Package 'msImpute'

November 12, 2026

Type Package

Title Imputation of label-free mass spectrometry peptides

Version 1.22.0

Description MsImpute is a package for imputation of peptide intensity in proteomics experiments. It additionally contains tools for MAR/MNAR diagnosis and assessment of distortions to the probability distribution of the data post imputation. The missing values are imputed by low-rank approximation of the underlying data matrix if they are MAR (method = ``v2''), by Barycenter approach if missingness is MNAR (``v2-mnar''), or by Peptide Identity Propagation (PIP).

Depends R (> 4.1.0)

SystemRequirements python

Imports softImpute, methods, stats, graphics, pdist, reticulate, scran, data.table, FNN, matrixStats, limma, mvtnorm, tidyr, dplyr

License GPL (>=2)

Encoding UTF-8

LazyData true

BugReports https://github.com/DavisLaboratory/msImpute/issues

RoxygenNote 7.1.1

Suggests BiocStyle, knitr, rmarkdown, ComplexHeatmap, imputeLCMD

VignetteBuilder knitr

biocViews MassSpectrometry, Proteomics, Software

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

git_branch RELEASE_3_23

git_last_commit c3ac90e

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Soroor Hediyeh-zadeh [aut, cre] (ORCID:
<https://orcid.org/0000-0001-7513-6779>)

Maintainer Soroor Hediyeh-zadeh <hediyehzadeh.s@wehi.edu.au>

Contents

computeStructuralMetrics 2
CPD 4
evidenceToMatrix 4
findVariableFeatures 6
KNC 7
KNN 7
msImpute 8
mspip 10
plotCV2 12
pxd007959 13
pxd010943 13
pxd014777 14
scaleData 15
selectFeatures 17

Index **20**

`computeStructuralMetrics` *Metrics for the assessment of post-imputation structural preservation*

Description

For an imputed dataset, it computes within phenotype/experimental condition similarity (i.e. preservation of local structures), between phenotype distances (preservation of global structures), and the Gromov-Wasserstein (GW) distance between original (source) and imputed data.

Usage

```
computeStructuralMetrics(x, group = NULL, y = NULL, k = 2)
```

Arguments

`x`	numeric matrix. An imputed data matrix of log-intensity.
`group`	factor. A vector of biological groups, experimental conditions or phenotypes (e.g. control, treatment).
`y`	numeric matrix. The source data (i.e. the original log-intensity matrix), preferably subsetted on highly variable peptides (see `findVariableFeatures`).
`k`	numeric. Number of Principal Components used to compute the GW distance. default to 2.

Details

For each group of experimental conditions (e.g. treatment and control), the group centroid is calculated as the average of observed peptide intensities. Withinness for each group is computed as sum of the squared distances between samples in that group and the group centroid. Betweenness is computed as sum of the squared distances between group centroids. When comparing imputation approaches, the optimal imputation strategy should minimize the within group distances, hence smaller withinness, and maximizes between group distances, hence larger betweenness. The GW metric considers preservation of both local and global structures simultaneously. A small GW distance suggests that imputation has introduced small distortions to global and local structures overall, whereas a large distance implies significant distortions. When comparing two or more imputation methods, the optimal method is the method with smallest GW distance. The GW distance is computed on Principal Components (PCs) of the source and imputed data, instead of peptides. Principal components capture the geometry of the data, hence GW computed on PCs is a better measure of preservation of local and global structures. The PCs in the source data are recommended to be computed on peptides with high biological variance. Hence, users are recommended to subset the source data only on highly variable peptides (hvp) (see `findVariableFeatures`). Since the hvp peptides have high biological variance, they are likely to have enough information to discriminate samples from different experimental groups. Hence, PCs computed on those peptides should be representative of the original source data with missing values. If the samples cluster by experimental group in the first couple of PCs, then a choice of k=2 is reasonable. If the desired separation/clustering of samples occurs in later PCs (i.e. the first few PCs are dominated by batches or unwanted variability), then it is recommended to use a larger number of PCs to compute the GW metric. If you are interested in how well the imputed data represent the original data in all possible dimensions, then set k to the number of samples in the data (i.e. the number of columns in the intensity matrix). GW distance estimation requires `python`. See example. All metrics are on log scale.

