Omics Untargeted Key Scripts

Preface

R based open-source collection of scripts called *OUKS* (Omics Untargeted Key Script) providing comprehensive nine step LC-MS untargeted metabolomic profiling data processing toolbox.

The session info snapshot, information about used packages with comments and scripts are available from https://github.com/plyush1993/OUKS. Each script consists of contents, comments, references and notes in places where functions should be adjusted for experimental parameters (usually "# adjust to your data"). Working directory was set in each script by setwd function, all data tables were loaded into environment in csv format.

Description and instruction for every function can be obtained by the code: ?function. Self-written functions are commented in the script. We utilized a special format for the name of every experimental injection file: "150. qc30 b6 QC30.CDF" for QC sample and "167. s64_2 b7 MS 45.CDF" for study file. Thus, the first value is a run order, then the index of sample with repeat number, batch value and clinical/experimental ID (or QC ID). This form allows to obtaining key information directly from the file name and automatically constructs data frame in the R environment. You can load a similar table to your environment manually if you prefer a different name format.

Raw data (.CDF format) and table with metadata ("metadata.csv") are the only requirements to reproduce the entire code and are available from Metabolomics Workbench, study ID: ST001682 (http://doi.org/10.21228/M8ZT4C). Other input files (.csv data tables and .R/.RData files) can be generated in the appropriate script or downloaded from the GitHub repository. The entire processing scheme can be easily adapted to the researcher's needs and their

raw data or data tables. The information about used packages and some troubleshooting procedures are described in "installed packages.docx" file. Other useful information, references and significant additions are listed below by each script file name.

To reproduce any script code, download all files to the working directory folder, set the working directory and load (install) the corresponding packages. The R Markdown document and reports files were provided as an example to reproduce the *OUKS* code script (available in "Report (Rmd)" folder).

The only requirements are to be familiar with the basic syntax of the R language, PC with Internet connection and Windows OS (desirable), RStudio and R (≥ 4.0.0).

OUKS has been published in the Journal of Proteome Research. If you use this software to analyze your own data, please cite it as below, thanks:

Ivan V. Plyushchenko, Elizaveta S. Fedorova, Natalia V. Potoldykova, Konstantin A. Polyakovskiy, Alexander I. Glukhov, and Igor A. Rodin Journal of Proteome Research 2022 21 (3), 833-847 DOI: https://doi.org/10.1021/acs.jproteome.1c00392.

Please send any comment, suggestion or question you may have to the author (Mr. Ivan Plyushchenko), email: plyushchenko.ivan@gmail.com.

1. Randomization

All samples were analyzed at random order to prevent systematic bias [1,2]. Each analytical batch consisted of ten samples in two technical repeats (repeat samples were acquired after last tenth sample and repeats were analyzed at the same order as first repeat). The QC samples were acquired at the beginning of the batch and after every five injections (overall five QC samples for each batch). The code generates a random sequence of samples in accordance with the user's conditions: the number of samples, technical and biological repeats and batch size.

2. Integration

In our case 311 injects in 13 batches were acquired. All raw data were stored in a specific folder. Seven injects were selected for integration and alignment parameters optimization via IPO [3] (1 (batch 1, 1st QC from 5), 75 (3, 5), 101 (5, 1), 163 (7, 3), 219 (9, 4), 257 (11, 2), 311 (13, 3 of 3)) in order to decrease time of computing. QC samples spanning all study were randomly selected (at the beginning, end and middle of batch).

Best parameters were selected for XCMS processing [4] according to IPO optimization. Only bw and minfrac parameters were manually checked according to [5,6]. The "bw" parameter did not introduce any benefits in terms of the number of peaks. The decreasing of "minfrac" parameter sequentially increase the number of peaks and "minfrac" was finally set equal to

minimum proportion of the smallest experimental group (0.2 for QC group). Peaks integration, grouping and retention time alignment were performed for all 311 sample injections (without blank injections). The final peak table was described by total number of peaks and fraction of missing values.

Warpgroup algorithm [7] for increasing of precision XCMS results integration was also tested. However, time of computing was approximately 6 days (test was performed via 5 random peak groups). The algorithm has not been implemented for this reason.

ncGTW algorithm [8] for increasing of precision XCMS alignment results was also implemented. Alternative way is checking different “bw “values.

Other approaches for XCMS parameters selection (Autotuner [9] in semi-automatic mode and MetaboAnalystR [10] in automatic) were also implemented as alternative options. Also, old version of XCMS functions (in consistent with xcmsSet objects instead of XCMSnExp objects in recent version) and new capabilities, that are provided by “refineChromPeaks” function and subset-based alignment, were integrated into the script.

