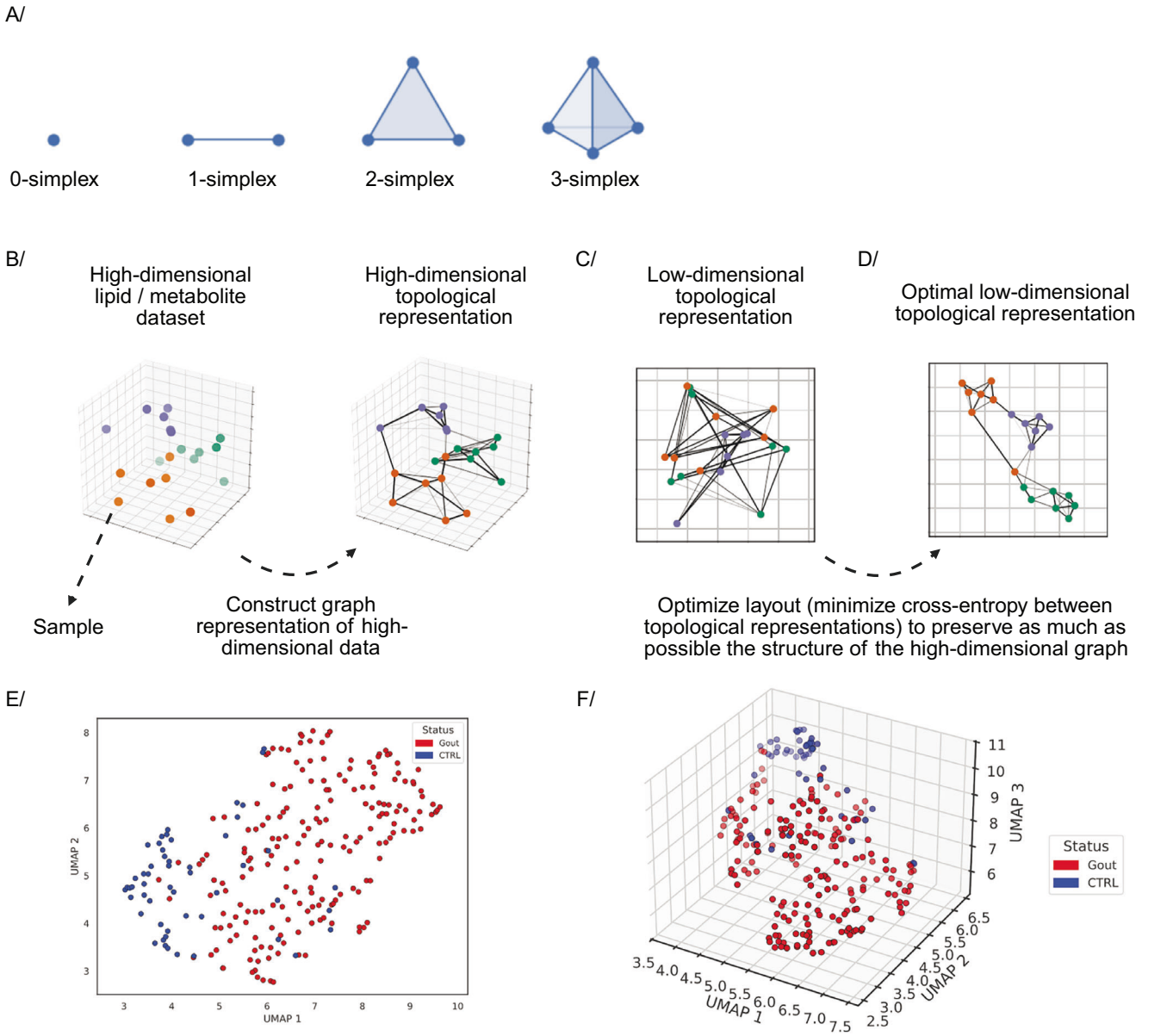


Fig. 6 | Uniform manifold approximation and projection algorithm. **A**–**D** Uniform Manifold Approximation and Projection algorithm. **A** Examples of simplices. **B** High-dimensional topological representation. **C** Low-dimensional topological representation. **D** The layout of the low-dimensional topological representation is optimized to maximally preserve the structure of the high-dimensional graph. **E**, **F** 2D & 3D UMAP scatter plots for a dataset composed of lipid concentrations analyzed via the LC-MS in plasma samples of patients with gout (red dots) and healthy subjects (blue dots). Figure adapted from[86,87] and Created in BioRender. Demeulemeester, J. (2025) https://BioRender.com/xcvtpps. The dataset has been published by Kvasnička et al. in their manuscript *Alterations in lipidome profiles distinguish early-onset hyperuricemia, gout, and the effect of urate-lowering treatment*[127].

equal to 0.5, and all values below 0.5 belong to controls, while above 0.5 to condition.

PLS can yield informative visualizations. A score plot can be obtained from two LVs of (O)PLS-DA[93] (see GitBook), or similarly to PCA – a biplot[94]. A performant (O)PLS-DA classifier will separate biological groups in its score plot. The discriminating features the model has revealed can be presented using variable importance plots (VIP)[70] or the S-plot for OPLS-DA (see GitBook)[95]. The S-plot is a scatter plot depicting covariance (y-axis) and correlation (x-axis) between the measured lipid or metabolite concentrations and the predictive scores. The molecules that are farthest from the origin are considered to be most important for discriminating between groups (see GitBook)[95]. A loading plot presenting weights for two selected LVs can also be shown[94]. For (O)PLS-DA, a receiver operating characteristic (ROC) curve can show classifier performance across a range of threshold values. The greater the area under the curve, the better the classifier (AUC = 1 indicates perfect classification) (see GitBook)[96].

Hierarchical clustering analysis. Clustering is broadly used in lipidomics and metabolomics to investigate similarities and differences between groups of samples or features. For simplicity, we focus on sample clustering (data points, observations), but these concepts may readily be applied to features, too. In the first step of clustering, a distance matrix is computed, containing the pairwise distances between all samples. The most used is Euclidean distance, i.e., the length of a straight line connecting one sample to the other. Hierarchical clustering is an algorithm for grouping similar objects into clusters. Generally, two types of hierarchical clustering are distinguished: agglomerative and divisive (Fig. S1A). The former starts with every observation as a separate cluster and iteratively combines