Value

list of three metrics: withinness (sum of squared distances within a phenotype group), betweenness (sum of squared distances between the phenotypes), and gromov-wasserstein distance (if `xna` is not NULL). if `group` is NULL only the GW distance is returned. All metrics are on log scale.

References

Hediyeh-zadeh, S., Webb, A. I., & Davis, M. J. (2020). MSImpute: Imputation of label-free mass spectrometry peptides by low-rank approximation. bioRxiv.

Examples

```
data(pxd010943)
y <- log2(data.matrix(pxd010943))
y <- y[complete.cases(y),]
group <- as.factor(gsub("_[1234]", "", colnames(y)))
computeStructuralMetrics(y, group, y=NULL)
```

CPD *CPD*

Description

Spearman correlation between pairwise distances in the original data and imputed data. CPD quantifies preservation of the global structure after imputation. Requires complete datasets - for developers/use in benchmark studies only.

Usage

```
CPD(xorigin, ximputed)
```

Arguments

xorigin	numeric matrix. The original log-intensity data. Can not contain missing values.
ximputed	numeric matrix. The imputed log-intensity data. Can not contain missing values.

Value

numeric

Examples

```
data(pxd007959)
y <- pxd007959$y
y <- y[complete.cases(y),]
# for demonstration we use same y for xorigin and ximputed
CPD(y, y)
```

evidenceToMatrix *Creates intensity matrix from tabular data in evidence table of MaxQuant*

Description

Every `Modified sequence - Charge` is considered as a precursor feature. Only the feature with maximum intensity is retained. The columns are run names, the rows are peptide ids (in the `Modified.sequence_Charge` format)

Usage

```
evidenceToMatrix(
  evidence,
  run_id = "Raw.file",
  peptide_id = "PeptideID",
  return_EList = FALSE,
  weights = NULL
)
```

Arguments

evidence	data.frame. The evidence table read from evidence.txt, or data.frame created by mspip.
run_id	character. The name of the column of evidence containing the run/raw file name. These form the columns of the intensity data matrix.
peptide_id	character. The name of the column of evidence containing the peptide ids. These form the rows of the intensity data matrix.
return_EList	logical. If TRUE, returns a EListRaw object storing both the intensity data matrix and observation-level weights from mspip (propagation confidence score), otherwise returns a matrix.
weights	character. The name of the column of evidence containing weights from mspip. default to NULL. Set this to "weight" if you want the weights from PIP stored in the weights slot of the EListRaw object.

Details

The EListRaw object created by the function is intended to bridge msImpute and statistical methods of limma. The object can be passed to normalizeBetweenArrays for normalisation, which can then be passed to lmFit and eBayes for fitting linear models per peptide and Empirical Bayes moderation of t-statistics respectively. The weights slot is recognized by lmFit, which incorporates the uncertainty in intensity values inferred by PIP into the test statistic. The function is also a generic tool to create a matrix or limma-compatible objects from the evidence table of MaxQuant.

Value

a numeric matrix of intensity data, or a EListRaw object containing such data and observation-level weights from mspip.

Author(s)

Soroor Hediyeh-zadeh

See Also

mspip

`findVariableFeatures` *Find highly variable peptides*

Description

For each peptide, the total variance is decomposed into biological and technical variance using package `scran`

Usage

```
findVariableFeatures(y)
```

Arguments

`y`	numeric matrix giving log-intensity. Can contain NA values. Peptides with insufficient observations will be ignored.

Details

A loess trend is fitted to total sample variances and mean intensities. For each peptide, the biological variance is then computed by subtracting the estimated technical variance from the loess fit from the total sample variance.

Value

A data frame where rows are peptides and columns contain estimates of biological and technical variances. Peptides are ordered by biological variance.