3. Imputation

Artifacts checking was performed via MetProc [11] which is based on the comparison of missing rates (all peaks were retained). Nine methods of missing value imputation (MVI) were tested for previously obtained peak table, closely to [12]. The half minimum, KNN (k-nearest neighbors), PLS (partial least squares), PPCA (probabilistic principal component analysis, or other PCA-based), CART (Classification and Regression Tree), three types of RF (random forest) and QRILC (quantile regression for the imputation of left-censored missing data) algorithms were implemented for data table without any NA values and with randomly introducing NA values (proportion of introducing was equal to proportion of NA in the original dataset) [13]. The best algorithm was selected by the normalized root-mean-square error (NRMSE, value between 0 and 1) value close to zero (the RF implementation from missForest package). Assumption about decreasing of mean weighted NRMSE value after dividing the procedure for each group was sequentially tested and rejected (no statistically significant increment). Sum of squared error in Procrustes analysis on principal component scores (minimal value required) and correlation coefficient (maximum value required) were implemented as other quality metrics for MVI.

Moreover, other univariate MVI methods are implemented: converting NA values into the one single number (for example, 0), replacing by mean or median or minimum values and by random generated numbers. Values for random generation are produced by normal distribution with mean equal to the noise level (which was determined by IPO optimization or set manually) and standard deviation which was calculated from the noise (for example, 0.3*noise).

4. Correction

30 methods for signal drift correction and batch effects removal were implemented in this work, which include: model-based (WaveICA [14], WaveICA 2.0 [15], EigenMS [16], RUVSeqs family [17, 18] (RUVSeqs, RUVSeqg, RUVSeqr), Parametric/Non-Parametric Combat, Ber, Ber-Bagging [19]), internal standards based [20, 21] (SIS, NOMIS, CCMN, BMIS), QC metabolites based [21] (ruvrand, ruvrandclust), QC samples based regression algorithms are: cubic smoothing splines [22, 23] (in two implementations via pmp and notame packages), local polynomial regression fitting (LOESS) and RF [24], support vector machine (SVM) [25], least-squares/ robust (LM/RLM) and censored (TOBIT) [26], LOESS or splines with mclust [27], 5 new algorithms each with single, two features and batchwise modes (all also in subtraction/division versions) – KNN (caret package), decision tree (rpart), bagging tree (ipred), XGB (xgboost) and catboost (catboost) gradient boostings and QC-NORM for between batches correction [28]. QC-NORM was implemented in two versions: division-based [28] and subtraction-based [29]. Moreover, sequential combinations of methods without consideration of batch number and QC-NORM were also performed. De facto, random forest method is equal to bagging for the regression with only one variable, but due to some differences in trees growing and pruning, modeling results differ. Also, some other algorithms were tested: Cubist (Cubist package), conditional trees (partykit) and smoothing splines with auto-determination of knots (mgcv). Modeling results for these algorithms did not outperform existing approaches and were not included into the script.

5 new methods in single division mode were performed according to the equations (1-3):

$$M_i = f_{i,QC}\left(y_{i,QC} \sim x_{i,QC}\right) \quad (1)$$

$$F_i = M_i(x_i) \quad (2)$$

$$I_i' = \frac{I_i}{F_i} * 1000 \quad (3)$$

where, i – is the index of metabolite, $_{QC}$ – is the index of QC samples (all samples are denoted without this index), M – is a machine learning model, is fitted by function f, y – is an intensity vector (dependent variable), x – is a run order vector (independent variable), F – is a vector of predicted values by the model M (correction factor), I' – is a vector of a corrected intensity values, I – is a vector of an original intensity values.

Equations (1-3) are one of the basic algorithms for all QC sample based algorithms for the signal drift correction and are most similar to the statTarget package [24]. The MetNormalizer package [25] also performs clustering of features via Spearman correlation and the same correction operation (division). Correction algorithm, which is described by the eq. 1-3, was marked as single division mode (“d”).

$$I_i' = I_i - F_i + mean(y_{i,QC}) \quad (4)$$

where, i – is the index of metabolite, $_{QC}$ – is the index of QC samples (all samples are denoted without this index), y – is an intensity vector, F – is a vector of predicted values by the model M (correction factor), I' – is a vector of a corrected intensity values, I – is a vector of an original intensity values, *mean* – the function, that generates the mean value from the input vector.

Equation (4) is the modification for equation (3) of previously described algorithm (eq. 1-3), which is called single subtraction mode (eq. 1-2,4; "s"). The same mode was implemented in the BatchCorrMetabolomics [26] and notame [23] packages. Other mode in notame utilizes a correction factor with some changes (including prediction for the 1st QC sample and raw data).

Some QC sample based algorithms consider batch number. Batchwise mode ("bw") is identical to the two types of single mode (eq. 1-3 division and 1-2,4 subtraction), but each equation was repeated for each batch separately. This implementation is similar to pmp package [22] (the difference with division mode in the correction factor, which includes the median value of the feature) and is closely to batchCorr package [27] (the differences are in the correction factor and clustering features by the mclust algorithm).

Other way is an inclusion batch number into the regression model equation (into eq. 1, as in package BatchCorrMetabolomics [26]). This implementation was called Two Features mode ("tf") and was implemented both for subtraction and division modes.

Thus, totally 30 algorithms were proposed: each of 5 regression algorithms were implemented in 6 modes (3 (single, "bw", "tf") ×2 (division, subtraction)). All new correction methods in all modes were written with apply family functions and progress bar.

