Package ‘imputeLCMD’

October 13, 2022

Type Package

Title A Collection of Methods for Left-Censored Missing Data Imputation

Version 2.1

Date 2022-06-09

Maintainer Samuel Wieczorek <samuel.wieczorek@cea.fr>

Description A collection of functions for left-censored missing data imputation. Left-censoring is a special case of missing not at random (MNAR) mechanism that generates non-responses in proteomics experiments. The package also contains functions to artificially generate peptide/protein expression data (log-transformed) as random draws from a multivariate Gaussian distribution as well as a function to generate missing data (both randomly and non-randomly). For comparison reasons, the package also contains several wrapper functions for the imputation of non-responses that are missing at random. * New functionality has been added: a hybrid method that allows the imputation of missing values in a more complex scenario where the missing data are both MAR and MNAR.

License GPL (>= 2)

Depends R (>= 2.10), tmvtnorm, norm, pcaMethods, impute

NeedsCompilation no

Repository CRAN

Date/Publication 2022-06-10 11:50:02 UTC

RoxygenNote 7.2.0

Encoding UTF-8

Author Cosmin Lazar [aut],
Thomas Burger [aut],
Samuel Wieczorek [cre, ctb]

R topics documented:

`generate.ExpressionData` *Generate expression data*

Description

this function generates artificial peptide abundance data with DA proteins samples are drawn from a gaussian distribution

Usage

```
generate.ExpressionData(
  nSamples1,
  nSamples2,
  meanSamples,
  sdSamples,
  nFeatures,
  nFeaturesUp,
  nFeaturesDown,
  meanDynRange,
  sdDynRange,
  meanDiffAbund,
  sdDiffAbund
)
```

Arguments

`nSamples1`	number of samples in condition 1
`nSamples2`	number of samples in condition 2

`meanSamples`	xxx
`sdSamples`	xxx
`nFeatures`	number of total features
`nFeaturesUp`	number of features up regulated
`nFeaturesDown`	number of features down regulated
`meanDynRange`	mean value of the dynamic range
`sdDynRange`	sd of the dynamic range
`meanDiffAbund`	xxx
`sdDiffAbund`	xxx

Value

A list containing the data, the conditions label and the regulation label (up/down/no)

`generate.RollUpMap` *Generate roll up map*

Description

Tthis function generates a map for peptide to protein roll-up

Usage

```
generate.RollUpMap(nProt, pep.Expr.Data)
```

Arguments

`nProt`	number of proteins to map to the peptide expression data
`pep.Expr.Data`	matrix of peptide expression data

Value

the peptide to protein map (for each row in pep.prot.Map the corresponding value corresponds to the index of the protein that peptide is mapped to)

impute.MAR *imputation under MAR/MCAR hypothesis*

Description

This function performs missing values imputation under MAR/MCAR hypothesis. The imputation of MVs is performed for each protein containing MAR/MCAR missing values

Usage

```
impute.MAR(dataSet.mvs, model.selector, method = "MLE")
```

Arguments

`dataSet.mvs`	expression matrix containing abundances with MVs (either peptides or proteins)
`model.selector`	binary vector; "1" indicates MAR/MCAR proteins
`method`	the method to be used for MAR/MCAR missing values. Possible values: MLE (default), SVD, KNN

Value

dataset containing only MNAR (assumed to be left-censored) missing data

impute.MAR.MNAR *Imputation under MCAR and MNAR hypothesis*

Description

this function performs missing values imputation under MCAR and MNAR hypothesis

Usage

```
impute.MAR.MNAR(
  dataSet.mvs,
  model.selector,
  method.MAR = "KNN",
  method.MNAR = "QRILC"
)
```

Arguments

`dataSet.mvs`	expression matrix containing abundances with MVs (either peptides or proteins)
`model.selector`	- binary vector; "1" indicates MCAR proteins
`method.MAR`	- the method to be used for MAR missing values - possible values: MLE (default), SVD, KNN
`method.MNAR`	- the method to be used for MAR missing values