See Also

computeStructuralMetrics

Examples

```
data(pxd007959)
findVariableFeatures(data.matrix(log2(pxd007959$y)))
```

KNC *k-nearest class means (KNC)*

Description

The fraction of k-nearest class means in the original data that are preserved as k-nearest class means in imputed data. KNC quantifies preservation of the mesoscopic structure after imputation. Requires complete datasets - for developers/use in benchmark studies only.

Usage

```
KNC(xorigin, ximputed, class, k = 3)
```

Arguments

xorigin	numeric matrix. The original log-intensity data. Can contain missing values.
ximputed	numeric matrix. The imputed log-intensity data.
class	factor. A vector of length number of columns (samples) in the data specifying the class/label (i.e. experimental group) of each sample.
k	number of nearest class means. default to k=3.

Value

numeric The proportion of preserved k-nearest class means in imputed data.

Examples

```
data(pxd007959)
y <- pxd007959$y
y <- y[complete.cases(y),]
# for demonstration we use same y for xorigin and ximputed
KNC(y, y, class = as.factor(pxd007959$samples$group))
```

KNN *k-nearest neighbour (KNN)*

Description

The fraction of k-nearest neighbours in the original data that are preserved as k-nearest neighbours in imputed data. KNN quantifies preservation of the local, or microscopic structure. Requires complete datasets - for developers/use in benchmark studies only.

Usage

```
KNN(xorigin, ximputed, k = 3)
```

Arguments

xorigin	numeric matrix. The original log-intensity data. Can not contain missing values.
ximputed	numeric matrix. The imputed log-intensity data. Can not contain missing values.
k	number of nearest neighbours. default to k=3.

Value

numeric The proportion of preserved k-nearest neighbours in imputed data.

Examples

```
data(pxd007959)
y <- pxd007959$y
y <- y[complete.cases(y),]
# for demonstration we use same y for xorigin and ximputed
KNN(y, y)
```

---

msImpute *Imputation of peptide log-intensity in mass spectrometry label-free proteomics by low-rank approximation*

Description

Returns a completed matrix of peptide log-intensity where missing values (NAs) are imputated by low-rank approximation of the input matrix. Non-NA entries remain unmodified. msImpute requires at least 4 non-missing measurements per peptide across all samples. It is assumed that peptide intensities (DDA), or MS1/MS2 normalised peak areas (DIA), are log2-transformed and normalised (e.g. by quantile normalisation).

Usage

```
msImpute(
  y,
  method = c("v2-mnar", "v2", "v1"),
  group = NULL,
  a = 0.2,
  rank.max = NULL,
  lambda = NULL,
  thresh = 1e-05,
  maxit = 100,
  trace.it = FALSE,
  warm.start = NULL,
  final.svd = TRUE,
  biScale_maxit = 20,
  gauss_width = 0.3,
```

```
  gauss_shift = 1.8
)
```

Arguments

y	Numeric matrix giving log-intensity where missing values are denoted by NA. Rows are peptides, columns are samples.
method	character. Allowed values are "v2" for msImputev2 imputation (enhanced version) for MAR. method="v2-mnar" (modified low-rank approx for MNAR), and "v1" initial release of msImpute
group	character or factor vector of length ncol(y)
a	numeric. the weight parameter. default to 0.2. Weights the MAR-imputed distribution in the imputation scheme.
rank.max	Numeric. This restricts the rank of the solution. is set to min(dim(y)-1) by default in "v1".
lambda	Numeric. Nuclear-norm regularization parameter. Controls the low-rank property of the solution to the matrix completion problem. By default, it is determined at the scaling step. If set to zero the algorithm reverts to "hardImputation", where the convergence will be slower. Applicable to "v1" only.
thresh	Numeric. Convergence threshold. Set to 1e-05, by default. Applicable to "v1" only.
maxit	Numeric. Maximum number of iterations of the algorithm before the algorithm is converged. 100 by default. Applicable to "v1" only.
trace.it	Logical. Prints traces of progress of the algorithm. Applicable to "v1" only.
warm.start	List. A SVD object can be used to initialize the algorithm instead of random initialization. Applicable to "v1" only.
final.svd	Logical. Shall final SVD object be saved? The solutions to the matrix completion problems are computed from U, D and V components of final SVD. Applicable to "v1" only.
biScale_maxit	number of iteration for the scaling algorithm to converge . See scaleData. You may need to change this parameter only if you're running method=v1. Applicable to "v1" only.
gauss_width	numeric. The width parameter of the Gaussian distribution to impute the MNAR peptides (features). This the width parameter in the down-shift imputation method.
gauss_shift	numeric. The shift parameter of the Gaussian distribution to impute the MNAR peptides (features). This the width parameter in the down-shift imputation method.