15 methods for the quality checking and comparison of corrections methods [26, 28, 30, 31] were performed, which include: guided principal component analysis (gPCA), principal variance component analysis (PVCA), fraction of features with p-value below the significance level in the ANOVA model, mean Pearson correlation coefficient for QC samples, mean Silhouette score in HCA on PCs scores for QC samples, fraction of features with p-value below the significance level in the One-Sample test (sequential implementation of Shapiro-Wilk normality test and corresponding type of test (t-test or Wilcoxon)) for QC samples, fraction of features under the criterion of 30% RSD reproducibility in QC samples, PCA, hierarchical cluster analysis (HCA), box-plots and scatter plot (total intensity versus run order) data projection and visualization, mean Bhattacharyya distance for QC samples between batches, mean dissimilarity score for QC samples, mean classification accuracy for QC samples after hierarchical clustering on PCs scores, Partial R-square (PC-PR2) method. The best correction method should demonstrate the maximum values of the fraction of features under the criterion of 30% RSD reproducibility in QC samples and the mean Pearson correlation coefficient for QC samples. The values of other

metrics should be the lowest. PCA, box-plots, scatter plots and HCA dendrogram should show the absence of tendency for the samples to cluster according to batch number or run order, and simultaneously QC samples should be placed almost between study samples.

5. Annotation

Features in peak table (the output of XCMS processing) were annotated via CAMERA [32], RAMClustR [33], xMSannotator [34] and mWISE [35]. The sr and st parameters in RAMClustR were optimized by functions from [5]. Other settings which include mass detection accuracy in ppm and m/z values and time deviation in seconds were derived from IPO optimized parameters and manual inspection respectively. Annotations should be performed for files in the same folder as for XCMS integration and alignment, xcms objects ("xcms obj … .RData") should be obtained by user. A search of isotope peaks, adducts, neutral losses and in-source fragments were included in all algorithms. The xMSannotator also was considered retention time and database search (HMDB, KEGG, LipidMaps, T3DB, and ChemSpider).

Some functions in xMSannotator package are not available for R version ≥ 4.0.0. For this reason, you should load file "fix for R 4.0 get_peak_blocks_modulesvhclust.R" in your environment (based on https://github.com/yufree/xMSannotator, other instructions are listed in the script). mWISE package forked from b2slab/mWISE to plyush1993/mWISE and depends were manually changed to R (>= 4.0) (available in "Auxiliary files (RData)" folder).

Each algorithm was characterized by a fraction of annotated features. The xMSannotator and mWISE provided the maximum value of annotated ions.

Also, metID [36] can be used for simple databases search from peak table. Databases from (https://github.com/jaspershen/demoData/tree/master/inst/ms2_database) should be copied in "ms2_database" folder in metID library folder.

6. Filtering

Several most common options are available [1, 23, 37]: RSD based filtering by QC samples, annotation based (remove non-annotated features), by descriptive statistics for biological samples (by mean, max, min, median values of the feature or sd), by the fraction of missing values in the feature for biological samples, by the cutoff between median values of feature in QS or biological samples and blanks injections and by the Pearson correlation coefficient for samples (mainly for repeated injections). The mean values for every feature for all repeats are also computed. The mean values for two repeats were obtained in our study.

7. Normalization

Five normalization methods are performed [38, 39]: mass spectrometry total useful signal (MSTUS), probabilistic quotient normalization (PQN), quantile, median fold change and sample-specific factor. Also, several scaling and transformation algorithms are introduced: log transformation, auto- , range- , pareto- , vast- , level- , power-, etc. scaling.

Biological and phenotyping metadata (classification class, age, sex, BMI, etc.) as well as experimental conditions (run order, batch number, etc.) can be adjusted by linear model (LM) [40] or linear mixed effect model (LMM) [41] fitting. Both algorithms were implemented in the study.

- The LMM approach fits the model according to a user-defined formula. In this study, model was constructed for each metabolite intensity as an dependent variable; age, class and sex – as fixed effects and batch ID – as a random effect. Then, the original data was employed as input and predicted values from the models were the final output.

  $$\Phi_i = f\left(I_i \sim \beta_{0,i} + \mu_0 + \sum_{j=1}^{n} \beta_{j,i} * x_j + \varepsilon_i\right) \quad (5)$$

  $$I_i' = \Phi_i(I_i) \quad (6)$$

  where, i – is the index of metabolite, j – is the index of independent variable $j = 1,2, \ldots n$ (fixed effects), Φ – is a LMM model is fitted by function f, β_0 – is the intercept constant, ε – is the random error, β – is the coefficient of the regression model, x – is an independent variable, I – is a dependent variable (a vector of an original intensity values), μ_0 – is the random effect, I' – is a vector of an adjusted intensity values, that was predicted by model Φ.

