Package ‘pmp’

May 30, 2024

Type Package

Title Peak Matrix Processing and signal batch correction for metabolomics datasets

Version 1.16.0

Maintainer@R c(person(c(``Gavin",``Rhys"), ``Lloyd", role=c(``aut",``cre"), email=``g.r.lloyd@bham.ac.uk"))

Description Methods and tools for (pre-)processing of metabolomics datasets (i.e. peak matrices), including filtering, normalisation, missing value imputation, scaling, and signal drift and batch effect correction methods. Filtering methods are based on: the fraction of missing values (across samples or features); Relative Standard Deviation (RSD) calculated from the Quality Control (QC) samples; the blank samples. Normalisation methods include Probabilistic Quotient Normalisation (PQN) and normalisation to total signal intensity. A unified user interface for several commonly used missing value imputation algorithms is also provided. Supported methods are: k-nearest neighbours (knn), random forests (rf), Bayesian PCA missing value estimator (bpca), mean or median value of the given feature and a constant small value. The generalised logarithm (glog) transformation algorithm is available to stabilise the variance across low and high intensity mass spectral features. Finally, this package provides an implementation of the Quality Control-Robust Spline Correction (QCRSC) algorithm for signal drift and batch effect correction of mass spectrometry-based datasets.

License GPL-3

biocViews MassSpectrometry, Metabolomics, Software, QualityControl, BatchEffect

Depends R (>= 4.0)

Imports stats, impute, pcaMethods, missForest, ggplot2, methods, SummarizedExperiment, S4Vectors, matrixStats, grDevices, reshape2, utils

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests testthat, covr, knitr, rmarkdown, BiocStyle, gridExtra, magick

VignetteBuilder knitr

Collate 'checkPeakMatrix.R' 'utils.R' 'data.R' 'filters.R' 'glog_transformation.R' 'mv_imputation.R' 'normalisation.R' 'sbc_main.R' 'sbc_methods.R' 'sbc_plot.R'

git_url https://git.bioconductor.org/packages/pmp

git_branch RELEASE_3_19

git_last_commit d3489ab

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-29

Author Andris Jankevics [aut], Gavin Rhys Lloyd [aut, cre], Ralf Johannes Maria Weber [aut]

Maintainer Gavin Rhys Lloyd <g.r.lloyd@bham.ac.uk>

Contents

filter_peaks_by_blank . . . 2
filter_peaks_by_fraction . . . 4
filter_peaks_by_rsd . . . 5
filter_samples_by_mv . . . 6
glog_plot_optimised_lambda . . . 7
glog_transformation . . . 8
MTBLS79 . . . 9
mv_imputation . . . 10
normalise_to_sum . . . 12
pqn_normalisation . . . 13
processing_history . . . 14
QCRSC . . . 15
remove_peaks . . . 17
sbc_plot . . . 17

Index **19**

`filter_peaks_by_blank` *Filter features by blank samples*

Description

Metabolomics datasets often contain many features of non-biological origin e.g. those associated with extraction and analysis solvents. This tool facilitates the removal of such features from the data matrix, as defined using an appropriate blank sample.