Details

msImpute operates on the softImpute-als algorithm in softImpute package. The algorithm estimates a low-rank matrix (a smaller matrix than the input matrix) that approximates the data with a reasonable accuracy. SoftImpute-als determines the optimal rank of the matrix through the lambda parameter, which it learns from the data. This algorithm is implemented in method="v1". In v2 we have used a information theoretic approach to estimate the optimal rank, instead of relying on softImpute-als defaults. Similarly, we have implemented a new approach to estimate lambda

from the data. Low-rank approximation is a linear reconstruction of the data, and is only appropriate for imputation of MAR data. In order to make the algorithm applicable to MNAR data, we have implemented `method="v2-mnar"` which imputes the missing observations as weighted sum of values imputed by msImpute v2 (`method="v2"`) and random draws from a Gaussian distribution. Missing values that tend to be missing completely in one or more experimental groups will be weighted more (shrunken) towards imputation by sampling from a Gaussian parameterised by smallest observed values in the sample (similar to minProb, or Perseus). However, if the missing value distribution is even across the samples for a peptide, the imputed values for that peptide are shrunken towards low-rank imputed values. The judgment of distribution of missing values is based on the EBM metric implemented in `selectFeatures`, which is also a information theory measure.

Value

Missing values are imputed by low-rank approximation of the input matrix. If input is a numeric matrix, a numeric matrix of identical dimensions is returned.

Author(s)

Soroor Hediyeh-zadeh

References

Hastie, T., Mazumder, R., Lee, J. D., & Zadeh, R. (2015). Matrix completion and low-rank SVD via fast alternating least squares. The Journal of Machine Learning Research, 16(1), 3367-3402.

Hediyeh-zadeh, S., Webb, A. I., & Davis, M. J. (2020). MSImpute: Imputation of label-free mass spectrometry peptides by low-rank approximation. bioRxiv.

See Also

selectFeatures

Examples

```
data(pxd010943)
y <- log2(data.matrix(pxd010943))
group <- gsub("_[1234]","", colnames(y))
yimp <- msImpute(y, method="v2-mnar", group=group)
```

`mspip` *Fills missing values by Peptide Identity Propagation (PIP)*

Description

Peptide identity (sequence and charge) is propagated from MS-MS or PASEF identified features in evidence.txt to MS1 features in allPeptides.txt that are detected but not identified. A confidence score (probability) is assigned to every propagation. The confidence scores can be used as observation-level weights in `limma::lmFit` to account for uncertainty in inferred peptide intensity values.

Usage

```
mspip(
  path_txt,
  k = 10,
  thresh = 0,
  skip_weights = TRUE,
  tims_ms = FALSE,
  group_restriction = NULL,
  nlandmarks = 50
)
```

Arguments

path_txt	character. The path to MaxQuant txt directory
k	numeric. The k nearest neighbors to be used for identity propagation. default to 10.
thresh	numeric. The uncertainty threshold for calling a Identity Transfer as confident. Sequence to peptide feature assignments with confidence score (probability) above a threshold (specified by thresh) are considered as confident assignments.The rest of the assignments are discarded and not reported in the output.
skip_weights	logical. If TRUE, the propagation confidence scores are also reported. The confidence scores can be used as observation-level weights in limma linear models to improve differential expression testing. default to FALSE.
tims_ms	logical. Is data acquired by TIMS-MS? default to FALSE.
group_restriction	A data.frame with two columns named Raw.file and group, specifying run file and the (experimental) group to which the run belongs. Use this option for Unbalanced PIP
nlandmarks	numeric. Number of landmark peptides used for measuring neighborhood/coelution similarity. Default to 50.