- The LM factorwise approach works in a similar way. At first, linear model was constructed for each metabolite intensity as a dependent variable and other covariates – as independent variables (Age, Sex, Batch, Class, Order in our study). One of the covariates should be noted as a feature of interest (this biological factor was kept, Class in our study). Then, residuals of models were obtained and design matrix was constructed for the feature of interest and constant intercept. For each model, design matrix was multiplied by regression coefficients of intercept and feature of interest and then was summed with residuals. Obtained values were summed with the difference between their mean values and mean values of the original data. The resulted values were the final output.

  $$\Phi_i = f\left(I_i \sim \beta_{0,i} + \sum_{j=1}^{n} \beta_{j,i} * x_j + \varepsilon_i\right) \quad (7)$$

  $$I_i' = (M_i * coef_i) + res_i \quad (8)$$

  $$I_i'' = I_i' + \left(mean(I_i) - mean(I_i')\right) \quad (9)$$

  where, i – is the index of metabolite, j – is the index of independent variable $j = 1,2, \ldots n$ (fixed effects), Φ – is a LM model is fitted by function f, β_0 – is the intercept constant, ε – is the

random error, β – is the coefficient of the regression model, x – is an independent variable, I – is a dependent variable (a vector of an original intensity values), I' – is a vector of a transformed intensity values, I'' – is a vector of an adjusted intensity value, M – is a design matrix, that was constructed for the feature of interest and constant intercept, *coef* – is a vector of regression coefficients of intercept and feature of interest from the model Φ, *res* – is a vector of the residuals of model Φ, *mean* – the function, that generates the mean value from the input vector.

- Other type of LM adjustment was implemented in similar way to LMM (eq. 5-6 without random effect).
- GAM, GAMM [42], GBM, GBMM [43] adjustment were also realized (as in eq. 5-6).

Four algorithms were utilized for evaluation of normalization methods, including: MA- , Box- , relative log abundance (RLA)– plots and mean/range of relative abundance between all features [21, 44, 45]. Biological adjustment methods were tested by classification accuracy via four machine learning (ML) algorithms (RF, linear SVM, nearest shrunken centroids (PAM), PLS), the high value of accuracy indicates successful removing of biological variation. ML parameters are listed in Section 9. PCA modeling was used for visualization of final adjustment. LM and LMM models were fitted in a similar way as described above (eq. 7,5). The fraction of p-values for the target variable and/or other covariates in the model less than the threshold (0.05) can serve as other character of the adjustment quality (the fraction should be the lowest for covariates and the largest for the target variable).

8. Grouping

Molecular features from data table after integration and alignment can be clustered (grouped) by different algorithms in order to determine an independent molecular feature and dimensionality reduction. The pmd package [46] performs hierarchical cluster analysis for this purpose (and also provides reactomics). The notame package [23] and CROP algorithm [47] utilize the Pearson correlation coefficient. Final signals represent the largest/mean/median values of features groups.

All .RData files that are needed to implement the CROP algorithm can be loaded from the GitHub repository.

9. Statistics

Statistical analysis part is divided into several logical blocks. Basic information and principles behind this chapter are available in [48-52]. Calculations are performed with basic functions, if no package is specified.

Outlier detection is performed by calculation of Mahalanobis/Euclidean distance for PCA scores (packages: OutlierDetection, pcaMethods or ClassDiscovery).

Statistical filtration block includes numerous methods for selection of informative variables:

- 1). A combination of four ML models with stable feature extraction (SFE) and recursive feature selection (RFS) [53] (packages: caret, tuple). Lists of the top N (50,100, etc) important features from each model were selected according to the model-specific metric in order to perform SFE. The unique variables are subsequently extracted from the set of important features from all models that are matched in at least *n* times (from 2 to 4 or equal to 50%-100% frequency). RFS with Naïve Bayes(NB) or RF algorithm is applied to subsets of variables after SFE for collection final variable set. The *N* and *n* are determined experimentally. Filtration can be performed by SFE only or SFE with RFS.

  The following parameters were applied when building the models: initial dataset was divided into two subsets by the 10 folds cross-validation with 10 times repeats for internal validation. The hyper-parameters tune length was 10. The four base ML algorithms were implemented: RF, SVM with radial kernel function, PLS and PAM. Only the most important value was tuned in each model (SVM – cost of constraints violation, RF – number of variables at each split, PLS – number of components, PAM – shrinkage threshold). All models were optimized by maximum mean accuracy metric across all cross-validation resampling. RFS was performed by backwards selection between all variables in subset. ML parameters in RFS were equal to those described above.
- 2). Filtration by VIP value from PLS (Orthogonal-PLS) model (package: ropls). The filtration threshold is determined experimentally.
- 3). Filtration by permutation importance from RF model (packages: permimp, party). The filtration threshold is determined experimentally.
- 4). Filtration by penalized or stepwise logistic regression models (packages: glmnet, MASS). The filtration threshold is determined experimentally.
- 5). Area Under Receiver Operating Characteristic (AUROC) calculations are computed in caret package. The trapezoidal rule is used to compute the area under the ROC curve for each predictor in binary classification. This area is used as the measure of variable importance and for filtration. The problem is decomposed into all pairwise tasks for multiclass outcomes and the area under the curve is calculated for each class pair. The mean value of AUROC across all predictors is directly calculated for binary classification task, for multiclass – at first, sum of AUROC for all groups pairs is determined and is divided by number of class labels and then

mean value of AUROC across all predictors is measured. The filtration threshold is determined experimentally.