Usage

```
filter_peaks_by_blank(
  df,
  fold_change,
  classes,
  blank_label,
  qc_label = NULL,
  remove_samples = TRUE,
  remove_peaks = TRUE,
  fraction_in_blank = 0
)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
fold_change	`numeric(1)`, fold_change minimum fold change between analytical and blank samples.
classes	`character()`, vector of class labels. Must be the same length as the number of sample in the input peak table. If input is `SummarizedExperiment` object, use `SummarizedExperiment_object$meta_data_column_name`.
blank_label	`character(1)`, class label used to identify blank samples.
qc_label	`character(1)` or `NULL`, class label used to identify QC samples.
remove_samples	`logical(1)`, remove blank samples from peak matrix or not.
remove_peaks	`logical(1)`, remove filtered features from peak matrix or not.
fraction_in_blank	`numeric(1)`, value between 0 to 1 to specify fraction in how many blanks peaks should be present.

Details

If parameter `qc_label` is not `NULL`, QC samples which will be used to calculate the median signal intensity.

Value

Object of class `SummarizedExperiment`. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns `numeric()` matrix-like object of filtered data set. Function `flags` are added to the object `attributes` and is a DataFrame-class with five columns. The same `DataFrame` object containing flags is added to `rowData()` element of `SummarizedExperiment` object as well.

Columns in `rowData()` or `flags` element contain:
`median_non_blanks` median intensities of features of non-blank samples;
`median_blanks` median intensities of features of blank samples;
`fold_change` fold change between analytical and blank samples;

`blank_flags` `integer()`, if 0 feature is flagged to be removed;
`blank_fraction_flags` `numeric()`, fraction in how many blank samples peaks is present.

Examples

```
df <- MTBLS79[ ,MTBLS79$Batch == 1]
df$Class[1:2] <- "Blank"
out <- filter_peaks_by_blank(df=df, fold_change=1.2,
   classes=df$Class, blank_label="Blank", qc_label=NULL,
   remove_samples=FALSE, remove_peaks=TRUE, fraction_in_blank=0)
```

`filter_peaks_by_fraction`

Filter features by fraction of missing values

Description

Metabolomics datasets often contain 'features' with irreproducible peak intensity values, or with large numbers of missing values. This tool facilitates the remove of such features from a data matrix, based upon the relative proportion (minimum fraction) of samples containing non-missing values.

Usage

```
filter_peaks_by_fraction(
  df,
  min_frac,
  classes = NULL,
  method = "QC",
  qc_label = "QC",
  remove_peaks = TRUE
)
```

Arguments

`df`	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
`min_frac`	`numeric(1)`, value between 0 and 1, a threshold of fraction of detection.
`classes`	`character()`, vector of class labels. Must be the same length as the number of sample in the input peak table. If input is `SummarizedExperiment` object, use `SummarizedExperiment_object$meta_data_column_name`.
`method`	`character(1)`, method to use. `QC` - within QC samples, `within` - within each sample class or `across` - across all samples.
`qc_label`	`character(1)` or `NULL`, class label used to identify QC samples.
`remove_peaks`	`logical(1)`, remove filtered features from peak matrix or not.

Value

Object of class `SummarizedExperiment`. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns `numeric()` matrix-like object of filtered data set. Function `flags` are added to the object `attributes` and is a DataFrame-class with five columns. The same `DataFrame` object containing flags is added to `rowData()` element of `SummarizedExperiment` object as well.

Columns in `rowData()` or `flags` element contain `fractions` of missing values per feature within QC samples (mehtod `QC`), across (method `across`) or within (mehtod `within`) each sample group.

Examples

```
df <- MTBLS79[ ,MTBLS79$Batch == 1]
out <- filter_peaks_by_fraction(df=df, min_frac=1,
    classes=df$Class, method='QC', qc_label='QC')

out <- filter_peaks_by_fraction(df=df, min_frac=1,
    classes=df$Class, method='across', qc_label='QC')

out <- filter_peaks_by_fraction(df=df, min_frac=1,
    classes=df$Class, method='within', qc_label='QC')
```

`filter_peaks_by_rsd` *Filter features by RSD% of QC samples*

Description

Metabolomics datasets often contain 'features' with irreproducible peak intensity values, or with large numbers of missing values. This tool facilitates the remove of such features from a data matrix, based upon relative standard deviation of intensity values for a given feature within specified QC samples.

Usage