Details

Data completeness is maximised by Peptide Identity Propagation (PIP) from runs where a peptide is identified by MSMS or PASEF to runs where peptide is not fragmented (hence MS2 information is not available), but is detected at the MS1 level. mspip reports a confidence score for each peptide that was identified by PIP. The intensity values of PIP peptides can be used to reduce missing values, while the reported confidence scores can be used to weight the contribution of these peptide intensity values to variance estimation in linear models fitted in limma.

Author(s)

Soroor Hediyeh-zadeh

See Also

evidenceToMatrix

`plotCV2` *Plot mean-CV^2 trend*

Description

For each peptide, the squares of coefficient of variations are computed and plotted against average log-intensity. Additionally, a loess trend is fitted to the plotted values. Outlier observations (possibly originated from incorrect match between runs), are detected and highlighted. Users can use this plot as a diagnostic plot to determine if filtering by average intensity is required.

Usage

```
plotCV2(y, trend = TRUE, main = NULL, ...)
```

Arguments

`y`	numeric matrix of log-intensity
`trend`	logical. Should a loess trend be fitted to CV^2 and mean values. Default to TRUE.
`main`	character string. Title of the plot. Default to NULL.
`...`	any parameter passed to `plot`.

Details

Outliers are determined by computing the RBF kernels, which reflect the chance that an observed point belong to the dataset (i.e. is close enough in distance to other data points). Users can determine the cut-off for intensity-based filtering with respect to the mean log-intensity of the outlier points.

Value

A plot is created on the current graphics device.

Examples

```
data(pxd010943)
y <- pxd010943
y <- log2(y)
ppCV2 <- plotCV2(y)
```

pxd007959 *Processed peptide intensity matrix and experimental design table from PXD007959 study*

Description

Extracellular vesicles isolated from the descending colon of pediatric patients with inflammatory bowel disease and control patients. Characterizes the proteomic profile of extracellular vesicles isolated from the descending colon of pediatric patients with inflammatory bowel disease and control participants. This object contains data from peptide.txt table output by MaxQuant. Rows are Modified Peptide IDs. Charge state variations are treated as distinct peptide species. Reverse complements and contaminant peptides are discarded. Peptides with more than 4 observed intensity values are retained. Additionally, qualified peptides are required to map uniquely to proteins. Two of the samples with missing group annotation were excluded. The peptide.txt and experimentalDesignTemplate files can be downloaded as RDS object from https://github.com/soroorh/proteomicscasestudies. Code for data processing is provided in package vignette.

Usage

```
pxd007959
```

Format

A list of two: samples (data frame of sample descriptions), and y (numeric matrix of peptide intensity values)

Source

http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD007959

References

Zhang X, Deeke SA, Ning Z, Starr AE, Butcher J, Li J, Mayne J, Cheng K, Liao B, Li L, Singleton R, Mack D, Stintzi A, Figeys D, Metaproteomics reveals associations between microbiome and intestinal extracellular vesicle proteins in pediatric inflammatory bowel disease. Nat Commun, 9(1):2873(2018)

pxd010943 *SWATH-MS Analysis of Gfi1-mutant bone marrow neutrophils*

Description

Contains Peak Area for peptides in PXD010943. This study investigates the proteomic alterations in bone marrow neutrophils isolated from 5-8 week old Gfi1+/-, Gfi1K403R/-, Gfi1R412X/-, and Gfi1R412X/R412X mice using the SWATH-MS technique. This dataset consists of 13 SWATH-DIA runs on a TripleTOF 5600 plus (SCIEX). Rows are peptides. Charge state variations are treated as distinct peptide species. Peptides with more than 4 observed intensity values are retained. The peptide.txt and experimentalDesignTemplate files can be downloaded as RDS object from https://github.com/soroorh/proteomicscasestudies. Code for data processing is provided in package vignette.