- 6). Univariate filtering (UVF) is a subsequent implementation of Shapiro-Wilk normality test, Bartlett test of homoscedasticity and Wilcox, Welch, Student tests with Benjamini-Hochberg method for multiple comparison (significance level was set 5% by default). The feature is filtered if significant level is reached between two groups in binary classification and at least any two groups – in multilabel task.
- 7). Kruskal-Wallis and t-test can be performed separately. The feature is filtered if significant level is reached.
- 8). Filtration by fold change value. The value is set manually. Fold change calculation for multiclass dataset can be also performed.
- 9). Filtration by moderated t-test [54] (limma package). The significant level is set manually.
- 10). Filtration by LM and LMM models (eq. 7,5; packages: MetabolomicsBasics, lme4, lmerTest). The features are filtered by p-value (set manually) for the target variable.
- 11). Filtration by RUV2 algorithm ([55], package NormalizeMets). The p-value cutoff for the target variable is set manually.
- 12). Filtration by two-dimensional false discovery rate control [56].
- 13). Filtration by removing highly correlated features (package caret). The absolute values of pair-wise correlations are considered. If two variables have a high correlation, the function looks at the mean absolute correlation of each variable and removes the variable with the largest mean absolute correlation. The cutoff for pairwise absolute correlation is set manually.
- 14). Filtration by correlation coefficient. Each variable is compared with vector of target numeric variable. Correlation cutoff is set manually.

  Statistical filtration output can be adapted for any combination of filtration algorithms by selection of intersected variables or all unique features.

Another part of "Statistics" script is a classification model task. At first, dataset is divided into training and validation subsets. Then, resampling procedure is determined (bootstrap or cross-validation) and classification metric (ROC or accuracy) is specified (internal validation). In the next part ML model is fitted to training set via caret wrapper function. Hyperparameter tune length is set to 10 and can be changed. The resulted ML model is used for prediction on validation set (external validation) and resulted model overview with resampling statistics are provided. Also, logistic, penalized and stepwise logistic regression (packages: glmnet, MASS) are constructed in similar way. Performance of the models are tested by construction of confusion matrix on training and predicted class labels and ROC analysis. Generalized Weighted Quantile Sum (gWQS)

Regression [57] is also available as option for classification task. Features can be visualized via box-plot or scatter-plot by groups.

Regression model task is constructed in the same manner as classification task. The differences are as follows: metric is RMSE, MAE or R^2; performance is tested by MAE, RMSE, R^2 values; optimal subset of variables selection, penalized regression and stepwise regression are performed via leaps, glmnet and MASS packages; visualization is conducted by scatter plot of predicted and training data points. gWQS is also available for regression task.

Next block is testing of biomarker set. Features can be visualized via scatter plot by data points and box or violin plots by one or two groups. MANOVA and PERMANOVA (packages: vegan, pairwiseAdonis) procedures are available. Means comparison can be performed as: fold change calculation, t-/moderated t-/Wilcoxon/ Kruskal-Wallis tests, two-way ANOVA and one-way ANOVA (equal to UVF). In all cases, p-values are adjusted for multiple comparisons.

Metabolomic-Wide association study (MWAS) and analysis of covariance (ANCOVA) are carried out in the form of LM and LMM modeling (eq. 7,5) or similarly by GAM, GAMM and nonlinear dose-response curves (DRC). Also, LM, LMM, GLM, GLMM and correlation analysis are available (with Class as dependent variable) [58].

Signal modeling can be implemented by LM, LMM, GAM, GAMM and some other nonlinear functions for Dose-Response curve (DRC) analysis.

N-factor analysis is implemented by LM/LMM/GAM/GAMM/DRC modeling equally to MWAS/ANCOVA and Signal modeling case. ANOVA-simultaneous component analysis (ASCA, package MetStaT, [59]) is also accessible as an option for multiway analysis, two-way ANOVA, PLS, sPLS, two-dimensional false discovery rate control, PVCA and PC-PR2.

Analysis of repeated measures is performed by LM/LMM/GAM/GAMM/DRC modeling (eq. 7,5) and multilevel sPLS algorithm (package mixOmics, [60]).

Time series or longitudinal analysis is introduced by LM/LMM/GAM/GAMM/DRC modeling (eq. 7,5), ASCA and multivariate empirical Bayes statistics (package timecourse, [61]). Also, dose-response modeling is available (DRomics [62], TOXcms [63]), profile modeling by LMM Splines (timeOmics [64]) and PVCA, PC-PR2.

Multivariate data visualization and projection is provided by unsupervised data analysis. PCA, HCA, k-means clustering (all in packages: factoextra, FactoMineR, dendextend), HCA on PCA scores, heatmap (package pheatmap), t-distributed stochastic neighbor embedding (package Rtsne) and validation with optimization of clustering (Dunn index, silhouette analysis, gap stats, Rand index, p-value for HCA, classification accuracy, etc.; packages: NbClust, clustertend, mclust, clValid, fpc, pvclust) are available as an options for unsupervised data projection.