https://doi.org/10.1038/s41467-025-63751-1

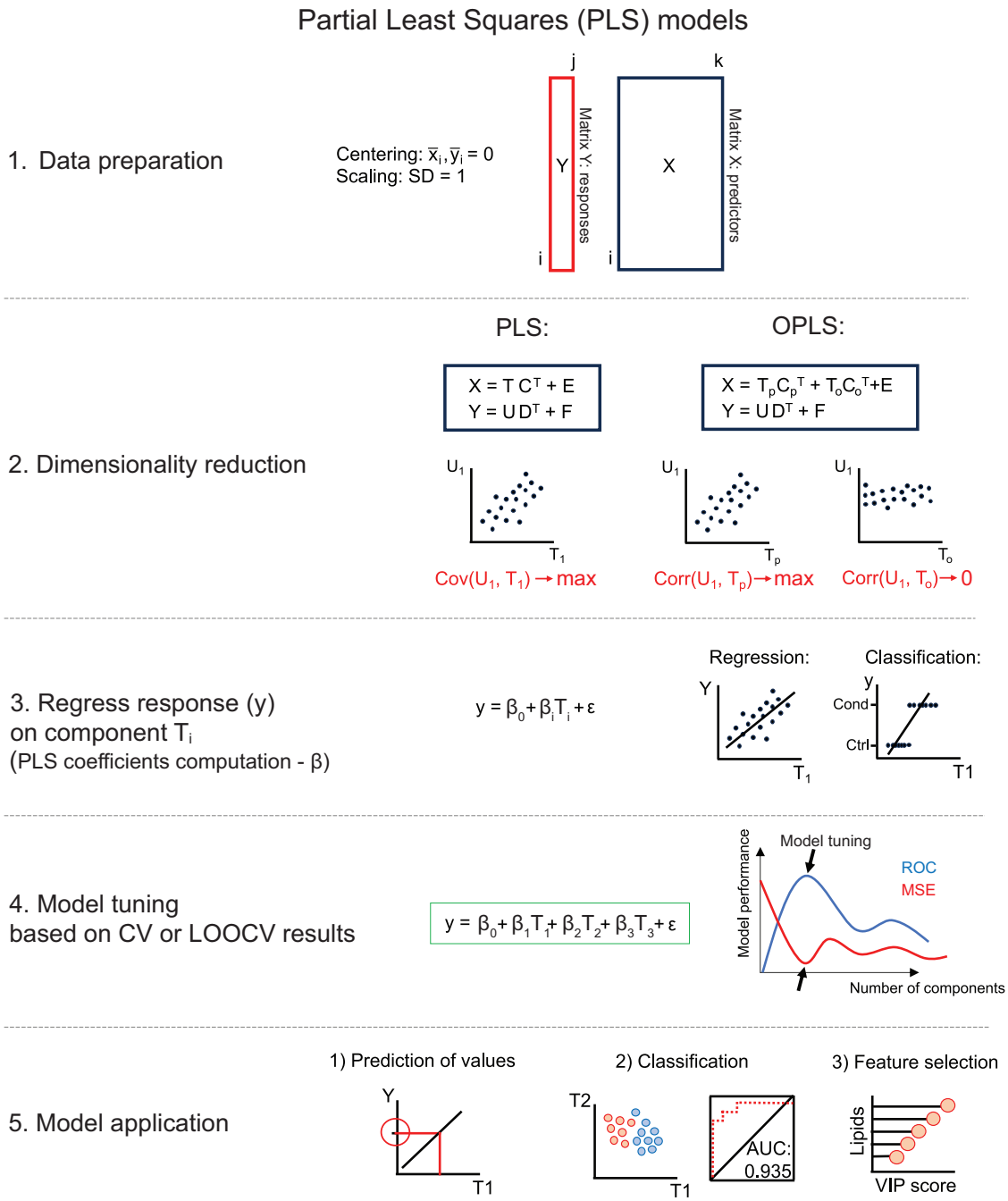


Fig. 7 | Key steps of (O)PLS model preparation and application. *X* denotes the matrix of predictors – with lipid or metabolite concentrations; *Y* is the vector of responses; *T* and *U* – are scores for *X* and *Y*; *C* and *D* are loadings for *X* and *Y*, respectively; *E* and *F* are residual matrices. For OPLS, the scores and loadings are additionally split into predictive (p) and orthogonal (o) parts. If *Y* is a vector, no decomposition is needed in step 2.

the closest clusters until only one remains. The latter approach starts with all observations in one cluster and iteratively splits them until each observation forms its own individual cluster. Among the myriad of criteria for deciding which clusters to merge in agglomerative clustering, Ward's linkage is popular for agglomerative clustering of lipidomics and metabolomics data, as it seeks to iteratively merge those clusters that minimize within-cluster variance[70].

Graphical representation of clustering analysis. Hierarchical clustering can be visualized using dendrograms. These consist of branches; samples on the same branch belong to the same cluster (Fig. S1A, B). The most common are rectangular dendrograms plotted vertically or horizontally (Fig. S1A), but circular (Fig. S1B) or triangular dendrograms can also be used[97,98]. Dendrograms are frequently plotted alongside heat maps of lipids or metabolites of interest[70] with the smallest *p*-values, the highest weights in PCA, or VIP scores from supervised approaches like (O)PLS. Using heat maps, trends for sample clusters can be readily captured if clustering is applied to features, too (Fig. S1C, D).

Beginners guide to R and Python libraries for lipidomics and metabolomics data visualization

Getting started with R and Python

We strongly encourage using RStudio as the integrated development environment (IDE) in the first individual steps of preparing simple R scripts [https://posit.co/download/rstudio-desktop]. RStudio is one of the most user-friendly graphical interfaces for R. It enables straightforward data loading, code preparation, highly organized data management, and simple generation and export of outputs (e.g., visualizations). RStudio also supports code optimization and debugging. The RStudio Education platform [https://education.rstudio.com/] can help streamline the learning path and provide a beginner with the latest updates on learning tools, e.g., the excellent book by H. Wickham and G. Grolemund R for Data Science (2nd edition published in June 2023)[99]. In the case of Python, we leverage Jupyter, a web-based interactive computing platform for recording work in Notebooks [https://jupyter.org]. The GitBook also contains an example of a Jupyter Notebook application for R. The advantage of Jupyter Notebooks relies on creating a structured overview combining code and outputs, which is particularly useful while working in a data science team. Indeed, RStudio also provides the R Markdown tool [https://rmarkdown.rstudio.com/], which is an integrated markup file operated within RStudio that can contain both plain text and executable code and is an excellent tool for producing reports and version control. The instructions for installing R, Python, RStudio, R Markdown, and activating Jupyter Notebook for Python and R can be found in the GitBook. In the text below, we will discuss packages (or libraries), referring to collections of code, documentation, example (sample) data, and most importantly – functions. To perform operations, functions need parameters the user provides via function arguments. Functions can be used, for instance, for plotting, performing computations, and manipulating data. Specific functions may only become available after downloading, installing, and loading the appropriate packages (see Gitbook examples). Common packages can usually be downloaded from repositories, such as the comprehensive R archive network (CRAN; [http://CRAN.R-project.org]); the Python package Index (PyPI; [https://pypi.org]); or resources like Bioconductor project delivering open source and open development software for bioinformatics [http://new.bioconductor.org]. All commands and functionality provided by a package are described in its vignettes (or library documentation). Occasionally, additional information and tutorials are published in the form of a GitHub book, or simply – a blog, or even YouTube-based tutorials. Authors of widely used packages also create discussion groups or forums for users to discuss issues or to report bugs (coding errors) in the packages. One such example of a coding forum, which we recommend users of all levels to search for solutions, is Stack Overflow [https://stackoverflow.com]. The complementary GitBook to this manuscript contains scripts, which are regular text files containing R/Python commands.

Below, we present a collection of packages for data science, with a focus on applications in lipidomics and metabolomics. We have strived to highlight packages offering publication-quality outputs with minimal command complexity while also considering their flexibility for output modification, simplicity of installation, and the comprehensiveness of vignettes and supporting materials. A summary of all the information is also provided as a table (Supplementary Table 2).

Data pre-processing and descriptive statistics in R and Python

R provides two excellent collections of packages that can be used to deal with data processing, namely *tidyverse* [https://www.tidyverse.org][100] and *tidymodels* [https://www.tidymodels.org]. The first collection is useful for general data science, speeding up and simplifying data importing and preparation, i.e., tidying, manipulation, and programming[100]. The second collection gathers packages facilitating every machine-learning step. Both collections constitute a complete solution for all initial operations, which are performed before statistical computations and visualization, for instance, defining column types in a data frame, missing values imputation, filtering, reorganization, selection of vectors, matrices, strings, or their elements, transformations, and scaling. All packages share similarities regarding design philosophy, data structures, and grammar[100]. Complex operations on the data can be performed within a single line of code using explicit commands (e.g., *arrange()*, *select()*, *filter()*, *slice()*, *glimpse()*, *pull()*, *summarize()*, *mutate()*, etc.) and chained together via functions called *pipes* (e.g., *%>%*), creating an efficient pipeline for data cleaning and preparation[100]. Such pipelines can also be used in the next step for statistical computations and visualizations, as functions from *tidyverse* and *tidymodels* are often compatible with commands from newly created libraries for statistical data analysis. Examples of such operations for *tidyverse* packages are presented in the GitBook.