Value

dataset containing complete abundances

`impute.MinDet` *Imputation with min value*

Description

this function performs missing values imputation by the minimum value observed

Usage

```
impute.MinDet(dataSet.mvs, q = 0.01)
```

Arguments

`dataSet.mvs`	expression matrix with MVs (either peptides or proteins)
`q`	the q quantile used to estimate the minimum

Value

dataset containing complete abundances

`impute.MinProb` *Imputation by random draws*

Description

This function performs missing values imputation by random draws from a gaussian

Usage

```
impute.MinProb(dataSet.mvs, q = 0.01, tune.sigma = 1)
```

Arguments

`dataSet.mvs`	expression matrix containing abundances with MVs (either peptides or proteins)
`q`	the q-th quantile used to estimate the minimum value observed for each sample
`tune.sigma`	coefficient that controls the sd of the MNAR distribution

Value

dataset containing complete abundances

impute.QRILC *imputation based on quantile regression*

Description

this function performs missing values imputation based quantile regression

Usage

```
impute.QRILC(dataSet.mvs, tune.sigma = 1)
```

Arguments

dataSet.mvs	expression matrix with MVs (either peptides or proteins)
tune.sigma	coefficient that controls the sd of the MNAR distribution

Value

a list containing: a matrix with the complete abundances, a list with the estimated parameters of the complete data distribution

impute.wrapper.KNN *Imputation with KNN*

Description

This function performs missing values imputation based on KNN algorithm

Usage

```
impute.wrapper.KNN(dataSet.mvs, K)
```

Arguments

dataSet.mvs	expression matrix with MVs (either peptides or proteins)
K	the number of neighbors

Value

dataset containing complete abundances

`impute.wrapper.MLE` *imputation using the EM algorithm*

Description

This function performs missing values imputation using the EM algorithm

Usage

```
impute.wrapper.MLE(dataSet.mvs)
```

Arguments

`dataSet.mvs`	expression matrix with MVs (either peptides or proteins)

Value

expression matrix with MVs imputed

`impute.wrapper.SVD` *imputation based on SVD algorithm*

Description

this function performs missing values imputation based on SVD algorithm

Usage

```
impute.wrapper.SVD(dataSet.mvs, K)
```

Arguments

`dataSet.mvs`	expression matrix with MVs (either peptides or proteins)
`K`	the number of PCs

Value

expression matrix with MVs imputed

`impute.ZERO` *Imputation by 0.*

Description

This function performs missing values imputation by 0.

Usage

```
impute.ZERO(dataSet.mvs)
```

Arguments

`dataSet.mvs` expression matrix containing abundances with MVs (either peptides or proteins)

Value

dataset containing complete abundances

`insertMVs` *Generates missing values in data.*

Description

this function generates missing data in a complete data matrix

Usage

```
insertMVs(original, mean.THR, sd.THR, MNAR.rate)
```

Arguments

`original` complete data matrix containing all measurements

`mean.THR, sd.THR`
- parameters of the threshold distribution which controls the MVs rate (mean.THR should be initially set such that the result of the initial thresholding, in terms of no. of NAs, equals the desired total missing data rate) - example: if one wants to generate 30 mean.THR can be set as follows: mean.THR = quantile(pepExprsData, probs = 0.3) - sd.THR is usually set to a small value (e.g. 0.1)

`MNAR.rate` percentage of MVs which are missing not at random

Value

A list that contains the original complete data matrix, the data matrix with missing data and the percentage of missing data

`intensity_PXD000022` *Dataset PXD000022 from ProteomeXchange.*

Description

This dataset has been collected during a study designed to compare the protein content of the exosome-like vesicles (ELVs) released from C2C12 murine myoblasts during proliferation (ELV-MB), and after differentiation into myotuves (ELV-MT). The dataset within this package contains proteins intensity processed using MaxQuant. More information can be found on ProteomeExchange public repository (http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD000022) or in the original paper (see reference).