```
filter_peaks_by_rsd(df, max_rsd, classes, qc_label, remove_peaks = TRUE)
```

Arguments

`df`	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
`max_rsd`	`numeric()`, threshold of QC RSD% value
`classes`	`character()`, vector of class labels. Must be the same length as the number of sample in the input peak table. If input is `SummarizedExperiment` object, use `SummarizedExperiment_object$meta_data_column_name`.

qc_label	character(1) or NULL, class label used to identify QC samples.
remove_peaks	logical(1), remove filtered features from peak matrix or not.

Value

Object of class SummarizedExperiment. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns numeric() matrix-like object of filtered data set. Function flags are added to the object attributes and is a DataFrame-class with five columns. The same DataFrame-class object containing flags is added to rowData() element of SummarizedExperiment object as well.

Columns in rowData() or flags element contain:
rsd_QC numeric(), RSD% value of QC samples per feature;
rsd_flags integer(),if 0 feature is flagged to be removed.

Examples

```
df <- MTBLS79[ ,MTBLS79$Batch == 1]
out <- filter_peaks_by_rsd(df=df, max_rsd=20,
    classes=df$Class, qc_label='QC')
```

---

filter_samples_by_mv *Filter samples by missing values*

Description

Missing values in mass spectrometry metabolomic datasets occur widely and can originate from a number of sources, including for both technical and biological reasons. In order for robust conclusions to be drawn from down-stream statistical testing procedures, the issue of missing values must first be addressed. This tool facilitates the removal of samples containing a user-defined maximum percentage of missing values.

Usage

```
filter_samples_by_mv(df, max_perc_mv, classes = NULL, remove_samples = TRUE)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class numeric() or integer() of peak intensities, areas or other quantitative characteristic.
max_perc_mv	numeric(1), Value between 0 and 1 of threshold of missing value percentage in sample.
classes	character(), vector of class labels. Must be the same length as the number of sample in the input peak table. If input is SummarizedExperiment object, use SummarizedExperiment_object$meta_data_column_name.
remove_samples	logical(1), remove blank samples from peak matrix or not.

Value

Object of class `SummarizedExperiment`. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns `numeric()` matrix-like object of filtered data set. Function `flags` are added to the object `attributes` and is a DataFrame-class with five columns. The same `DataFrame` object containing flags is added to `rowData()` element of `SummarizedExperiment` object as well. If element `colData()` already exists flags are appended to existing values.

Columns in `colData()` or `flags` element contain:
`perc_mv` `numeric()`, fraction of missing values per sample;
`flags` `integer()`,if 0 feature is flagged to be removed.

Examples

```
df <- MTBLS79
out <- filter_samples_by_mv (df=df, max_perc_mv=0.8)
```

---

glog_plot_optimised_lambda

Plot SSE error of lambda optimisation process

Description

Plot SSE error of lambda optimisation process

Usage

```
glog_plot_optimised_lambda(
  df,
  optimised_lambda,
  classes,
  qc_label,
  plot_grid = 100
)
```

Arguments

`df`	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
`optimised_lambda`	`numeric(1)`, value of optimised lambda from glog_transformation output.
`classes`	`character()`, vector of class labels. Must be the same length as the number of sample in the input peak table. If input is `SummarizedExperiment` object, use `SummarizedExperiment_object$meta_data_column_name`.

qc_label	character(1) or NULL, class label used to identify QC samples.
plot_grid	integer(1), number of data points to use for SSE optimisation.

Value

Class ggplot object containing optimisation plot.

Examples

```
data <- MTBLS79[, MTBLS79$Batch == 1]
classes <- data$Class

data <- mv_imputation(df=data, method='knn')
out <- glog_transformation (df=data, classes=classes,
    qc_label='QC')

optimised_lambda <- S4Vectors::metadata(out)
optimised_lambda <-
    optimised_lambda$processing_history$glog_transformation$lambda_opt

glog_plot_optimised_lambda(df=data, classes=classes,
    qc_label="QC", optimised_lambda=optimised_lambda)
```

---

glog_transformation *Variance stabilising generalised logarithm (glog) transformation*

Description

Performs glog transformation on the data set. QC samples can be used to estimate technical variation in the data set and calculate transformation parameter λ (lambda). QC samples usually comprise a pool of aliquots taken from all other samples in the study and analysed repeatedly throughout an analytical batch.