The preprocessed data frames with normalized data can then be used for data quality assessment (DQA) and computing summary statistics. A large number of packages for DQA is available on CRAN, and a detailed analysis was performed by Mariño et al.[101]. Several solutions in R allow the obtaining of detailed tables with descriptive (group) statistics using a single command. Publication-ready tables with summary statistics can also be obtained in R using, e.g., the *gtsummary* package [https://www.danieldsjoberg.com/gtsummary][102]. Selected packages and commands suitable for beginners are presented in Table S1 and the GitBook. Complementary to the descriptive statistics, correlations for all variables can be easily computed, e.g., via *ggpubr* [https://rpkgs.datanovia.com/ggpubr/], *rstatix* [https://rpkgs.datanovia.com/rstatix/], or *Hmisc* packages [https://CRAN.R-project.org/package=Hmisc], or by basic R command *cor()*.

However, most patterns and descriptive statistics can be visualized using R's plotting capacities. Examples of the most user-friendly libraries for novices are presented in Fig. S2. Beginners in R can rely on solutions like *ggpubr* (publication-ready plots, [https://rpkgs.datanovia.com/ggpubr/]), *tidyplots*[103] (publication-ready plots, [https://tidyplots.org/]), *ggstatsplot* (based plots with statistical details, [https://indrajeetpatil.github.io/ggstatsplot/])[104], *dlookr* [https://choonghyunryu.github.io/dlookr/], *DataExplorer* (diagnosis, exploration or transformation of data, [https://boxuancui.github.io/DataExplorer/]), *GGally* (excellent visual summary of descriptive statistics, [https://ggobi.github.io/ggally/]), or *ggplot2* [https://ggplot2.tidyverse.org/]. Most of these packages wrap plotting and customization into one function, with significant flexibility in modifying the output. Furthermore, these libraries are compatible with *tidyverse* and *tidymodels* collections, or are a part of them (*ggplot2*), reducing the code complexity[100]. The interactive R graphics are created, e.g., via the *plotly* package [https://plotly.com/r]. However, the code for generating sophisticated charts can get complex, and experience may be necessary to prepare appropriate scripts. Usually, the more customization desired, the more likely direct use of the *ggplot2* library itself will be required. R scripts for preparing all plot types from Fig. S2 are presented in the GitBook.

In Python, *pandas* [https://pandas.pydata.org] is the package of choice for importing and manipulating tabular raw data. *Pandas* can read delimited text, Excel, and database files, and stores these in a DataFrame object (for 2-dimensional tabular data) or in a Series object (for 1-dimensional data). These objects allow for an easy access to data in rows and columns by means of their indices or names. Moreover, *pandas* can handle missing data and has a wide range of built-in functions that allow manipulations, such as filtering, reorganizing, scaling, and transforming the data. Similar to R, simple descriptive statistics for both numeric and categorical data can be obtained in *pandas* with a call to a single function (*describe*). *Pandas* also has a powerful group by functionality that allows for data aggregation, allowing summary statistics to be calculated per group.

There are several options to visualize *pandas*' descriptive statistics as either data tables directly or as figures, such as bar charts, box, and density plots. In Jupyter Notebook, the *pandas'* tables can be displayed directly in the browser. To obtain tables formatted to publication standards, *pandas* has several export options to enable rendering by other programs, such as LaTeX. For figures, the most comprehensive solutions in Python are *matplotlib* [https://matplotlib.org] and *plotly* packages [https://plotly.com/python]. Several alternatives are available, however, that simplify plotting, although this typically comes at the cost of flexibility. *Seaborn* [https://seaborn.pydata.org/], based on *matplotlib* and *plotly express* [https://plotly.com/python/plotly-express] allows plotting and customization within a single function. Simple plots can be created from *pandas* DataFrames directly with a single function call on the DataFrame (*bar, hist, box, density, area, scatter*), which also relies on *matplotlib* or *plotly* under the hood.

Univariate statistics in R and Python

The *rstatix* package [https://rpkgs.datanovia.com/rstatix] is a good option for novices to begin with statistical testing in R (see also GitBook). This library is compatible with *tidyverse* tools, including pipes and commands for modifying data frames. After data pre-processing, data frames will often be in the wide format (unique features in the first column, followed by samples in the remaining columns). However, performing univariate statistics in this format is not advised as it requires performing a row-wise operation on each column individually. One of the simplest solutions to overcome this issue is to transform the lipidomics or metabolomics data from their regular wide format into a long data frame (features repeatedly appear in the first column for every time they are measured) (Fig. 8A). The long data frame can then be passed via a pipe to a function from the *rstatix* library to perform basic hypothesis testing, e.g., *t*-test/Mann–Whitney *U* test, ANOVA/Kruskal-Wallis test, Tukey *HSD*/Dunn posthoc (Fig. 8A). Additionally, long data frames are also handy for plotting selected data, however the format may not be suitable for performing complex computations, e.g., machine learning applications. If in doubt, users can always refer to examples given in the RDocumentation for the function, which can be easily accessed by placing the caret in the function name and pressing F1. The output of the statistical test function call is a data frame containing information on test type, the total number of observations, test statistics, degrees of freedom, and *p*-value. The table can be extended by additional corrections to the *p*-value and significance symbols (asterisks) graphically representing the *p*-value. If more advanced pairwise comparisons are required, the *PMCMRplus* [https://CRAN.R-project.org/package=PMCMRplus] package can be used[81]. Univariate statistics can also be performed and extracted as publication-ready tables via the *gtsummary* package (Fig. 8B; GitBook). Publication-ready plots with a graphical representation of univariate test results and descriptions can be produced by functions from the *ggstatsplot* package (Fig. 8C)[104]. Good alternatives are also available in the *ggpubr* package. The data frames generated by *rstatix* can be modified to contain the information necessary for plotting statistical annotations, which can then be added to *ggpubr* or *ggplot2* plots via *stat_pvalue_manual()* function from the *ggpubr* package. Examples are shown in Fig. 8D.

The results of the two groups' comparisons can be represented by publication-ready volcano plots, which can be prepared using the *EnhancedVolcano* library [https://github.com/kevinblighe/EnhancedVolcano] (Fig. S3A) from the Bioconductor collection. Interactive volcano plots can be generated via *plotly* (see GitBook). An alternative to volcano plots is lipid maps (lipid networks), which can be constructed using the Cytoscape software based on data exported from R (Fig. S3B) or fatty acyl-chain plots generated via *ggplot2* (Fig. S3C). The guide for preparing all of these plots is presented in the GitBook.

In Python, the most comprehensive package that offers univariate statistical hypothesis testing is *statsmodels* [https://www.statsmodels.org]. *Statsmodels* is focused on the tests and does not provide visualization. Drawing statistical annotations on *matplotlib* plots, such as lines and asterisks to indicate significance, is left to the end user using low-level drawing functions. Recently, the *statannotations* package [https://pypi.org/project/statannotations] was created to address this by drawing statistical annotations automatically on *matplotlib* and *seaborn* plots. It has the drawback that, compared to *statsmodels*, it is limited in the statistical tests it supports.