Usage

```
pxd010943
```

Format

A matrix

Source

http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD010943

References

Muench DE, Olsson A, Ferchen K, Pham G, Serafin RA, Chutipongtanate S, Dwivedi P, Song B, Hay S, Chetal K, Trump-Durbin LR, Mookerjee-Basu J, Zhang K, Yu JC, Lutzko C, Myers KC, Nazor KL, Greis KD, Kappes DJ, Way SS, Salomonis N, Grimes HL, Mouse models of neutropenia reveal progenitor-stage-specific defects. Nature, 582(7810):109-114(2020)

---

pxd014777 *Processed peptide intensity matrix from PXD014777 study*

Description

A Trapped Ion Mobility Spectrometry (TIMS) dataset of blood plasma from a number of patients acquired in two batches. This is a technical dataset published by MaxQuant to benchmark their software for ion mobility enhanced shotgun proteomics. Rows are Modified Peptide IDs. Charge state variations are treated as distinct peptide species. For peptides with multiple identification types, the intensity is considered to be the median of reported intensity values. Reverse complememts and contaminant peptides are discarded. Peptides with more than 4 observed intensity values are retained. This object contains data from peptide.txt table output by MaxQuant. The evidence.txt file can be downloaded as RDS object from https://github.com/soroorh/proteomicscasestudies. Code for data processing is provided in package vignette.

Usage

```
pxd014777
```

Format

A matrix

Source

http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD014777

References

Prianichnikov N, Koch H, Koch S, Lubeck M, Heilig R, Brehmer S, Fischer R, Cox J, MaxQuant Software for Ion Mobility Enhanced Shotgun Proteomics. Mol Cell Proteomics, 19(6):1058-1069(2020)

`scaleData` *Standardize a matrix to have optionally row means zero and variances one, and/or column means zero and variances one.*

Description

Standardize a matrix to have optionally row means zero and variances one, and/or column means zero and variances one.

Usage

```
scaleData(
  object,
  maxit = 20,
  thresh = 1e-09,
  row.center = TRUE,
  row.scale = TRUE,
  col.center = TRUE,
  col.scale = TRUE,
  trace = FALSE
)
```

Arguments

`object`	numeric matrix giving log-intensity where missing values are denoted by NA. Rows are peptides, columns are samples.
`maxit`	numeric. maximum iteration for the algorithm to converge (default to 20). When both row and column centering/scaling is requested, iteration may be necessary.
`thresh`	numeric. Convergence threshold (default to 1e-09).
`row.center`	logical. if row.center==TRUE (the default), row centering will be performed resulting in a matrix with row means zero. If row.center is a vector, it will be used to center the rows. If row.center=FALSE nothing is done.

row.scale	if row.scale==TRUE, the rows are scaled (after possibly centering, to have variance one. Alternatively, if a positive vector is supplied, it is used for row centering.
col.center	Similar to row.center
col.scale	Similar to row.scale
trace	logical. With trace=TRUE, convergence progress is reported, when iteration is needed.

Details

Standardizes rows and/or columns of a matrix with missing values, according to the biScale algorithm in Hastie et al. 2015. Data is assumed to be normalised and log-transformed. Please note that data scaling might not be appropriate for MS1 data. A good strategy is to compare mean-variance plot (plotCV2) before and after imputation. If the plots look differently, you may need to skip data scaling. The MS1 data are more variable (tend to have higher CV^2), and may contain outliers which will skew the scaling.

Value

A list of two components: E and E.scaled. E contains the input matrix, E.scaled contains the scaled data

References

Hastie, T., Mazumder, R., Lee, J. D., & Zadeh, R. (2015). Matrix completion and low-rank SVD via fast alternating least squares. The Journal of Machine Learning Research, 16(1), 3367-3402.

Hediyeh-zadeh, S., Webb, A. I., & Davis, M. J. (2020). MSImpute: Imputation of label-free mass spectrometry peptides by low-rank approximation. bioRxiv.

See Also

selectFeatures, msImpute

Examples

```
data(pxd010943)
y <- pxd010943
y <- log2(y)
keep <- (rowSums(!is.na(y)) >= 4)
y <- as.matrix.data.frame(y[keep,])
y <- scaleData(y, maxit=30)
```

`selectFeatures` *Select features for MAR/MNAR pattern examination*

Description

Two methods are provided to identify features (peptides or proteins) that can be informative of missing patterns. Method `hvp` fits a linear model to peptide dropout rate (proportion of samples were peptide is missing) against peptide abundance (average log2-intensity). Method `emb` is a information theoretic approach to identify missing patterns. It quantifies the heterogeneity (entropy) of missing patterns per biological (experimental group). This is the default method.

