Package ‘MSPrep’

February 3, 2024

Title Package for Summarizing, Filtering, Imputing, and Normalizing Metabolomics Data

Version 1.12.0

Description Package performs summarization of replicates, filtering by frequency, several different options for imputing missing data, and a variety of options for transforming, batch correcting, and normalizing data.

URL https://github.com/KechrisLab/MSPrep

BugReports https://github.com/KechrisLab/MSPrep/issues

Depends R (>= 4.1.0)

Imports SummarizedExperiment, S4Vectors, pcaMethods (>= 1.24.0), crmn, preprocessCore, dplyr (>= 0.7), tidyR, tibble (>= 1.2), magrittr, rlang, stats, stringr, methods, missForest, sva, VIM,

Suggests BiocStyle, knitr, rmarkdown, testthat (>= 1.0.2)

VignetteBuilder knitr

LazyData false

NeedsCompilation no

License GPL-3

biocViews Metabolomics, MassSpectrometry, Preprocessing

Encoding UTF-8

RoxygenNote 7.1.2

git_url https://git.bioconductor.org/packages/MSPrep
git_branch RELEASE_3_18
git_last_commit 511dcda
git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-02-02
Author  Max McGrath [aut, cre],
        Matt Mulvahill [aut],
        Grant Hughes [aut],
        Sean Jacobson [aut],
        Harrison Pielke-lombardo [aut],
        Katerina Kechris [aut, cph, ths]

Maintainer  Max McGrath <max.mcgrath@ucdenver.edu>

R topics documented:

<table>
<thead>
<tr>
<th>Package</th>
<th>R Topics</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>COPD_131</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>msFilter</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>msImpute</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>msNormalize</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>MSPrep</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>msPrepare</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>msquant</td>
<td></td>
<td>14</td>
</tr>
<tr>
<td>msSummarize</td>
<td></td>
<td>14</td>
</tr>
</tbody>
</table>

Index  17

COPD_131  Example mass spectrometry dataset

Description
Data contains LC-MS metabolite analysis for samples from 131 subjects with 3 technical replicates per subject. The first three columns indicate "Mass" (mass-to-charge ratio), "Retention.Time", and "Compound.Name" for the 662 unique metabolites observed in the samples. The remaining columns indicate abundance for each of the 662 mass/retention-time combination for each subject/replicate combination.

Usage
data(COPD_131)

Format
Data frame containing 662 observations of 396 samples

Mass  Mass-to-charge ratio
Retention.Time  Retention-time

Compound.Name  Compound name for each mass/retention time combination

X10062C_1  The remaining columns indicate metabolite abundances found in each Subject/Replicate combination. Each column begins with an 'X', followed by the subject ID, and then the replicate (1, 2, or 3), each separated by '_'.


**msFilter**

**Source**


**References**


This data is available at the NIH Common Fund’s National Metabolomics Data Repository (NMDR) website, the Metabolomics Workbench, https://www.metabolomicsworkbench.org, where it has been assigned Project ID PR000438. The data can be accessed directly via its Project DOI: 10.21228/M8FC7C. This work is supported by NIH grant, U2C-DK119886.

**Examples**

data(COPD_131)

---

**msFilter**

*Function for filtering abundance data set.*

**Description**

Filters compounds to those found in specified proportion of samples.

**Usage**

msFilter(
    data,  
    filterPercent = 0.8,  
    compVars = c("mz", "rt"),  
    sampleVars = c("subject_id"),  
    colExtraText = NULL,  
    separator = NULL,  
    missingValue = NA,  
    returnToSE = FALSE,  
    returnToDF = FALSE
)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Data set as either a data frame or ‘SummarizedExperiment’.</td>
</tr>
<tr>
<td>filterPercent</td>
<td>Decimal value indicating filtration threshold. Compounds which are present</td>
</tr>
<tr>
<td></td>
<td>in fewer samples than the specified proportion will be removed.</td>
</tr>
<tr>
<td>compVars</td>
<td>Vector of the columns which identify compounds. If a ‘SummarizedExperiment’</td>
</tr>
<tr>
<td></td>
<td>is used for ‘data’, row variables will be used.</td>
</tr>
<tr>
<td>sampleVars</td>
<td>Vector of the ordered sample variables found in each sample column.</td>
</tr>
</tbody>
</table>
colExtraText  Any extra text to ignore at the beginning of the sample columns names. Unused for 'SummarizedExperiments'.
separator  Character or text separating each sample variable in sample columns. Unused for 'SummarizedExperiment'.
missingValue  Specifies the abundance value which indicates missing data. May be a numeric or 'NA'.
returnToSE  Logical value indicating whether to return as 'SummarizedExperiment'
returnToDF  Logical value indicating whether to return as data frame.

