Package: MSPrep (via r-universe)

October 24, 2026

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

Version 1.22.0

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

PackageStatus Deprecated

Config/pak/sysreqs make libicu-dev libpng-dev libxml2-dev libssl-dev zlib1g-dev

Repository Bioconductor 3.23

Date/Publication 2026-04-28 12:53:53 UTC

RemoteUrl https://github.com/bioc/MSPrep

RemoteRef RELEASE_3_23

RemoteSha 7f48077a8d4f76c48a6f418aee3db265cb3c3e45

Contents

COPD_131 2
msFilter 3
msImpute 4
msNormalize 7
MSPrep 9
msPrepare 11
msquant 14
msSummarize 14

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 '_'.

Source

https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Project&ProjectID=PR000438

References

Nichole Reisdorph (NaN). Untargeted LC-MS metabolomics analysis of human COPD plasma, HILIC & C18, metabolomics_workbench, V1.

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

`data`	Data set as either a data frame or 'SummarizedExperiement'.
`filterPercent`	Decimal value indicating filtration threshold. Compounds which are present in fewer samples than the specified proportion will be removed.
`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 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", "bpca", "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 'SummarizedExperiement'.
imputeMethod	String specifying imputation method. Options are "halfmin" (half the minimum value), "bpca" (Bayesian PCA), and "knn" (k-nearest neighbors).
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.
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

Oba, S.et al.(2003) A Bayesian missing value estimation for gene expression profile data. Bioinformatics, 19, 2088-2096

Stacklies, W.et al.(2007) pcaMethods A bioconductor package providing PCA methods for incomplete data. Bioinformatics, 23, 1164-1167.

A. Kowarik, M. Templ (2016) Imputation with R package VIM. Journal of Statistical Software, 74(7), 1-16.

Stekhoven D. J., & Buehlmann, P. (2012). MissForest - non-parametric missing value imputation for mixed-type data. Bioinformatics, 28(1), 112-118.

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)

bpcaImputedDF <- 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 'SummarizedExperiement'.
`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

Bolstad, B.M.et al.(2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics, 19, 185-193

DeLivera, A.M.et al.(2012) Normalizing and Integrating Metabolomic Data. Anal. Chem, 84, 10768-10776.

Gagnon-Bartsh, J.A.et al.(2012) Using control genes to correct for unwanted variation in microarray data. Biostatistics, 13, 539-552.

Johnson, W.E.et al.(2007) Adjusting batch effects in microarray expression data using Empirical Bayes methods. Biostatistics, 8, 118-127.

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

Wang, W.et al.(2003) Quantification of Proteins and Metabolites by Mass Spectrometry without Isotopic Labeling or Spiked Standards. Anal. Chem., 75, 4818-4826.

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

Bolstad, B.M.et al.(2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics, 19, 185-193

DeLivera, A.M.et al.(2012) Normalizing and Integrating Metabolomic Data. Anal. Chem, 84, 10768-10776.

Gagnon-Bartsh, J.A.et al.(2012) Using control genes to correct for unwanted variation in microarray data. Biostatistics, 13, 539-552.

Hughes G, Cruickshank-Quinn C, Reisdorph R, Lutz S, Petrache I, Reisdorph N, Bowler R, Kechris K. MSPrep–Summarization, normalization and diagnostics for processing of mass spectrometry-based metabolomic data. Bioinformatics. 2014;30(1):133-4. Epub 2013/11/01. doi: 10.1093/bioinformatics/btt589. PubMed PMID: 24174567; PMCID: PMC3866554.

Johnson, W.E.et al.(2007) Adjusting batch effects in microarray expression data using Empirical Bayes methods. Biostatistics, 8, 118-127.

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

Oba, S.et al.(2003) A Bayesian missing value estimation for gene expression profile data. Bioinformatics, 19, 2088-2096

Redestig, H.et al.(2009) Compensation for Systematic Cross-Contribution Improves Normalization of Mass Spectrometry Based Metabolomics Data. Anal. Chem., 81, 7974-7980.

Stacklies, W.et al.(2007) pcaMethods: A bioconductor package providing PCA methods for incomplete data. Bioinformatics, 23, 1164-1167.

Wang, W.et al.(2003) Quantification of Proteins and Metabolites by Mass Spectrometry without Isotopic Labeling or Spiked Standards. Anal. Chem., 75, 4818-4826.

Examples

```
# Load example data
data(msquant)

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

```
  missingValue = 1,
  filterPercent = 0.8,
  imputeMethod = "knn",
  normalizeMethod = "quantile + ComBat",
  transform = "log10",
  covariatesOfInterest = c("spike"),
  compVars = c("mz", "rt"),
  sampleVars = c("spike", "batch", "replicate",
                 "subject_id"),
  colExtraText = "Neutral_Operator_Dif_Pos_",
  separator = "_")
```

`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

```
msPrepare(
  data,
  cvMax = 0.5,
  minPropPresent = 1/3,
  filterPercent = 0.8,
  imputeMethod = c("halfmin", "bpca", "knn", "rf", "none"),
  kKnn = 5,
  nPcs = 3,
  maxIterRf = 10,
  nTreeRf = 100,
  compoundsAsNeighbors = FALSE,
  normalizeMethod = c("median", "ComBat", "quantile", "quantile + ComBat",
    "median + ComBat", "CRMN", "RUV", "SVA", "none"),
  nControl = 10,
  controls = NULL,
  nComp = 2,
  kRUV = 3,
  covariatesOfInterest = NULL,
  batch = NULL,
  transform = c("log10", "log2", "none"),
  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 'SummarizedExperiement'.
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.

`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

```
# Load example data
data(msquant)

# Call function to tidy, summarize, filter, impute, and normalize data
peparedData <- 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)
```

msquant *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 seperated by '_'.

References

Hughes, G., Cruickshank-Quinn, C., Reisdorph, R., Lutz, S., Petrache, I., Reisdorph, N., ... Kechris, K. (2014). MSPrep–summarization, normalization and diagnostics for processing of mass spectrometry-based metabolomic data. Bioinformatics (Oxford, England), 30(1), 133–134. doi:10.1093/bioinformatics/btt589

Examples

```
data(msquant)
```

msSummarize *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

```
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 'SummarizedExperiement'.
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 data nput 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