Usage

```
glog_transformation(df, classes, qc_label, lambda = NULL)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class numeric() or integer() of peak intensities, areas or other quantitative characteristic.
classes	character(), vector of class labels. Must be the same length as the number of sample in the input peak table. If input is SummarizedExperiment object, use SummarizedExperiment_object$meta_data_column_name.
qc_label	character(1) or NULL, class label used to identify QC samples.
lambda	NULL or numeric(1), if not NULL will use provided value for glog lambda.

Details

Many univariate and multivariate statistical tests require homogeneity and n ormality of dataset variance. Real-world metabolomics datasets often fail to meet these criteria due to asymmetric (i.e. non-'normal') and/or heteroscedatic (i.e. non-homogenous) variance structure. To address this issue, `glog` data transformations may be applied.
For each cell within the data matrix, transform the raw value (x) according to: $log10(x + sqrt(x^2 + \lambda))$. The parameter λ is typically calculated using quality control (QC) samples analysed throughout an analysis batch.

Value

Object of class `SummarizedExperiment`. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns the same R data structure as input with all value of data type `numeric()`.

References

Parsons HM et. al., BMC Bionf., 8(234), 2007. https://doi.org/10.1186/1471-2105-8-234

Examples

```
df <- MTBLS79[, MTBLS79$Batch == 1]
out <- mv_imputation(df=df, method="knn")
out <- glog_transformation (df=out, classes=df$Class,
    qc_label="QC")
```

MTBLS79 — *Direct-infusion mass spectrometry (DIMS) data set*

Description

Data set of 20 biological (cow vs sheep) serum samples that were analysed repeatedly, in 8 batches across 7 days.

Usage

```
MTBLS79
```

Format

A RangedSummarizedExperiment-class object.
`assay(MTBLS79)` Peak intensities of the DIMS data set. Contains 172 samples and 2488 features.
`colData(MTBLS79)` Sample meta data containing 4 columns.
`Batch` - `character()`, sample batch name.
`Sample_Rep` - `character()`, sample replicate code.
`Class` - `character()`, sample class labels.
`Class2` - `character()`, alternative sample class labels grouping together replicate samples.

Details

Code below includes all commands used to generate MTBLS79 object.

```
library (openxlsx)
library (SummarizedExperiment)

download.file(destfile = "MTBLS79.xlsx", mode="wb",
url = "ftp://ftp.ebi.ac.uk/pub/databases/metabolights/studies/public/
   MTBLS79/Dataset07__SFPM.xlsx")
wb <- openxlsx::loadWorkbook(xlsxFile="MTBLS79.xlsx")

MTBLS79 <- list()

MTBLS79$assay <- openxlsx::readWorkbook(wb, "data", colNames=T, rowNames=T)

# Last row of the peak matrix represent mean intensities across all samples.
MTBLS79$assay <- MTBLS79$assay[-c(nrow(MTBLS79$assay)), ]

# Transpose peak matrix, so that features are in rows and samples in columns.
MTBLS79$assay <- as.matrix(t(MTBLS79$assay))

# Missing values in the input data are stored as 0, replace with NA
MTBLS79$assay[MTBLS79$assay == 0] <- NA

rownames(MTBLS79$assay) <- round(as.numeric(rownames(MTBLS79$assay)), 5)

MTBLS79$colData <- openxlsx::readWorkbook(wb, "meta", colNames=T, rowNames=F)
MTBLS79$colData <- MTBLS79$colData[-c(nrow(MTBLS79$colData)), 1:4]