Usage

```
data(intensity_PXD000022)
```

Format

A data frame with 660 observations on the following 7 variables.

`Protein.IDs` Peptides/Proteins names

`Intensity.MB.1` a numeric vector

`Intensity.MB.2` a numeric vector

`Intensity.MB.3` a numeric vector

`Intensity.MT.1` a numeric vector

`Intensity.MT.2` a numeric vector

`Intensity.MT.3` a numeric vector

Source

Original MaxQuant data: http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD000022

References

Forterre A, Jalabert A, Berger E, Baudet M, Chikh K, et al. (2014) Proteomic Analysis of C2C12 Myoblast and Myotube Exosome-Like Vesicles: A New Paradigm for Myoblast-Myotube Cross Talk? PLoS ONE 9(1): e84153. doi:10.1371/journal.pone.0084153

intensity_PXD000052 *Dataset PXD000052 from ProteomeXchange.*

Description

This dataset has been collected during a study designed to perform the proteomic analysis of the SLP76 interactome in resting and activated primary mast cells. Four SLP76 replicates (with two analytical replicates each) have been affinity-purified from both resting and activated primary mast cells. The dataset within this package contains proteins intensity processed using MaxQuant. More information can be found on ProteomeExchange public repository (http://proteomecentral.proteomexchange.org/cgi/GetDatas
or in the original paper (see reference).

Usage

```
data(intensity_PXD000052)
```

Format

A data frame with 1991 observations on the following 17 variables.

`Protein.IDs` Peptides/Proteins names

`iBAQ.stSLP_activ1` a numeric vector

`iBAQ.stSLP_activ2` a numeric vector

`iBAQ.stSLP_activ3` a numeric vector

`iBAQ.stSLP_activ4` a numeric vector

`iBAQ.stSLP_rest1` a numeric vector

`iBAQ.stSLP_rest2` a numeric vector

`iBAQ.stSLP_rest3` a numeric vector

`iBAQ.stSLP_rest4` a numeric vector

`iBAQ.WT_activ1` a numeric vector

`iBAQ.WT_activ2` a numeric vector

`iBAQ.WT_activ3` a numeric vector

`iBAQ.WT_activ4` a numeric vector

`iBAQ.WT_rest1` a numeric vector

`iBAQ.WT_rest2` a numeric vector

`iBAQ.WT_rest3` a numeric vector

`iBAQ.WT_rest4` a numeric vector

Source

Original MaxQuant data: http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD000052

References

Bounab Y, Hesse AM, Iannascoli B, Grieco L, Coute Y, Niarakis A, Roncagalli R, Lie E, Lam KP, Demangel C, Thieffry D, Garin J, Malissen B, Da?ron M, Proteomic analysis of the SH2 domain-containing leukocyte protein of 76 kDa (SLP76) interactome in resting and activated primary mast cells [corrected]. Mol Cell Proteomics, 12(10):2874-89(2013).

`intensity_PXD000438` *Dataset PXD000438 from ProteomeXchange.*

Description

This dataset has been collected during a study designed to compare human primary tumor-derived xenograph proteomes of the two major histological non-small cel lung cancer subtypes: adenocarcinoma (ADC) and squamous cell carcinoma (SCC). The dataset within this package contains proteins intensity for 6 ADC and 6 SCC samples, processed using MaxQuant. More information can be found on ProteomeExchange public repository(http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD000438 or in the original paper (see reference).

Usage

```
data(intensity_PXD000438)
```

Format

A data frame with 3709 observations on the following 13 variables.