Value
A data frame or 'SummarizedExperiment' with filtered abundance data. Default return type is set to match the data input but may be altered with the 'returnToSE' or 'returnToDF' arguments.

Examples
# Load example data set, summarize replicates
data(msquant)

summarizedDF <- msSummarize(msquant,
                          compVars = c("mz", "rt"),
                          sampleVars = c("spike", "batch", "replicate", "subject_id"),
                          cvMax = 0.50,
                          minPropPresent = 1/3,
                          colExtraText = "Neutral_Operator_Dif_Pos_",
                          separator = "_",
                          missingValue = 1)

# Filter the dataset using a 80% filter rate
filteredDF <- msFilter(summarizedDF,
                        filterPercent = 0.8,
                        compVars = c("mz", "rt"),
                        sampleVars = c("spike", "batch", "subject_id"),
                        separator = "_")

msImpute  Function for imputing missing values in data.

Description
Replaces missing values with non-zero estimates calculated using a selected method.
Usage

msImpute(
  data,
  imputeMethod = c("halfmin", "b pca", "knn", "rf"),
  kKnn = 5,
  nPcs = 3,
  maxIterRf = 10,
  nTreeRf = 100,
  compoundsAsNeighbors = FALSE,
  compVars = c("mz", "rt"),
  sampleVars = c("subject_id"),
  colExtraText = NULL,
  separator = NULL,
  missingValue = NA,
  returnToSE = FALSE,
  returnToDF = FALSE
)

Arguments

data Data set as either a data frame or ‘SummarizedExperiment’.  
imputeMethod String specifying imputation method. Options are "halfmin" (half the minimum value), "b pca" (Bayesian PCA), and "knn" (k-nearest neighbors).  
kKnn Number of clusters for 'knn' method.  
nPcs Number of principle components used for re-estimation for 'b pca' method.  
maxIterRf Maximum number of iterations to be performed given the stopping criterion is not met beforehand for 'rf' method.  
nTreeRf Number of trees to grow in each forest for 'rf' method.  
compoundsAsNeighbors For KNN imputation. If TRUE, compounds will be used as neighbors rather than samples. Note that using compounds as neighbors is significantly slower than using samples.  
compVars Vector of the columns which identify compounds. If a ‘SummarizedExperiment’ is used for ‘data’, row variables will be used.  
sampleVars Vector of the ordered sample variables found in each sample column.  
colExtraText Any extra text to ignore at the beginning of the sample columns names. Unused for ‘SummarizedExperiments’.  
separator Character or text separating each sample variable in sample columns. Unused for ‘SummarizedExperiment’.  
missingValue Specifies the abundance value which indicates missing data. May be a numeric or ‘NA’.  
returnToSE Logical value indicating whether to return as ‘SummarizedExperiment’  
returnToDF Logical value indicating whether to return as data frame.
Value

A data frame or ‘SummarizedExperiment’ with missing data imputed. Default return type is set to match the data input but may be altered with the ‘returnToSE’ or ‘returnToDF’ arguments.

References


Examples

# Load, tidy, summarize, and filter example dataset
data(msquant)

summarizedDF <- msSummarize(msquant,
  compVars = c("mz", "rt"),
  sampleVars = c("spike", "batch", "replicate", "subject_id"),
  cvMax = 0.50,
  minPropPresent = 1/3,
  colExtraText = "Neutral_Operator_Dif_Pos_",
  separator = "_",
  missingValue = 1)

filteredDF <- msFilter(summarizedDF,
  filterPercent = 0.8,
  compVars = c("mz", "rt"),
  sampleVars = c("spike", "batch", "subject_id"),
  separator = "_")

# Impute dataset using 3 possible options
hmImputedDF <- msImpute(filteredDF, imputeMethod = "halfmin",
  compVars = c("mz", "rt"),
  sampleVars = c("spike", "batch", "subject_id"),
  separator = "_",
  missingValue = 0)

b pcaImputedDF <- msImpute(filteredDF, imputeMethod = "bpca",
  nPcs = 3,
  compVars = c("mz", "rt"),
  sampleVars = c("spike", "batch", "subject_id"),
  separator = "_",
  missingValue = 0)
knnImputedDF <- msImpute(filteredDF, imputeMethod = "knn",
  kKnn = 5,
  compVars = c("mz", "rt"),
  sampleVars = c("spike", "batch", "subject_id"),
  separator = " ",
  missingValue = 0)

msNormalize

Function for performing normalization and batch corrections on imputed data.