MTBLS79 <- SummarizedExperiment(assays=list(MTBLS79$assay),
colData=DataFrame(MTBLS79$colData))
```

Source

https://www.ebi.ac.uk/metabolights/MTBLS79

References

Kirwan et al, Scientific Data volume 1, Article number: 140012 (2014) https://www.nature.com/articles/sdata201412

`mv_imputation` *Missing value imputation using different algorithms*

Description

Missing values in metabolomics data sets occur widely and can originate from a number of sources, including technical and biological reasons.
Missing values imputation is applied to replace non-existing values with an estimated values while maintaining the data structure. A number of different methods are available as part of this function.

Usage

```
mv_imputation(
  df,
  method,
  k = 10,
  rowmax = 0.5,
  colmax = 0.5,
  maxp = NULL,
  check_df = TRUE
)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
method	`character(1)`, missing value imputation method. Supported methods are `knn`, `rf`, `bpca`, `sv`, `'mn'` and `'md'`.
k	`numeric(1)`, for a given sample containing a missing value, the number of nearest neighbours to include to calculate a replacement value. Used only for method `knn`.
rowmax	`numeric(1)`, the maximum percentage of missing data allowed in any row. For any rows exceeding given limit, missing values are imputed using the overall mean per sample. Used only for method `knn`.
colmax	`numeric(1)`, the maximum percent missing data allowed in any column. If any column exceeds given limit, the function will report an error Used only for method `knn`.
maxp	`integer(1)`, number of features to run on single core. If set to NULL will use total number of features.
check_df	`logical(1)`, if set to TRUE will check if input data needs to be transposed, so that features are in rows.

Details

Supported missing value imputation methods are:

`knn` - K-nearest neighbour. For each feature in each sample, missing values are replaced by the mean average value (non-weighted) calculated from its `k` closest neighbours in multivariate space (default distance metric: euclidean distance);

`rf` - Random Forest. This method is a wrapper of missForest function. For each feature, missing values are iteratively imputed until a maximum number of iterations (10), or until the difference between consecutively-imputed matrices becomes positive. Trees per forest are set to 100, variables included per tree are calculate using formula $sqrt(totalnumberofvariables)$;

`bpca` - Bayesian principal component analysis. This method is a wrapper of pca function. Missing values are replaced by the values obtained from principal component analysis regression with a Bayesian method. Therefore every imputed missing value does not occur multiple times, neither across the samples nor across the metabolite features;

`sv` - Small value. For each feature, replace missing values with half of the lowest value recorded in the entire data matrix;

`'mn'` - Mean. For each feature, replace missing values with the mean average (non-weighted) of all other non-missing values for that variable;

`'md'` - Median. For each feature, replace missing values with the median of all other non-missing values for that variable.

Value

Object of class `SummarizedExperiment`. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns the same R data structure as input with all value of data type `numeric()`.

Examples

```
df <- MTBLS79[ ,MTBLS79$Batch == 1]
out <- mv_imputation(df=df, method='knn')
```

`normalise_to_sum` *Normalisation by total sum of the features per sample*

Description

For each sample, every feature intensity value is divided by the total sum of all feature intensity values measured in that sample (`NA` values ignored by default), before multiplication by 100; the unit is %.

Usage

```
normalise_to_sum(df, check_df = TRUE)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
check_df	`logical(1)`, if set to TRUE will check if input data needs to be transposed, so that features are in rows.

Value

Object of class `SummarizedExperiment`. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns the same R data structure as input with all value of data type `numeric()`.

Examples

```
df <- MTBLS79[ ,MTBLS79$Batch == 1]
out <- normalise_to_sum (df=df)
```

---

pqn_normalisation *Probabilistic quotient normalisation (PQN)*

Description

For every feature the mean response is calculated across all QC samples. A reference vector is then generated. The median between the reference vector and every sample is computed obtaining a vector of coefficients related to each sample. Each sample is then divided by the median value of the vector of coefficients; this median value is different for each sample. This method was adapted by Dieterle et al. (2006) (see references). Its purpose is to take into account the concentration changes of some metabolite features that affect limited regions of the data.

Usage