Multivariate statistics in R and Python

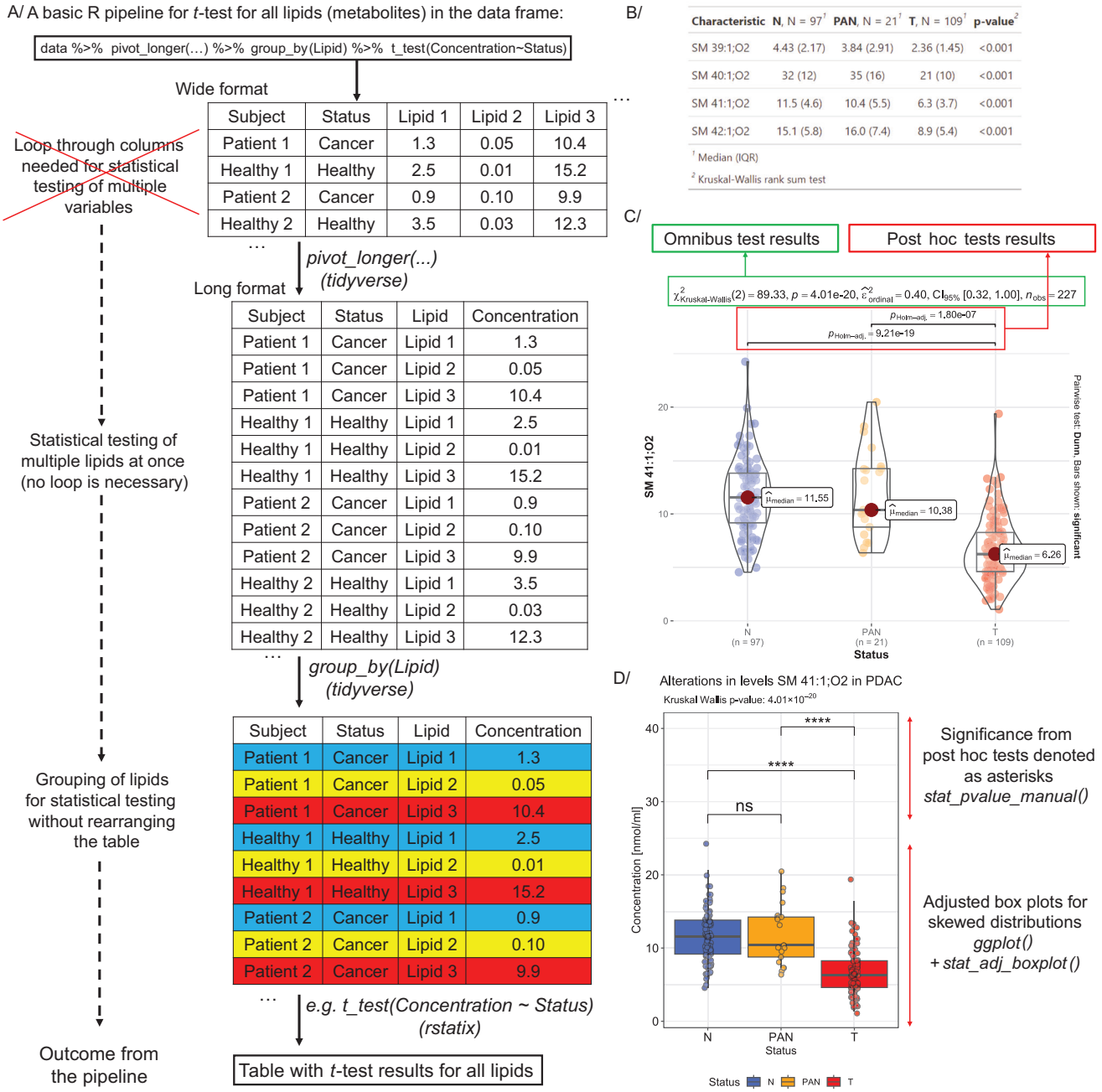
Several options are available for PCA in R, including base R and packages from Bioconductor, such as *pcaMethods*[105,106], *ropls*[107,108], *mixOmics* [http://mixomics.org][92]. Beginners should start with the simplest options available, such as the *ropls* package, which provides scores and loadings plots, a scree plot, and the dataset diagnostic plots based on two commands, additionally allowing for scaling of features inside the PCA-computing function. Another method relies on computing PCA on transformed and scaled data using the base R commands *princomp()* or *prcomp()*. Then, the output can be visualized using the *ggplot2*-dependent package *factoextra* [https://rpkgs.datanovia.com/factoextra/] as score and loading plots, a scree plot, and biplots. PCA output can also be plotted using *ggplot2* directly, or interactive plots can be produced using *plotly*. Examples showing these solutions are presented in the GitBook. Likewise for t-SNE and UMAP, there are several implementations both in Python and R. For t-SNE, the most commonly used libraries are *sklearn.manifold.TSNE* (Python) and *Rtsne* (R) [https://cran.r-project.org/web/packages/Rtsne/index.html]. For UMAP, the most common Python package is *umap-learn* [https://umap-learn.readthedocs.io/en/latest/], and in R is the *umap* library [https://cran.r-project.org/web/packages/umap/index.html], which is built upon Python implementation.

For (O)PLS, R provides two Bioconductor packages – *mixOmics* (PLS)[92] and *ropls* ((O)PLS)[108]. While performing the supervised dimensionality reduction and obtaining a scores plot for (O)PLS is manageable for beginners, training a regression model is more demanding from both a mathematical- and coding point of view. Although both *mixOmics* and *ropls* enable training models and applying them to particular tasks, the *caret* R package [https://CRAN.R-project.org/package=caret] can facilitate analyses by unifying model training, tuning, and predictions, providing one strict pipeline through uncomplicated commands[109]. Alternatively, *tidymodels* offers a unified pipeline for training PLS model relying on *mixOmics* engine. Both *caret* and *tidymodels* are compatible with *pROC* (preparation of ROC curves) to evaluate model performance[96]. The application of *caret*, *tidymodels* with *mixOmics*, and *ropls* libraries, including training of (O)PLS-DA models, is presented step-by-step in the GitBook.

The dendrograms and (associated) heat maps described here (Fig. S1) are generated in R using two Bioconductor packages – *ggtree*[97,98] and *ComplexHeatmap*[110–112]. Both libraries have detailed vignettes and tutorials suited to novices and experts. The *tidyHeatmap* [https://stemangiola.github.io/tidyHeatmap/] is another useful R package designed for pipe-friendly, modular heatmap production based on tidy principles[113]. Finally, Bioconductor provides a package *InteractiveComplexHeatmaps*[114,115] for turning *ComplexHeatmaps* outputs into interactive graphics.

In Python, the *scikit-learn* package [https://scikit-learn.org/stable] provides comprehensive multivariate data analysis and includes functions for data pre-processing, clustering, regression, classification, and dimensionality reduction. As is typical in the Python ecosystem, *scikit-learn* does not offer built-in data visualization but rather requires the user to supply its output to an external plotting package, typically *matplotlib* or *seaborn*. The *scikit-learn* documentation does, however,

https://doi.org/10.1038/s41467-025-63751-1



Subject	Status	Lipid 1	Lipid 2	Lipid 3
Patient 1	Cancer	1.3	0.05	10.4
Healthy 1	Healthy	2.5	0.01	15.2
Patient 2	Cancer	0.9	0.10	9.9
Healthy 2	Healthy	3.5	0.03	12.3

Subject	Status	Lipid	Concentration
Patient 1	Cancer	Lipid 1	1.3
Patient 1	Cancer	Lipid 2	0.05
Patient 1	Cancer	Lipid 3	10.4
Healthy 1	Healthy	Lipid 1	2.5
Healthy 1	Healthy	Lipid 2	0.01
Healthy 1	Healthy	Lipid 3	15.2
Patient 2	Cancer	Lipid 1	0.9
Patient 2	Cancer	Lipid 2	0.10
Patient 2	Cancer	Lipid 3	9.9
Healthy 2	Healthy	Lipid 1	3.5
Healthy 2	Healthy	Lipid 2	0.03
Healthy 2	Healthy	Lipid 3	12.3