Correlation analysis is represented by computing correlation matrix and correlograms (packages: Hmisc, corrplot, psych).

Distance analysis part allows to calculate distance matrix for observations or features by distance metrics (Euclidean, maximum, Manhattan, Canberra, binary or Minkowski) or correlation and plot heatmap.

In the next section, effect size (Cohen's d and Hedges'g) and power analysis with sample size calculation are available (packages: effectsize, pwr).

Conclusion

The distribution of the number of strings in script file could be used for visualization of the relative contribution of each of nine processing steps into the overall workflow (Fig. 1).

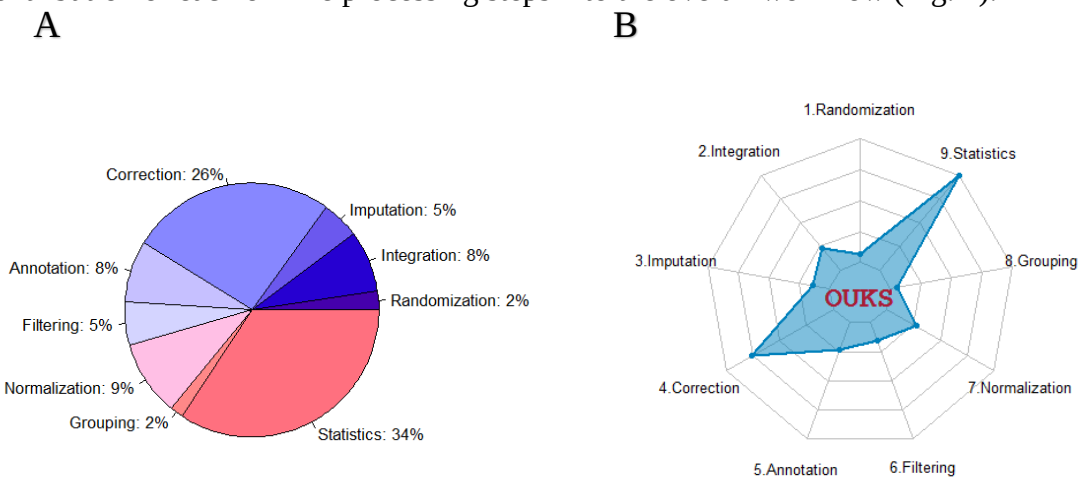

Fig. 1. Pie (A) and spider (B) charts for visualization of relative contribution of each of nine processing steps into overall workflow.

Notes

- In peak table after XCMS all batch index with label “b12_4” was renamed to “b13”.
- The klaR package should be exactly version 0.6-14 for RFS.
- The shapiro.wilk.test function from cwhmisc package should be used in case some error occurred with normality test in UVF and OST.
- caret wrapper function for ML training (“train” function) allows to fit 238 models (2019-03-27, http://topepo.github.io/caret/index.html). Some algorithms may require additional packages to be installed.

- Only one or none hyperparameter of ML models was tuned. For grid or random search of multiple hyperparameters tuning see: http://topepo.github.io/caret/model-training-and-tuning.html#basic-parameter-tuning.
- ML, XCMS, IPO, Warpgroup, ncGTW, mWISE RAMClustR, metID, RF MVI (missForest and StatTools implementations) and calculation of p-values for HCA can be accelerated via parallel processing inside R script.

References

1. Pezzatti, Julian, et al. "Implementation of liquid chromatography-high resolution mass spectrometry methods for untargeted metabolomic analyses of biological samples: A tutorial." Analytica Chimica Acta 1105 (2020): 28-44.

2. Dudzik, Danuta, et al. "Quality assurance procedures for mass spectrometry untargeted metabolomics. a review." Journal of pharmaceutical and biomedical analysis 147 (2018): 149-173.

3. Tautenhahn, Ralf, Christoph Boettcher, and Steffen Neumann. "Highly sensitive feature detection for high resolution LC/MS." BMC bioinformatics 9.1 (2008): 504.

4. Libiseller, Gunnar, et al. "IPO: a tool for automated optimization of XCMS parameters." BMC bioinformatics 16.1 (2015): 118.

5. Fernández-Ochoa, Álvaro, et al. "A Case Report of Switching from Specific Vendor-Based to R-Based Pipelines for Untargeted LC-MS Metabolomics." Metabolites 10.1 (2020): 28.

6. Albóniga, Oihane E., et al. "Optimization of XCMS parameters for LC–MS metabolomics: an assessment of automated versus manual tuning and its effect on the final results." Metabolomics 16.1 (2020): 14.

7. Mahieu, Nathaniel G., Jonathan L. Spalding, and Gary J. Patti. "Warpgroup: increased precision of metabolomic data processing by consensus integration bound analysis." Bioinformatics 32.2 (2016): 268-275.

8. Wu, Chiung-Ting, et al. "Targeted realignment of LC-MS profiles by neighbor-wise compound-specific graphical time warping with misalignment detection." Bioinformatics 36.9 (2020): 2862-2871.