```
pqn_normalisation(
  df,
  classes,
  qc_label,
  ref_mean = NULL,
  qc_frac = 0,
  sample_frac = 0,
  ref_method = "mean"
)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
classes	`character()`, vector of class labels. Must be the same length as the number of sample in the input peak table. If input is `SummarizedExperiment` object, use `SummarizedExperiment_object$meta_data_column_name`.
qc_label	`character(1)` or `NULL`, class label used to identify QC samples.
ref_mean	`numeric()` or `NULL`, Vector of reference mean values to use instead of calculating from QC sample group. If set to `NULL`, QC sample data will be used.
qc_frac	`numeric()` A value between 0 and 1 to indicate the minimum proportion of QC samples a feature must be present in for it to be included when computing the reference. Default `qc_frac = 0`.
sample_frac	`numeric()` A value between 0 and 1 to indicate the minimum proportion of samples a feature must be present in for it to be considered when computing the normalisation coefficients. Default `sample_frac = 0`.
ref_method	`character()` Method used to compute the reference from the QC samples. Default `ref_method = 'mean'`. Allowed values are "mean" or "median".

Value

Object of class `SummarizedExperiment`. If input data are matrix-like (e.g. an ordinary matrix, a data frame) object, the same R data structure as the input will be returned with all values of the data type.

`numeric()`.

References

Dieterle F. et al., Anal. Chem., 78(13), 2006. http://dx.doi.org/10.1021/ac051632c

Examples

```
df <- MTBLS79[ , MTBLS79$Batch==1]
pqn_normalisation(df=df,
    classes=df$Class, qc_label='QC')
```

---

processing_history *Return history of applied functions and argument from pmp package.*

Description

Return history of applied functions and argument from pmp package.

Usage

```
processing_history(df)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.

Value

List of function names and argument values.

Examples

```
df <- MTBLS79[ ,MTBLS79$Batch == 1]
df$Class[1:2] <- "Blank"
out <- filter_peaks_by_blank(df=df, fold_change=1.2,
   classes=df$Class, blank_label="Blank", qc_label=NULL,
   remove_samples=FALSE, remove_peaks=TRUE, fraction_in_blank=0)
processing_history(out)
```

QCRSC — *Quality Control-Robust Spline Correction (QC-RSC)*

Description

Implementation of Quality QC-RSC algorithm for signal drift and batch effect correction within/across a multi-batch direct infusion mass spectrometry (DIMS) and liquid chromatography mass spectrometry (LCMS) datasets. This version supports missing values, but requires at least 4 data point for quality control (QC) samples measured within each analytical batch. The smoothing parameter (spar) can be optimised using leave-one-out cross validation to avoid overfitting.

Usage

```
QCRSC(
  df,
  order,
  batch,
  classes,
  spar = 0,
  log = TRUE,
  minQC = 5,
  qc_label = "QC",
  spar_lim = c(-1.5, 1.5)
)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
order	`numeric()`, A numeric vector indicating the order in which samples were measured.
batch	`numeric()` or `character()`, a vector indicating the batch each sample was measured in. If only one batch was measured then all values should be set to 1
classes	`character()`, vector of class labels. Must be the same length as the number of sample in the input peak table. If input is `SummarizedExperiment` object, use `SummarizedExperiment_object$meta_data_column_name`.
spar	`numeric(1)`, Spline smoothing parameter. Should be in the range 0 to 1. If set to 0 it will be estimated using leave-one-out cross-validation.
log	`logical(1)`, to perform the signal correction fit on the log scaled data. Default is TRUE.
minQC	`integer(1)`, Minimum number of measured quality control (QC) samples required for signal correction within feature per batch. For features where signal was measured in less QC samples than threshold signal correction won't be applied.
qc_label	`character(1)` or `NULL`, class label used to identify QC samples.
spar_lim	A 2 element numeric vector containing the min and max values of spar when searching for an optimum. Default `spar_lim = c(-1.5,1.5)`

Value

Object of class `SummarizedExperiment`. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns the same R data structure as input with all value of data type `numeric()`.