Characteristic	N, N = 97[1]	PAN, N = 21[1]	T, N = 109[1]	p-value[2]
SM 39:1;O2	4.43 (2.17)	3.84 (2.91)	2.36 (1.45)	<0.001
SM 40:1;O2	32 (12)	35 (16)	21 (10)	<0.001
SM 41:1;O2	11.5 (4.6)	10.4 (5.5)	6.3 (3.7)	<0.001
SM 42:1;O2	15.1 (5.8)	16.0 (7.4)	8.9 (5.4)	<0.001

[1] Median (IQR)
[2] Kruskal-Wallis rank sum test

Fig. 8 | Hypothesis testing in R: *t*-test example. A A pipeline for performing statistical test for all lipids (metabolites) in a data frame relying on the *tidyverse* and *rstatix* packages. **B** R-generated publication-ready table containing elements of descriptive statistics (median, *IQR*), the total number of observations in each group, and a *p*-value from the Kruskal-Wallis test – based on the *gtsummary* package. **C** Violin box plots with complete univariate statistics obtained via a single *ggbetweenstats()* function from *ggstatsplot* package. **D** Box plots adjusted for skewed distributions (*ggplot2* and *litteR* libraries) with results of Dunn post hoc test depicted using asterisks (*ggpubr* package).

provide examples of how to plot the analysis results with *matplotlib*. The application of *scikit-learn* and *Seaborn* to PCA and PLS-DA models is presented step-by-step in the GitBook.

Conclusions and perspectives

This review aims to bridge the gap between theory and application, offering a comprehensive understanding and allowing effective utilization of key statistical methods in lipidomics and metabolomics. By providing access to a range of R and Python tools via a GitBook repository, it equips researchers with practical resources to begin using these programming languages for statistical data analysis.

There is a noticeable trend toward making R and Python tools more accessible for beginners, evident in the development of libraries such as *ggpubr, ggstatsplot*, or *tidyplots* for R, as well as *seaborn* for Python. These libraries allow users to perform advanced data analysis or visualization with just one function. Simultaneously, more advanced R users rely on access to comprehensive modular solutions (all-in-one collections), where often an R object is initially created and then downstream processed through a series of libraries that streamline each step of data processing and mining. For instance, capable R users can process a variety of raw mass spectrometry data with *tidyMass*[116] [https://www.tidymass.org/], *RforMassSpectrometry* [https://www.

rformassspectrometry.org/], *MetaboAnalystR*[117] [https://www.metaboanalyst.ca/docs/RTutorial.xhtml], or well-developed *xcms* package[118–120] [https://www.bioconductor.org/packages/release/bioc/html/xcms.html]. Then, perform initial analysis and visualization within these libraries, and finally create sophisticated, publication-ready, high-quality graphics using *tidyverse*, *ggpubr*, *plotly*, or specialized -omics libraries like *lipidr*[121] [https://www.lipidr.org/] and *LipidSigR*[46] [https://lipidsig.bioinfomics.org/lipidsigr/]. Similarly, once trained, Python users can employ all-encompassing *tidyMS*[122] [https://tidyms.readthedocs.io/en/latest/] or *OpenMS*[123] [https://openms.de/] for data processing and *seaborn* or *matplotlib* for visualization. Utilizing open-source tools provides -omics scientists with greater flexibility and a broader array of solutions, as seen in projects and toolboxes like metaRbolomics within Bioconductor[124]. This approach also decreases dependence on costly vendor software and supports scalable, reproducible, and standardized workflows.

Moreover, proficiency in these programming languages fosters adaptability in tackling emerging challenges in metabolomics and lipidomics research. By developing these skills, researchers can enhance their own analyses and extract deeper insights from complex omics data. This, in turn, drives advancements, for instance, in biomarker discovery, disease mechanisms, or personalized medicine within clinical and biomedical sciences.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Code availability

The GitBook can be accessed at: [https://laboratory-of-lipid-metabolism-a.gitbook.io/omics-data-visualization-in-r-and-python/].

References

1. Géhin, C., Fowler, S. J. & Trivedi, D. K. Chewing the fat: how lipidomics is changing our understanding of human health and disease in 2022. *Anal. Sci. Adv.* **4**, 104–131 (2023).
2. Kvasnička, A. et al. Clinical lipidomics in the era of the big data. *Clin. Chem. Lab. Med.* **61**, 587–598 (2023).
3. L. Symons, J. et al. Lipidomic atlas of mammalian cell membranes reveals hierarchical variation induced by culture conditions, subcellular membranes, and cell lineages. *Soft Matter* **17**, 288–297 (2021).
4. Surma, M. A. et al. Mouse lipidomics reveals inherent flexibility of a mammalian lipidome. *Sci. Rep.* **11**, 19364 (2021).
5. Slade, E. et al. Age and sex are associated with the plasma lipidome: findings from the GOLDN study. *Lipids Health Dis.* **20**, 30 (2021).
6. Weir, J. M. et al. Plasma lipid profiling in a large population-based cohort. *J. Lipid Res.* **54**, 2898–2908 (2013).
7. Beyene, H. B. et al. High-coverage plasma lipidomics reveals novel sex-specific lipidomic fingerprints of age and BMI: Evidence from two large population cohort studies. *PLoS Biol.* **18**, e3000870 (2020).
8. Lindqvist, H. M. et al. A randomized controlled dietary intervention improved the serum lipid signature towards a less atherogenic profile in patients with rheumatoid arthritis. *Metabolites* **11**, 632 (2021).
9. Eichelmann, F. et al. Deep lipidomics in human plasma: cardiometabolic disease risk and effect of dietary fat modulation. *Circulation* **146**, 21–35 (2022).
10. Israelsen, M. et al. Comprehensive lipidomics reveals phenotypic differences in hepatic lipid turnover in ALD and NAFLD during alcohol intoxication⋆. *JHEP Rep.* **3**, 100325 (2021).
11. Meikle, P. J. et al. Statin action favors normalization of the plasma lipidome in the atherogenic mixed dyslipidemia of MetS: potential relevance to statin-associated dysglycemia. *J. Lipid Res.* **56**, 2381–2392 (2015).
12. Matthiesen, R. et al. Shotgun mass spectrometry-based lipid profiling identifies and distinguishes between chronic inflammatory diseases. *eBioMedicine* **70**, 103504 (2021).
13. Chua, E. C.-P. et al. Extensive diversity in circadian regulation of plasma lipids and evidence for different circadian metabolic phenotypes in humans. *Proc. Natl. Acad. Sci. USA* **110**, 14468–14473 (2013).
14. Gnocchi, D., Pedrelli, M., Hurt-Camejo, E. & Parini, P. Lipids around the clock: focus on circadian rhythms and lipid metabolism. *Biology* **4**, 104–132 (2015).
15. Sinturel, F., Spaleniak, W. & Dibner, C. Circadian rhythm of lipid metabolism. *Biochem Soc. Trans.* **50**, 1191–1204 (2022).
16. Huynh, K. et al. High-throughput plasma lipidomics: detailed mapping of the associations with cardiometabolic risk factors. *Cell Chem. Biol.* **26**, 71–84.e4 (2019).
17. Wolrab, D. et al. Lipidomic profiling of human serum enables detection of pancreatic cancer. *Nat. Commun.* **13**, 124 (2022).
18. Wolrab, D. et al. Plasma lipidomic profiles of kidney, breast and prostate cancer patients differ from healthy controls. *Sci. Rep.* **11**, 20322 (2021).
19. Afshinnia, F. et al. Lipidomic signature of progression of chronic kidney disease in the chronic renal insufficiency cohort. *Kidney Int. Rep.* **1**, 256–268 (2016).
20. Pei, K. et al. An overview of lipid metabolism and nonalcoholic fatty liver disease. *Biomed. Res. Int.* **2020**, 4020249 (2020).
21. Graessler, J. et al. Top-down lipidomics reveals ether lipid deficiency in blood plasma of hypertensive patients. *PLOS ONE* **4**, e6261 (2009).
22. Vvedenskaya, O. et al. Nonalcoholic fatty liver disease stratification by liver lipidomics. *J. Lipid Res.* **62**, 100104 (2021).
23. Han, S. et al. TIGER: technical variation elimination for metabolomics data using ensemble learning architecture. *Brief. Bioinform.* **23**, bbab535 (2022).
24. Altman, N. & Krzywinski, M. Sources of variation. *Nat. Methods* **12**, 5–6 (2015).
25. Olshansky, G., Giles, C., Salim, A. & Meikle, P. J. Challenges and opportunities for prevention and removal of unwanted variation in lipidomic studies. *Prog. Lipid Res.* **87**, 101177 (2022).
26. McDonald, J. G. et al. Introducing the lipidomics minimal reporting checklist. *Nat. Metab.* **4**, 1086–1088 (2022).
27. Köfeler, H. C. et al. Recommendations for good practice in MS-based lipidomics. *J. Lipid Res.* **62**, 100138 (2021).
28. Liebisch, G. et al. Lipidomics needs more standardization. *Nat. Metab.* **1**, 745–747 (2019).
29. Liebisch, G. et al. Shorthand notation for lipid structures derived from mass spectrometry. *J. Lipid Res.* **54**, 1523–1530 (2013).
30. Liebisch, G. et al. Update on LIPID MAPS classification, nomenclature, and shorthand notation for MS-derived lipid structures. *J. Lipid Res.* **61**, 1539–1555 (2020).
31. Holčapek, M., Liebisch, G. & Ekroos, K. Lipidomic analysis. *Anal. Chem.* **90**, 4249–4257 (2018).
32. Kopczynski, D. et al. The lipidomics reporting checklist a framework for transparency of lipidomic experiments and repurposing resource data. *J. Lipid Res.* **65**, 100621 (2024).
33. Ruiz-Perez, D., Guan, H., Madhivanan, P., Mathee, K. & Narasimhan, G. So you think you can PLS-DA?. *BMC Bioinform.* **21**, 2 (2020).
34. Wei, R. et al. Missing value imputation approach for mass spectrometry-based metabolomics data. *Sci. Rep.* **8**, 663 (2018).
35. Frölich, N., Klose, C., Widén, E., Ripatti, S. & Gerl, M. J. Imputation of missing values in lipidomic datasets. *Proteomics* **24**, 2300606 (2024).