9. McLean, Craig, and Elizabeth B. Kujawinski. "AutoTuner: high fidelity and robust parameter selection for metabolomics data processing." Analytical chemistry 92.8 (2020): 5724-5732.

10. Pang, Zhiqiang, et al. "MetaboAnalystR 3.0: Toward an optimized workflow for global metabolomics." Metabolites 10.5 (2020): 186.

11. Chaffin, Mark D., et al. "MetProc: separating measurement artifacts from true metabolites in an untargeted metabolomics experiment." Journal of proteome research 18.3 (2018): 1446-1450.

12. Wei, Runmin, et al. "Missing value imputation approach for mass spectrometry-based metabolomics data." Scientific reports 8.1 (2018): 1-10.

13. Di Guida, Riccardo, et al. "Non-targeted UHPLC-MS metabolomic data processing methods: a comparative investigation of normalisation, missing value imputation, transformation and scaling." Metabolomics 12.5 (2016): 93.

14. Deng, Kui, et al. "WaveICA: A novel algorithm to remove batch effects for large-scale untargeted metabolomics data based on wavelet analysis." Analytica chimica acta 1061 (2019): 60-69.

15 Deng, Kui, et al. "WaveICA 2.0: a novel batch effect removal method for untargeted metabolomics data without using batch information." Metabolomics 17.10 (2021): 1-8.

16. Karpievitch, Yuliya V., et al. "Metabolomics data normalization with EigenMS." PloS one 9.12 (2014): e116221.

17. Risso, Davide, et al. "Normalization of RNA-seq data using factor analysis of control genes or samples." Nature biotechnology 32.9 (2014): 896-902.

18. Marr, Sue, et al. "LC-MS based plant metabolic profiles of thirteen grassland species grown in diverse neighbourhoods." Scientific data 8.1 (2021): 1-12.

19. Bararpour, Nasim, et al. "DBnorm as an R package for the comparison and selection of appropriate statistical methods for batch effect correction in metabolomic studies." Scientific reports 11.1 (2021): 1-13.

20. Drotleff, Bernhard, and Michael Lämmerhofer. "Guidelines for selection of internal standard-based normalization strategies in untargeted lipidomic profiling by LC-HR-MS/MS." Analytical chemistry 91.15 (2019): 9836-9843.

21. Livera, Alysha M. De, et al. "Statistical methods for handling unwanted variation in metabolomics data." Analytical chemistry 87.7 (2015): 3606-3615.

22. Kirwan, J. A., et al. "Characterising and correcting batch variation in an automated direct infusion mass spectrometry (DIMS) metabolomics workflow." Analytical and bioanalytical chemistry 405.15 (2013): 5147-5157.

23. Klåvus, Anton, et al. "“Notame”: Workflow for Non-Targeted LC–MS Metabolic Profiling." Metabolites 10.4 (2020): 135.

24. Luan, Hemi, et al. "statTarget: A streamlined tool for signal drift correction and interpretations of quantitative mass spectrometry-based omics data." Analytica chimica acta 1036 (2018): 66-72.

25. Shen, Xiaotao, et al. "Normalization and integration of large-scale metabolomics data using support vector regression." Metabolomics 12.5 (2016): 89.

26. Wehrens, Ron, et al. "Improved batch correction in untargeted MS-based metabolomics." Metabolomics 12.5 (2016): 88.

27. Brunius, Carl, Lin Shi, and Rikard Landberg. "Large-scale untargeted LC-MS metabolomics data correction using between-batch feature alignment and cluster-based within-batch signal intensity drift correction." Metabolomics 12.11 (2016): 173.

28. Sánchez-Illana, Ángel, et al. "Evaluation of batch effect elimination using quality control replicates in LC-MS metabolite profiling." Analytica chimica acta 1019 (2018): 38-48.

29 Broadhurst, David, et al. "Guidelines and considerations for the use of system suitability and quality control samples in mass spectrometry assays applied in untargeted clinical metabolomic studies." Metabolomics 14.6 (2018): 1-17.

30. Caesar, Lindsay K., Olav M. Kvalheim, and Nadja B. Cech. "Hierarchical cluster analysis of technical replicates to identify interferents in untargeted mass spectrometry metabolomics." Analytica chimica acta 1021 (2018): 69-77.

31 Fages, Anne, et al. "Investigating sources of variability in metabolomic data in the EPIC study: the Principal Component Partial R-square (PC-PR2) method." Metabolomics 10.6 (2014): 1074-1083.

32. Kuhl, Carsten, et al. "CAMERA: an integrated strategy for compound spectra extraction and annotation of liquid chromatography/mass spectrometry data sets." Analytical chemistry 84.1 (2012): 283-289.

33. Broeckling, Corey David, et al. "RAMClust: a novel feature clustering method enables spectral-matching-based annotation for metabolomics data." Analytical chemistry 86.14 (2014): 6812-6817.