Author(s)

Andris Jankevics <a.jankevics@bham.ac.uk>

References

Kirwan et al, Anal. Bioanal. Chem., 405 (15), 2013 https://dx.doi.org/10.1007/s00216-013-6856-7

Examples

```
classes <- MTBLS79$Class
batch <- MTBLS79$Batch
order <- c(1:ncol(MTBLS79))

out <- QCRSC(df = MTBLS79[1:10, ], order = order, batch = MTBLS79$Batch,
classes = MTBLS79$Class, spar = 0, minQC = 4)
```

remove_peaks — *Remove features from peak intensity matrix*

Description

Filter to remove features.

Usage

```
remove_peaks(df, rem_index)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
rem_index	`logical()`, vector containing `TRUE` vales for features to remove. Should be the same length as number of features in input data.

Value

Object of class `SummarizedExperiment`. If input data are a matrix-like (e.g. an ordinary matrix, a data frame) object, function returns the same R data structure as input with all value of data type `numeric()`.

Examples

```
df <- MTBLS79[ ,MTBLS79$Batch == 1]
rem_index <- vector(mode="logical",
    length=nrow(SummarizedExperiment::assay(df)))
rem_index[c(1, 20, 456, 789)] <- TRUE
out <- remove_peaks(df=df, rem_index=rem_index)
```

sbc_plot — *Plot QCRSC corrected outputs*

Description

Plot the output from signal batch correction for the selected or the first 100 features.

Usage

```
sbc_plot(
  df,
  corrected_df,
  classes,
  batch,
  indexes = NULL,
  qc_label = "QC",
  output = "sbcms_plots.pdf"
)
```

Arguments

df	A matrix-like (e.g. an ordinary matrix, a data frame) or RangedSummarizedExperiment-class object with all values of class `numeric()` or `integer()` of peak intensities, areas or other quantitative characteristic.
corrected_df	Output from QCRSC function.
classes	`character()`, vector of class labels. Must be the same length as the number of sample in the input peak table. If input is `SummarizedExperiment` object, use `SummarizedExperiment_object$meta_data_column_name`.
batch	`numeric()` or `character()`, a vector indicating the batch each sample was measured in. If only one batch was measured then all values should be set to 1
indexes	`numeric()`, a vector defining which features to plot. If set to `NULL` will plot the first 100.
qc_label	`character(1)` or `NULL`, class label used to identify QC samples.
output	`character()`, a filename of the output pdf file. Can include the path. If set to `NULL` output will be list object containing class `ggplot` plots.

Value

Pdf file or `list()` object `ggplot` class showing data before and after signal correction.

Examples

```
order <- c(1:ncol(MTBLS79))
data <- MTBLS79[1:10, ]

out <- QCRSC(df =data, order=order, batch=MTBLS79$Batch,
   classes=MTBLS79$Class, spar=0, minQC=4)
plots <- sbc_plot (df=data, corrected_df=out, classes=MTBLS79$Class,
   batch=MTBLS79$Batch, output=NULL)
```

Index

∗ **datasets**
MTBLS79, 9

DataFrame-class, *3*, *5–7*

filter_peaks_by_blank, 2
filter_peaks_by_fraction, 4
filter_peaks_by_rsd, 5
filter_samples_by_mv, 6

glog_plot_optimised_lambda, 7
glog_transformation, 8

missForest, *12*
MTBLS79, 9
mv_imputation, 10

normalise_to_sum, 12

pca, *12*
pqn_normalisation, 13
processing_history, 14

QCRSC, 15, *18*

RangedSummarizedExperiment-class, *3–9*, *11*, *13–18*
remove_peaks, 17

sbc_plot, 17