36. Do, K. T. et al. Characterization of missing values in untargeted MS-based metabolomics data and evaluation of missing data handling strategies. *Metabolomics* **14**, 128 (2018).
37. Armitage, E. G., Godzien, J., Alonso-Herranz, V., López-Gonzálvez, Á & Barbas, C. Missing value imputation strategies for metabolomics data. *Electrophoresis* **36**, 3050–3060 (2015).
38. Kokla, M., Virtanen, J., Kolehmainen, M., Paananen, J. & Hanhineva, K. Random forest-based imputation outperforms other methods for imputing LC-MS metabolomics data: a comparative study. *BMC Bioinform.* **20**, 492 (2019).
39. González-Domínguez, Á, Estanyol-Torres, N., Brunius, C., Landberg, R. & González-Domínguez, R. QComics: recommendations and guidelines for robust, easily implementable and reportable quality control of metabolomics data. *Anal. Chem.* **96**, 1064–1072 (2024).
40. van den Berg, R. A., Hoefsloot, H. C., Westerhuis, J. A., Smilde, A. K. & van der Werf, M. J. Centering, scaling, and transformations: improving the biological information content of metabolomics data. *BMC Genom.* **7**, 142 (2006).
41. Wong, G., Chan, J., Kingwell, B. A., Leckie, C. & Meikle, P. J. LICRE: unsupervised feature correlation reduction for lipidomics. *Bioinformatics* **30**, 2832–2833 (2014).
42. Vinaixa, M. et al. A guideline to univariate statistical analysis for LC/MS-based untargeted metabolomics-derived data. *Metabolites* **2**, 775–795 (2012).
43. Bowden, J. A. et al. Harmonizing lipidomics: NIST interlaboratory comparison exercise for lipidomics using SRM 1950-Metabolites in Frozen Human Plasma. *J. Lipid Res.* **58**, 2275–2288 (2017).
44. Chocholoušková, M. et al. Intra-laboratory comparison of four analytical platforms for lipidomic quantitation using hydrophilic interaction liquid chromatography or supercritical fluid chromatography coupled to quadrupole - time-of-flight mass spectrometry. *Talanta* **231**, 122367 (2021).
45. Pang, Z. et al. MetaboAnalyst 5.0: narrowing the gap between raw spectra and functional insights. *Nucleic Acids Res.* **49**, W388–W396 (2021).
46. Lin, W.-J. et al. LipidSig: a web-based tool for lipidomic data analysis. *Nucleic Acids Res.* **49**, W336–W345 (2021).
47. Mohamed, A. & Hill, M. M. LipidSuite: interactive web server for lipidomics differential and enrichment analysis. *Nucleic Acids Res.* **49**, W346–W351 (2021).
48. LIPID MAPS. https://www.lipidmaps.org/resources/tools/stats.
49. Sun, X. & Weckwerth, W. COVAIN: a toolbox for uni- and multivariate statistics, time-series and correlation network analysis and inverse estimation of the differential Jacobian from metabolomics covariance data. *Metabolomics* **8**, 81–93 (2012).
50. Del Prete, E. et al. ADViSELipidomics: a workflow for analyzing lipidomics data. *Bioinformatics* **38**, 5460–5462 (2022).
51. Karpievitch, Y. V., Dabney, A. R. & Smith, R. D. Normalization and missing value imputation for label-free LC-MS analysis. *BMC Bioinform.* **13**, S5 (2012).
52. Ou, H. et al. Imputation for lipidomics and metabolomics (ImpLiMet): a web-based application for optimization and method selection for missing data imputation. *Bioinform. Adv.* **5**, vbae209 (2025).
53. Webb-Robertson, B.-J. M. et al. Review, evaluation, and discussion of the challenges of missing value imputation for mass spectrometry-based label-free global proteomics. *J. Proteome Res.* **14**, 1993–2001 (2015).
54. De Livera, A. M. et al. Normalizing and integrating metabolomics data. *Anal. Chem.* **84**, 10768–10776 (2012).
55. Wu, Y. & Li, L. Sample normalization methods in quantitative metabolomics. *J. Chromatogr. A* **1430**, 80–95 (2016).
56. Lipid Species Quantification – lipidomicstandards.org. https://lipidomicstandards.org/lipid-species-quantification/.
57. Low, B., Wang, Y., Zhao, T., Yu, H. & Huan, T. Closing the knowledge gap of post-acquisition sample normalization in untargeted metabolomics. *ACS Meas. Sci. Au* **4**, 702–711 (2024).
58. Drotleff, B. & Lämmerhofer, M. Guidelines for selection of internal standard-based normalization strategies in untargeted lipidomic profiling by LC-HR-MS/MS. *Anal. Chem.* **91**, 9836–9843 (2019).
59. Ghafari, N. & Sleno, L. Challenges and recent advances in quantitative mass spectrometry-based metabolomics. *Anal. Sci. Adv.* **5**, e2400007 (2024).
60. Livera, A. M. D. et al. Statistical methods for handling unwanted variation in metabolomics data. *Anal. Chem.* **87**, 3606–3615 (2015).
61. Fan, S. et al. Systematic error removal using random forest for normalizing large-scale untargeted lipidomics data. *Anal. Chem.* **91**, 3590–3596 (2019).
62. Filzmoser, P. & Walczak, B. What can go wrong at the data normalization step for identification of biomarkers?. *J. Chromatogr. A* **1362**, 194–205 (2014).
63. 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.* **78**, 4281–4290 (2006).
64. Saccenti, E., Hoefsloot, H. C. J., Smilde, A. K., Westerhuis, J. A. & Hendriks, M. M. W. B. Reflections on univariate and multivariate analysis of metabolomics data. *Metabolomics* **10**, 361–374 (2014).
65. Mohan, S. & Su, M. K. Biostatistics and epidemiology for the toxicologist: measures of central tendency and variability—where is the "middle?" and what Is the "spread?". *J. Med Toxicol.* **18**, 235–238 (2022).
66. Christopher, A. *Interpreting and Using Statistics in Psychological Research*. https://us.sagepub.com/en-us/nam/interpreting-and-using-statistics-in-psychological-research/book245631 (SAGE Publications Inc., 2023).
67. Yadav, S. K., Singh, S. & Gupta, R. *Biomedical Statistics: A Beginner's Guide*. https://doi.org/10.1007/978-981-32-9294-9 (Springer, Singapore, 2019).
68. Ospina, R. & Marmolejo-Ramos, F. Performance of some estimators of relative variability. *Front. Appl. Mathe. Stat.* **5**, 43 (2019).
69. Rosner B. *Fundamentals of Biostatistics*(Cengage Learning, 2016).
70. Checa, A., Bedia, C. & Jaumot, J. Lipidomic data analysis: tutorial, practical guidelines and applications. *Anal. Chim. Acta* **885**, 1–16 (2015).
71. Hubert, M. & Vandervieren, E. An adjusted boxplot for skewed distributions. *Comput. Stat. Data Anal.* **52**, 5186–5201 (2008).
72. Krzywinski, M. & Altman, N. Visualizing samples with box plots. *Nat. Methods* **11**, 119–120 (2014).
73. Schulz, M., Walvoort, D. J. J., Barry, J., Fleet, D. M. & van Loon, W. M. G. M. Baseline and power analyses for the assessment of beach litter reductions in the European OSPAR region. *Environ. Pollut.* **248**, 555–564 (2019).
74. Hintze, J. L. & Nelson, R. D. Violin Plots: A Box Plot-Density Trace Synergism. *Am. Stat.* **52**, 181–184 (1998).
75. Gowda, H. et al. Interactive XCMS Online: simplifying advanced metabolomic data processing and subsequent statistical analyses. *Anal. Chem.* **86**, 6931–6939 (2014).
76. Fagerland, M. W. t-tests, non-parametric tests, and large studies—a paradox of statistical practice?. *BMC Med. Res. Methodol.* **12**, 78 (2012).
77. Azizi, F., Ghasemi, R. & Ardalan, M. Two common mistakes in applying ANOVA test: guide for biological researchers. Preprint at https://doi.org/10.20944/preprints202207.0082.v1 (2022).
78. Analysis of Variance. in *The Concise Encyclopedia of Statistics* 9–11. https://doi.org/10.1007/978-0-387-32833-1_8 (Springer, New York, 2008).