34. Uppal, Karan, Douglas I. Walker, and Dean P. Jones. "xMSannotator: an R package for network-based annotation of high-resolution metabolomics data." Analytical chemistry 89.2 (2017): 1063-1067.

35 Barranco-Altirriba, Maria, et al. "mWISE: An Algorithm for Context-Based Annotation of Liquid Chromatography–Mass Spectrometry Features through Diffusion in Graphs." Analytical Chemistry (2021).

36 Shen, Xiaotao, et al. "metID: an R package for automatable compound annotation for LC2MS-based data." Bioinformatics (2021).

37. Helmus, Rick, et al. "patRoon: open source software platform for environmental mass spectrometry based non-target screening." Journal of Cheminformatics 13.1 (2021): 1-25.

38. Wu, Yiman, and Liang Li. "Sample normalization methods in quantitative metabolomics." Journal of Chromatography A 1430 (2016): 80-95.

39. Li, Bo, et al. "Performance evaluation and online realization of data-driven normalization methods used in LC/MS based untargeted metabolomics analysis." Scientific reports 6 (2016): 38881.

40. António, Carla, ed. Plant metabolomics: Methods and protocols. Humana Press, 2018.

41. Wanichthanarak, Kwanjeera, et al. "Accounting for biological variation with linear mixed-effects modelling improves the quality of clinical metabolomics data." Computational and structural biotechnology journal 17 (2019): 611-618.

42 Wood, Simon N. Generalized additive models: an introduction with R. CRC press, 2017.

43 Sigrist, Fabio. "Gaussian Process Boosting." arXiv preprint arXiv:2004.02653 (2020).

44. Gromski, Piotr S., et al. "The influence of scaling metabolomics data on model classification accuracy." Metabolomics 11.3 (2015): 684-695.

45. Cuevas-Delgado, Paula, et al. "Data-dependent normalization strategies for untargeted metabolomics-a case study." Analytical and Bioanalytical Chemistry (2020): 1-15.

46. Yu, Miao, Mariola Olkowicz, and Janusz Pawliszyn. "Structure/reaction directed analysis for LC-MS based untargeted analysis." Analytica chimica acta 1050 (2019): 16-24.

47. Kouril, Stepán, et al. "CROP: Correlation-based reduction of feature multiplicities in untargeted metabolomic data." Bioinformatics 36.9 (2020): 2941-2942.

48. Li, Shuzhao, ed. Computational Methods and Data Analysis for Metabolomics. Humana Press, 2020.

49. Kuhn, Max, and Kjell Johnson. Applied predictive modeling. Vol. 26. New York: Springer, 2013.

50. James, Gareth, et al. An introduction to statistical learning. Vol. 112. New York: springer, 2013.

51. Lewis, Nigel Da Costa. 100 Statistical Tests in R: What to Choose, how to Easily Calculate, with Over 300 Illustrations and Examples. Heather Hills Press, 2013.

52. Kabacoff, Robert. R in Action. Shelter Island, NY, USA: Manning publications, 2011.

53. Plyushchenko, Ivan, et al. "An approach for feature selection with data modelling in LC-MS metabolomics." Analytical Methods 12.28 (2020): 3582-3591.

54. Smyth, Gordon K. "Limma: linear models for microarray data." Bioinformatics and computational biology solutions using R and Bioconductor. Springer, New York, NY, 2005. 397-420.

55. De Livera, Alysha M., et al. "Normalizing and integrating metabolomics data." Analytical chemistry 84.24 (2012): 10768-10776.

56 Yi, Sangyoon, et al. "2dFDR: a new approach to confounder adjustment substantially increases detection power in omics association studies." Genome biology 22.1 (2021): 1-18.

57. Tanner, Eva M., Carl-Gustaf Bornehag, and Chris Gennings. "Repeated holdout validation for weighted quantile sum regression." MethodsX 6 (2019): 2855-2860.

58 Rodriguez-Martinez, Andrea, et al. "MWASTools: an R/bioconductor package for metabolome-wide association studies." Bioinformatics 34.5 (2018): 890-892.

59. Smilde, Age K., et al. "ANOVA-simultaneous component analysis (ASCA): a new tool for analyzing designed metabolomics data." Bioinformatics 21.13 (2005): 3043-3048.

60. Liquet, Benoit, et al. "A novel approach for biomarker selection and the integration of repeated measures experiments from two assays." BMC bioinformatics 13.1 (2012): 1-14.

61. Tai, Yu Chuan, and Terence P. Speed. "A multivariate empirical Bayes statistic for replicated microarray time course data." The Annals of Statistics (2006): 2387-2412.

62 Larras, Floriane, et al. "DRomics: a Turnkey Tool to support the use of the dose–response framework for omics data in ecological risk assessment." Environmental science & technology 52.24 (2018): 14461-14468.

63. Yao, Cong-Hui, et al. "Dose-response metabolomics to understand biochemical mechanisms and off-target drug effects with the TOXcms software." Analytical chemistry 92.2 (2019): 1856-1864.

64. Bodein, Antoine, et al. "timeOmics: an R package for longitudinal multi-omics data integration." Bioinformatics (2021).