`Protein.IDs` Peptides/Proteins names

`Intensity.092.1` a numeric vector

`Intensity.092.2` a numeric vector

`Intensity.092.3` a numeric vector

`Intensity.441.1` a numeric vector

`Intensity.441.2` a numeric vector

`Intensity.441.3` a numeric vector

`Intensity.561.1` a numeric vector

`Intensity.561.2` a numeric vector

`Intensity.561.3` a numeric vector

`Intensity.691.1` a numeric vector

`Intensity.691.2` a numeric vector

`Intensity.691.3` a numeric vector

Source

Original MaxQuant data: http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD000438

References

Zhang W, Wei Y, Ignatchenko V, Li L, Sakashita S, Pham NA, Taylor P, Tsao MS, Kislinger T, Moran MF, Proteomic profiles of human lung adeno and squamous cell carcinoma using super-SILAC and label-free quantification approaches. Proteomics, 14(6):795-803(2014).

Examples

```
data(intensity_PXD000438)
```

`intensity_PXD000501` *Dataset PXD000501 from ProteomeXchange.*

Description

This dataset contains three biological replicates with three technical replicates each for the conditiones media (CM) and the whole cell lysates (WCL) of C8-D1A cell lines. The dataset within this package contains proteins iBAQ intensity processed using MaxQuant. More information can be found on ProteomeExchange public repository (http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD00050 or in the original paper (see reference).

Usage

```
data(intensity_PXD000501)
```

Format

A data frame with 7363 observations on the following 19 variables.

`Protein.IDs` Peptides/Proteins names

`iBAQ.secretome_set1_tech1` a numeric vector

`iBAQ.secretome_set1_tech2` a numeric vector

`iBAQ.secretome_set1_tech3` a numeric vector

`iBAQ.secretome_set2_tech1` a numeric vector

`iBAQ.secretome_set2_tech2` a numeric vector

`iBAQ.secretome_set2_tech3` a numeric vector

`iBAQ.secretome_set3_tech1` a numeric vector

`iBAQ.secretome_set3_tech2` a numeric vector

`iBAQ.secretome_set3_tech3` a numeric vector

`iBAQ.whole_set1_tech1` a numeric vector

`iBAQ.whole_set1_tech2` a numeric vector

`iBAQ.whole_set1_tech3` a numeric vector

`iBAQ.whole_set2_tech1` a numeric vector

`iBAQ.whole_set2_tech2` a numeric vector

`iBAQ.whole_set2_tech3`	a numeric vector
`iBAQ.whole_set3_tech1`	a numeric vector
`iBAQ.whole_set3_tech2`	a numeric vector
`iBAQ.whole_set3_tech3`	a numeric vector

Source

Original MaxQuant data: http://proteomecentral.proteomexchange.org/cgi/GetDataset?ID=PXD000501

References

Han D, Jin J, Woo J, Min H, Kim Y, Proteomic analysis of mouse astrocytes and their secretome by a combination of FASP and StageTip-based, high pH, reversed-phase fractionation. Proteomics, ():(2014).

Examples

```
data(intensity_PXD000501)
```

model.Selector — *Identifies row in the data matrix affected by a MNAR missingness mechanism*

Description

- this function determines row in the data matrix affected by a MNAR missingness mechanism - it is based on the assumption that the distributions of the mean values of proteins follows a normal distribution - the method makes use of a decision function defined as a tradeoff between the empirical CDF of the proteins' means and the theoretical CDF assuming that no MVs are present

Usage

```
model.Selector(dataSet.mvs)
```

Arguments

`dataSet.mvs`	expression matrix containing abundances with MVs (either peptides or proteins)

Value

flags vector; "1" denotes rows containing random missing values; "0" denotes rows containing left-censored missing values pep2prot *peptide to protein roll-up*

Description

this function performs peptide to protein roll-up

Usage

```
pep2prot(pep.Expr.Data, rollup.map)
```

Arguments

`pep.Expr.Data`	matrix of peptide expression data
`rollup.map`	the map to peptide to protein mapping

Value

matrix of peptide expression data

Index