79. RPubs - Post-Hoc Analysis with Tukey's Test. https://rpubs.com/aaronsc32/post-hoc-analysis-tukey.
80. Kruskal-Wallis Test. in *The Concise Encyclopedia of Statistics* 288–290. https://doi.org/10.1007/978-0-387-32833-1_216 (Springer, New York, 2008).
81. Pohlert, T. The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR)
82. Shannon, P. et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).
83. Jolliffe, I. T. & Cadima, J. Principal component analysis: a review and recent developments. *Philos. Trans. R. Soc. A:Math. Phys. Eng. Sci.* **374**, 20150202 (2016).
84. Importance of Feature Scaling. *scikit-learn* https://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html.
85. van der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
86. McInnes, L., Healy, J. & Melville, J. UMAP: Uniform Manifold Approximation and Projection. *J. Open Source Softw.* **3**, 861 (2018).
87. Sainburg, T., McInnes, L. & Gentner, T. Q. Parametric UMAP embeddings for representation and semisupervised learning. *Neural Comput.* **33**, 2881–2907 (2021).
88. Lee, L. C., Liong, C.-Y. & Jemain, A. A. Partial least squares-discriminant analysis (PLS-DA) for classification of high-dimensional (HD) data: a review of contemporary practice strategies and knowledge gaps. *Analyst* **143**, 3526–3539 (2018).
89. Liland, K. H., Stefansson, P. & Indahl, U. G. Much faster cross-validation in PLSR-modelling by avoiding redundant calculations. *J. Chemom.* **34**, e3201 (2020).
90. Trygg, J. & Wold, S. Orthogonal projections to latent structures (O-PLS). *J. Chemom.* **16**, 119–128 (2002).
91. Bylesjö, M. et al. OPLS discriminant analysis: combining the strengths of PLS-DA and SIMCA classification. *J. Chemom.* **20**, 341–351 (2006).
92. Rohart, F., Gautier, B., Singh, A. & Lê Cao, K.-A. mixOmics: an R package for 'omics feature selection and multiple data integration. *PLoS Comput. Biol.* **13**, e1005752 (2017).
93. Worley, B., Halouska, S. & Powers, R. Utilities for quantifying separation in PCA/PLS-DA scores plots. *Anal. Biochem.* **433**, 102–104 (2013).
94. Štefelová, N., Palarea-Albaladejo, J., Hron, K., Gába, A. & Dygrýn, J. Compositional PLS biplot based on pivoting balances: an application to explore the association between 24-h movement behaviours and adiposity. *Comput. Stat.* https://doi.org/10.1007/s00180-023-01324-w (2023).
95. Wiklund, S. et al. Visualization of GC/TOF-MS-based metabolomics data for identification of biochemically interesting compounds Using OPLS class models. *Anal. Chem.* **80**, 115–122 (2008).
96. Robin, X. et al. pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinform.* **12**, 77 (2011).
97. Yu, G. Using ggtree to visualize data on tree-like structures. *Curr. Protoc. Bioinform.* **69**, e96 (2020).
98. Yu, G., Smith, D. K., Zhu, H., Guan, Y. & Lam, T. T.-Y. ggtree: an r package for visualization and annotation of phylogenetic trees with their covariates and other associated data. *Methods Ecol. Evol.* **8**, 28–36 (2017).
99. Wickham, H. et al. *R for Data Science*, 2nd edn (2023).
100. Wickham, H. et al. Welcome to the Tidyverse. *J. Open Sour. Softw.* **4**, 1686 (2019).
101. Mariño, J., Kasbohm, E., Struckmann, S., Kapsner, L. A. & Schmidt, C. O. R Packages for data quality assessments and data monitoring: a software scoping review with recommendations for future developments. *Appl. Sci.* **12**, 4238 (2022).
102. Sjoberg, D. et al. Reproducible summary tables with the gtsummary Package. *R. J.* **13**, 570 (2021).
103. Engler, J. B. Tidyplots empowers life scientists with easy code-based data visualization. *Imeta* **4**, e70018 (2025).
104. Patil, I. Visualizations with statistical details: The 'ggstatsplot' approach. *J. Open Sour. Softw.* **6**, 3167 (2021).
105. Stacklies, W., Redestig, H. & Wright, K. pcaMethods: a collection of PCA methods. Bioconductor version: Release (3.17) https://doi.org/10.18129/B9.bioc.pcaMethods (2023).
106. Stacklies, W., Redestig, H., Scholz, M., Walther, D. & Selbig, J. pcaMethods—a bioconductor package providing PCA methods for incomplete data. *Bioinformatics* **23**, 1164–1167 (2007).
107. Thévenot, E. A., Roux, A., Xu, Y., Ezan, E. & Junot, C. Analysis of the human adult urinary metabolome variations with age, body mass index, and gender by implementing a comprehensive workflow for univariate and OPLS statistical analyses. *J. Proteome Res.* **14**, 3322–3335 (2015).
108. Thevenot, E. A. ropls: PCA, PLS(-DA) and OPLS(-DA) for multivariate analysis and feature selection of omics data. Bioconductor version: Release (3.17) https://doi.org/10.18129/B9.bioc.ropls (2023).
109. Kuhn, M. Building Predictive Models in R Using the caret Package. *J. Stat. Softw.* **28**, 1–26 (2008).
110. Gu, Z., Eils, R. & Schlesner, M. Complex heatmaps reveal patterns and correlations in multidimensional genomic data. *Bioinformatics* **32**, 2847–2849 (2016).
111. Gu, Z. Complex heatmap visualization. *Imeta* **1**, e43 (2022).
112. Gu, Z. ComplexHeatmap: Make Complex Heatmaps. Bioconductor version: Release (3.17) https://doi.org/10.18129/B9.bioc.ComplexHeatmap (2023).
113. Mangiola, S. & Papenfuss, A. T. tidyHeatmap: an R package for modular heatmap production based on tidy principles. *J. Open Sour. Softw.* **5**, 2472 (2020).
114. Gu, Z. & Hübschmann, D. Make interactive complex heatmaps in R. *Bioinformatics* **38**, 1460–1462 (2022).
115. Gu, Z. InteractiveComplexHeatmap: Make Interactive Complex Heatmaps. Bioconductor version: Release (3.17) https://doi.org/10.18129/B9.bioc.InteractiveComplexHeatmap (2023).
116. Shen, X. et al. TidyMass an object-oriented reproducible analysis framework for LC–MS data. *Nat. Commun.* **13**, 4365 (2022).
117. Pang, Z. et al. MetaboAnalystR 4.0: a unified LC-MS workflow for global metabolomics. *Nat. Commun.* **15**, 3675 (2024).
118. 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.* **78**, 779–787 (2006).
119. Tautenhahn, R., Böttcher, C. & Neumann, S. Highly sensitive feature detection for high resolution LC/MS. *BMC Bioinform.* **9**, 504 (2008).
120. Benton, H. P., Want, E. J. & Ebbels, T. M. D. Correction of mass calibration gaps in liquid chromatography-mass spectrometry metabolomics data. *Bioinformatics* **26**, 2488–2489 (2010).
121. Mohamed, A., Molendijk, J. & Hill, M. M. lipidr: a Software Tool for Data Mining and Analysis of Lipidomics Datasets. *J. Proteome Res.* **19**, 2890–2897 (2020).
122. Riquelme, G., Zabalegui, N., Marchi, P., Jones, C. M. & Monge, M. E. A Python-based pipeline for preprocessing LC–MS Data for untargeted metabolomics workflows. *Metabolites* **10**, 416 (2020).
123. Röst, H. L. et al. OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nat. Methods* **13**, 741–748 (2016).
124. Stanstrup, J. et al. The metaRbolomics Toolbox in Bioconductor and beyond. *Metabolites* **9**, 200 (2019).
125. Jirásko, R. et al. Altered plasma, urine, and tissue profiles of sulfatides and sphingomyelins in patients with renal cell carcinoma. *Cancers* **14**, 4622 (2022).