Description
Perform normalization and batch corrections on specified imputed dataset. Routines included are quantile, RUV (remove unwanted variation), SVA (surrogate variable analysis), median, CRMN (cross-contribution compensating multiple standard normalization), and ComBat to remove batch effects in raw, quantile, and median normalized data. Generates data driven controls if none exist.

Usage
msNormalize(
data,
normalizeMethod = c("median", "ComBat", "quantile", "quantile + ComBat",
  "median + ComBat", "CRMN", "RUV", "SVA"),
nControl = 10,
controls = NULL,
nComp = 2,
kRUV = 3,
batch = "batch",
covariatesOfInterest = NULL,
transform = c("log10", "log2", "ln", "none"),
compVars = c("mz", "rt"),
sampleVars = c("subject_id"),
colExtraText = NULL,
separator = NULL,
returnToSE = FALSE,
returnToDF = FALSE
)

Arguments
- **data**: Data set as either a data frame or ‘SummarizedExperiment’.
- **normalizeMethod**: Name of normalization method. "ComBat" (only ComBat batch correction), "quantile" (only quantile normalization), "quantile + ComBat" (quantile with ComBat batch correction), "median" (only median normalization), "median +
ComBat” (median with ComBat batch correction), "CRMN” (cross-contribution compensating multiple standard normalization), "RUV” (remove unwanted variation), “SVA” (surrogate variable analysis)

**nControl**
Number of controls to estimate/utilize (for CRMN and RUV).

**controls**
Vector of control identifiers. Leave blank for data driven controls. Vector of column numbers from metafin dataset of that control (for CRMN and RUV).

**nComp**
Number of factors to use in CRMN algorithm.

**kRUV**
Number of factors to use in RUV algorithm.

**batch**
Name of the sample variable identifying batch.

**covariatesOfInterest**
Sample variables used as covariates in normalization algorithms (required for ComBat, CRMN, and SVA).

**transform**
Select transformation to apply to data prior to normalization. Options are "log10", "log2", "ln" and "none".

**compVars**
Vector of the columns which identify compounds. If a ‘SummarizedExperiment’ is used for ‘data’, row variables will be used.

**sampleVars**
Vector of the ordered sample variables found in each sample column.

**colExtraText**
Any extra text to ignore at the beginning of the sample columns names. Unused for ‘SummarizedExperiments’.

**separator**
Character or text separating each sample variable in sample columns. Unused for ‘SummarizedExperiment’.

**returnToSE**
Logical value indicating whether to return as ‘SummarizedExperiment’

**returnToDF**
Logical value indicating whether to return as data frame.

**Value**
A data frame or ‘SummarizedExperiment’ with transformed and normalized data. Default return type is set to match the data input but may be altered with the ‘returnToSE’ or ‘returnToDF’ arguments.

**References**


Leek, J.T.et al.(2007) Capturing Heterogeneity in Gene Expression Studies by Surrogate Variable Analysis. PLoS Genetics, 3(9), e161

Examples

# Load, tidy, summarize, filter, and impute example dataset
data(msquant)

summarizedDF <- msSummarize(msquant,
    compVars = c("mz", "rt"),
    sampleVars = c("spike", "batch", "replicate", "subject_id"),
    cvMax = 0.50,
    minPropPresent = 1/3,
    colExtraText = "Neutral_Operator_Dif_Pos_",
    separator = ",",
    missingValue = 1)

filteredDF <- msFilter(summarizedDF,
    filterPercent = 0.8,
    compVars = c("mz", "rt"),
    sampleVars = c("spike", "batch", "subject_id"),
    separator = ",")

hmImputedDF <- msImpute(filteredDF,
    imputeMethod = "halfmin",
    compVars = c("mz", "rt"),
    sampleVars = c("spike", "batch", "subject_id"),
    separator = ",",
    missingValue = 0)

# Normalize data set
medianNormalizedDF <- msNormalize(hmImputedDF,
    normalizeMethod = "median",
    compVars = c("mz", "rt"),
    sampleVars = c("spike", "batch", "subject_id"),
    separator = ",")

MSPrep
 Package for summarizing, filtering, imputing, and normalizing metabolomics data.

Description

Package performs summarization of replicates, filtering by frequency, several different options for imputing missing data, and a variety of options for transforming, batch correcting, and normalizing data.

Details

Package for pre-analytic processing of mass spectrometry quantification data. Four functions are provided and are intended to be used in sequence (as a pipeline) to produce processed and normalized data. These are msSummarize(), msFilter(), msImpute(), and msNormalize(). The function
msPrepare() is also provided as a wrapper function combining the four previously mentioned functions.