Usage

```
selectFeatures(
  x,
  method = c("ebm", "hvp"),
  group,
  n_features = 500,
  suppress_plot = TRUE
)
```

Arguments

`x`	Numeric matrix giving log-intensity where missing values are denoted by NA. Rows are peptides, columns are samples.
`method`	character. What method should be used to find features? options include `method='hvp'` and `method='ebm'`
`group`	character or factor vector specifying biological (experimental) group e.g. control, treatment, WT, KO
`n_features`	Numeric, number of features with high dropout rate. 500 by default. Applicable if `method="hvp"`.
`suppress_plot`	Logical show plot of dropouts vs abundances. Default to TRUE. Applicable if `method="hvp"`.

Details

In general, the presence of group-wise (structured) blocks of missing values, where peptides are missing in one experimental group can indicate MNAR, whereas if such patterns are absent (or missingness is uniform across the samples), peptides are likely MAR. In the presence of MNAR, left-censored MNAR imputation methods should be chosen. Two methods are provided to explore missing patterns: `method=hvp` identifies top `n_features` peptides with high average expression that also have high dropout rate, defined as the proportion of samples where peptide is missing. Peptides with high (potentially) biological dropouts are marked in the `hvp` column in the output dataframe. This method does not use any information about experimental conditions (i.e. group). Another approach to explore and quantify missing patterns is by looking at how homogeneous or heterogeneous missing patterns are in each experimental group. This is done by computing entropy of

distribution of observed values. This is the default and recommended method for `selectFeatures`. Entropy is reported in `EBM` column of the output. A `NaN` EBM indicates peptide is missing at least in one experimental group. Features set to `TRUE` in `msImpute_feature` column are the features selected by the selected method. Users are encouraged to use the EBM metric to find informative features, hence why the `group` argument is required.

Value

A data frame with a logical column denoting the selected features

Author(s)

Soroor Hediyeh-zadeh

References

Hediyeh-zadeh, S., Webb, A. I., & Davis, M. J. (2020). MSImpute: Imputation of label-free mass spectrometry peptides by low-rank approximation. bioRxiv.

See Also

msImpute

Examples

```
data(pxd007959)
group <- pxd007959$samples$group
y <- data.matrix(pxd007959$y)
y <- log2(y)
hdp <- selectFeatures(y, method="ebm", group = group)
# construct matrix M to capture missing entries
M <- ifelse(is.na(y),1,0)
M <- M[hdp$msImpute_feature,]
# plot a heatmap of missingness patterns for the selected peptides
require(ComplexHeatmap)
hm <- Heatmap(M,
column_title = "dropout pattern, columns ordered by dropout similarity",
              name = "Intensity",
              col = c("#8FBC8F", "#FFEFDB"),
              show_row_names = FALSE,
              show_column_names = TRUE,
              cluster_rows = TRUE,
              cluster_columns = TRUE,
              show_column_dend = TRUE,
              show_row_dend = FALSE,
              row_names_gp =  gpar(fontsize = 7),
              column_names_gp = gpar(fontsize = 8),
              heatmap_legend_param = list(#direction = "horizontal",
              heatmap_legend_side = "bottom",
              labels = c("observed","missing"),
              legend_width = unit(6, "cm")),
         )
```

```
hm <- draw(hm, heatmap_legend_side = "left")
```

Index

∗ **datasets**
pxd007959, 13
pxd010943, 13
pxd014777, 14

computeStructuralMetrics, 2
CPD, 4

evidenceToMatrix, 4

findVariableFeatures, 6

KNC, 7
KNN, 7

msImpute, 8
mspip, 10

plotCV2, 12
pxd007959, 13
pxd010943, 13
pxd014777, 14

scaleData, 15
selectFeatures, 17
softImpute, *9*