126. Idkowiak, J. et al. Robust and high-throughput lipidomic quantitation of human blood samples using flow injection analysis with tandem mass spectrometry for clinical use. *Anal. Bioanal. Chem.* **415**, 935–951 (2023).
127. Kvasnička, A. et al. Alterations in lipidome profiles distinguish early-onset hyperuricemia, gout, and the effect of urate-lowering treatment. *Arthritis Res. Ther.* **25**, 234 (2023).

Acknowledgements

Authors acknowledge the support of ERC Adv grant No. 101095860 sponsored by the European Research Council and the project JA344644 sponsored by the Ministry of Education, Youth and Sports, Czech Republic. L.M.B. and J.V.S. are supported by the Movember Foundation and the Prostate Cancer Foundation of Australia (MRTA3) and the U.S. Department of Defense (PC180582). J.X.M.T. is supported by a PhD scholarship from the University of Adelaide and a supplementary PhD scholarship from the Freemasons Centre for Male Health and Wellbeing at the University of Adelaide. J.I. is supported by the Leuven Future Fund LISCO-BIOMED. J. Dehairs and J.I. are supported by the Research Foundation Flanders (FWO) – SBO – LIPOMACS S001623N.

Author contributions

M.H., R.J., and J.I. had the original idea for this project. J.I., J.Dehairs., D.O., A.K., M.E., R.C., and J.D. prepared the original draft of the manuscript. J.Dehairs. and J.I. had the original idea for creating GitBook, and then they contributed to reviewing/editing/drafting/testing the code together with J.S., D.O., J.X.M.T., A.K., M.E., R.C., X.S., V.V., M.G., and V.D.L. The visualization of graphs was realized by contributions from J.I., J.Dehairs., J.S., J.X.M.T., X.S., and V.D.L. The supervision was performed by R.J., L.B., W.W., D.F., J.D., K.H., J.V.S., and M.H. All coauthors contributed to the manuscript reviewing and editing, and approved the final version of the manuscript. M.H. was responsible for project administration.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-63751-1.

Correspondence and requests for materials should be addressed to Michal Holčapek.

Peer review information *Nature Communications* thanks Xiaotao Shen and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

[1]Department of Analytical Chemistry, Faculty of Chemical Technology, University of Pardubice, Pardubice, Czechia. [2]Laboratory of Lipid Metabolism and Cancer, Department of Oncology, Leuven Cancer Institute (LKI) and Leuven Institute for Single Cell Omics (LISCO), KU Leuven, Leuven, Flanders, Belgium. [3]Department of Biomedical Engineering, Faculty of Electrical Engineering and Communication, Brno University of Technology, Brno, Czechia. [4]Molecular Systems Biology (MOSYS), Department of Functional and Evolutionary Ecology, Faculty of Life Sciences, University of Vienna, Vienna, Austria. [5]Department of Molecular and Clinical Pathology and Medical Genetics, University Hospital Ostrava, Ostrava, Czechia. [6]Institute of Experimental Endocrinology, Biomedical Research Center, Slovak Academy of Sciences, Bratislava, Slovakia. [7]Institute of Neuroimmunology, Slovak Academy of Sciences, Bratislava, Slovakia. [8]South Australian Health and Medical Research Institute (SAHMRI), North Terrace, Adelaide, SA, Australia. [9]South Australian immunoGENomics Cancer Institute (SAiGENCI) & Freemasons Centre for Male Health and Well-Being, The University of Adelaide Medical School, North Terrace, Adelaide, SA, Australia. [10]Laboratory for Inherited Metabolic Disorders, Department of Clinical Biochemistry, University Hospital Olomouc and Faculty of Medicine and Dentistry, Palacký University Olomouc, Olomouc, Czechia. [11]Department of Medical Biochemistry, Oslo University Hospital, Oslo, Norway. [12]Laboratory of Integrative Cancer Genomics, VIB-KU Leuven Center for Cancer Biology, Leuven, Flanders, Belgium. [13]VIB Center for AI & Computational Biology, Leuven, Flanders, Belgium. [14]Department of Oncology, KU Leuven, Leuven, Flanders, Belgium. [15]Laboratory of Multi-Omic Integrative Bioinformatics, Department of Human Genetics, KU Leuven, Leuven, Flanders, Belgium. [16]Laboratory of Applied Mass Spectrometry, Department of Cellular and Molecular Medicine, KU Leuven, Leuven, Flanders, Belgium. [17]Metabolomics Core Facility, VIB-KU Leuven Center for Cancer Biology, Leuven, Flanders, Belgium. [18]Vienna Metabolomics Center (VIME), University of Vienna, Vienna, Austria. [19]Department of Mathematical Analysis and Applications of Mathematics, Faculty of Science, Palacký University Olomouc, Olomouc, Czech Republic. ✉e-mail: Michal.Holcapek@upce.cz