Author(s)
Max McGrath
Matt Mulvahill
Grant Hughes
Sean Jacobson
Harrison Pielke-Lombardo
Katerina Kechris

References

Examples

```r
# Load example data
data(msquant)

# Call function to tidy, summarize, filter, impute, and normalize data
preparedDF <- msPrepare(msquant,
                        minPropPresent = 1/3,
```
msPrepare

Summarize, filter, impute, transform and normalize metabolomics dataset

Description

Wrapper function for the entire MSPrep pre-analytics pipeline. Calls msSummarize(), msFilter, msImpute(), and msNormalize().

Usage

```r
msPrepare(
data,
cvMax = 0.5,
minPropPresent = 1/3,
filterPercent = 0.8,
imputeMethod = c("halfmin", "b pca", "knn", "rf", "none"),
kKnn = 5,
nPcs = 3,
maxIterRf = 10,
nTreeRf = 100,
compoundsAsNeighbors = FALSE,
nControl = 10,
controls = NULL,
nComp = 2,
kRUV = 3,
covariatesOfInterest = NULL,
b atch = NULL,
transform = c("log10", "log2", "none"),
replicate = "replicate",
compVars = c("mz", "rt"),
sampleVars = c("subject_id"),
colExtraText = NULL,
separator = "_")
```
separator = NULL,
missingValue = NA,
returnSummaryDetails = FALSE,
returnToSE = FALSE,
returnToDF = FALSE
}

Arguments

data            Data set as either a data frame or 'SummarizedExperiment'.
cvMax            Decimal value from 0 to 1 representing the acceptable level of coefficient of variation between replicates.
minPropPresent   Decimal value from 0 to 1 representing the minimum proportion present to summarize with median or mean. Below this the compound will be set to 0.
filterPercent    Decimal value indicating filtration threshold. Compounds which are present in fewer samples than the specified proportion will be removed.
imputeMethod     String specifying imputation method. Options are "halfmin" (half the minimum value), "bPCA" (Bayesian PCA), and "knn" (k-nearest neighbors), or "none" to skip imputation.
kKnn             Number of clusters for 'knn' method.
nPcs             Number of principle components used for re-estimation for 'bPCA' method.
maxIterRf        Maximum number of iterations to be performed given the stopping criterion is not met beforehand for 'rf' method.
nTreeRf          Number of trees to grow in each forest for 'rf' method.
compoundsAsNeighbors
For KNN imputation. If TRUE, compounds will be used as neighbors rather than samples. Note that using compounds as neighbors is significantly slower than using samples.
normalizeMethod  Name of normalization method. "ComBat" (only ComBat batch correction), "quantile" (only quantile normalization), "quantile + ComBat" (quantile with ComBat batch correction), "median" (only median normalization), "median + ComBat" (median with ComBat batch correction), "CRMN" (cross-contribution compensating multiple standard normalization), "RUV" (remove unwanted variation), "SVA" (surrogate variable analysis), or "none" to skip normalization.
nControl        Number of controls to estimate/utilize (for CRMN and RUV).
controls         Vector of control identifiers. Leave blank for data driven controls. Vector of column numbers from metafin dataset of that control (for CRMN and RUV).
nComp            Number of factors to use in CRMN algorithm.
kRUV             Number of factors to use in RUV algorithm.
covariatesOfInterest
Sample variables used as covariates in normalization algorithms (required for ComBat, CRMN, and SVA).
batch            Name of the sample variable identifying batch.
**msPrepare**

- **transform**: Select transformation to apply to data prior to normalization. Options are "log10", "log2", and "none".
- **replicate**: Name of sample variable specifying replicate. Must match an element in `sampleVars` or a column in the column data of a `SummarizedExperiment`.
- **compVars**: Vector of the columns which identify compounds. If a `SummarizedExperiment` is used for `data`, row variables will be used.
- **sampleVars**: Vector of the ordered sample variables found in each sample column.
- **colExtraText**: Any extra text to ignore at the beginning of the sample columns names. Unused for `SummarizedExperiments`.
- **separator**: Character or text separating each sample variable in sample columns. Unused for `SummarizedExperiment`.
- **missingValue**: Specifies the abundance value which indicates missing data. May be a numeric or 'NA'.
- **returnSummaryDetails**: Logical value specifying whether to return details of replicate summarization.
- **returnToSE**: Logical value specifying whether to return as `SummarizedExperiment`.
- **returnToDF**: Logical value specifying whether to return as data frame.

**Value**

A data frame or `SummarizedExperiment` with summarized technical replicates (if present), filtered compounds, missing values imputed, and transformed and normalized abundances. Default return type is set to match the data input but may be altered with the `returnToSE` or `returnToDF` arguments.

**Examples**

```r
# Load example data
data(msquant)

# Call function to tidy, summarize, filter, impute, and normalize data
preparedData <- msPrepare(msquant, cvMax = 0.50, minPropPresent = 1/3,
                           filterPercent = 0.8, imputeMethod = "halfmin",
                           normalizeMethod = "quantile",
                           compVars = c("mz", "rt"),
                           sampleVars = c("spike", "batch", "replicate",
                                          "subject_id"),
                           colExtraText = "Neutral_Operator_Dif_Pos_-",
                           separator = "_", missingValue = 1,
                           returnToSE = FALSE)
```
Example mass spectrometry dataset.

Description

Data contains LC-MS samples for 2 subjects, each run with several different study design settings: spike-in (1x, 2x, 4x), batch (01, 02, 03), and technical replicate (A, B, C). The first two columns indicate mass-to-charge ratio and retention-time for the 2644 unique metabolites observed in the samples. The remaining 54 columns indicate metabolite abundance for each subject/spike-in/batch/replicate combination.

Usage

data(msquant)

Format

Data frame containing 2644 observations of 56 samples

- **mz** Mass-to-charge ratio
- **rt** Retention-time
- **Neutral_Operator_Dif_Pos_1x_O1_A_01** The remaining columns specify metabolite abundances found in each subject/spike-in/batch/replicate combination. Each columns begins with 'Neutral_Operator_Dif_Pos' followed by the spike-in (1x, 2x, or 4x), then the batch (01, 02, or 03), the replicate (A, B, or C), and finally the subject ID (01 or 02), each separated by '_'.

References


Examples

data(msquant)

Function for summarizing technical replicates.

Description

Reads data and summarizes technical replicates as the mean of observations for compounds found in 2 or 3 replicates and with coefficient of variation below specified level, or median for those found in 3 replicates but with excessive coefficient of variation (CV). Compounds found in only 1 replicate are assigned as missing.
Usage

```r
msSummarize(
  data,
  cvMax = 0.5,
  minPropPresent = 1/3,
  replicate = "replicate",
  compVars = c("mz", "rt"),
  sampleVars = c("subject_id"),
  colExtraText = NULL,
  separator = NULL,
  missingValue = NA,
  returnSummaryDetails = FALSE,
  returnToSE = FALSE,
  returnToDF = FALSE
)
```

Arguments

data: Data set as either a data frame or `SummarizedExperiment`

cvMax: Decimal value from 0 to 1 representing the acceptable level of coefficient of variation between replicates.

minPropPresent: Decimal value from 0 to 1 representing the minimum proportion present to summarize with median or mean. Below this the compound will be set to 0.

replicate: Name of sample variable specifying replicate. Must match an element in `sampleVars` or a column in the column data of a `SummarizedExperiment`.

compVars: Vector of the columns which identify compounds. If a `SummarizedExperiment` is used for `data`, row variables will be used.

sampleVars: Vector of the ordered sample variables found in each sample column.

colExtraText: Any extra text to ignore at the beginning of the sample columns names. Unused for `SummarizedExperiments`.

separator: Character or text separating each sample variable in sample columns. Unused for `SummarizedExperiment`.

missingValue: Specifies the abundance value which indicates missing data. May be a numeric or `NA`.

returnSummaryDetails: Logical value specifying whether to return details of replicate summarization.

returnToSE: Logical value specifying whether to return as `SummarizedExperiment`

returnToDF: Logical value specifying whether to return as data frame.

Value

A data frame or `SummarizedExperiment` containing abundance data with summarized technical replicates. Default return type is set to match the input but may be altered with the `returnToSE` or `returnToDF` arguments. If `returnSummaryDetails` is selected, a list will be returned containing the summarized data and a separate tidy data frame with summarization details included for each compound/sample pair.
Examples

# Read in data file
data(msquant)

# Summarize technical replicates
summarizedDF <- msSummarize(msquant,
    compVars = c("mz", "rt"),
    sampleVars = c("spike", "batch", "replicate",
        "subject_id"),
    cvMax = 0.50,
    minPropPresent = 1/3,
    colExtraText = "Neutral_Operator_Dif_Pos_",
    separator = "_",
    missingValue = 1)
Index

* datasets
  COPD_131, 2
  msquant, 14

COPD_131, 2

msFilter, 3
msImpute, 4
msNormalize, 7
MSPrep, 9
msPrepare, 11
msquant, 14
msSummarize, 14