Package: GMSimpute (via r-universe)

August 28, 2024

Title Generalized Mass Spectrum Missing Peaks Abundance Imputation

Version 0.0.1.0

Description GMSimpute implements the Two-Step Lasso (TS-Lasso) and compound minimum to recover the abundance of missing peaks in mass spectrum analysis. TS-Lasso is a label-free imputation method that handles various types of missing peaks simultaneously. This package provides the procedure to generate missing peaks (or data) for simulation study, as well as a tool to estimate and visualize the proportion of missing at random.

Depends R (>= 3.5.0)

License GPL(>=2)

Encoding UTF-8

LazyData true

RoxygenNote 6.1.0

Imports utils, glmnet, ggplot2, reshape2

Repository https://qianli10000.r-universe.dev

RemoteUrl https://github.com/qianli10000/gmsimpute

RemoteRef HEAD

RemoteSha c0f55aa72c79ac90ad78d31041d7d355c2c638d9

Contents

GMS.Lasso *Generalized Mass Spectrum missing peaks imputation with Two-Step Lasso as default algorithm*

Description

GMS.Lasso recovers the abundance of missing peaks via either TS.Lasso or the minimum abundance per compound.

Usage

```
GMS.Lasso(input_data, alpha = 1, nfolds = 10, log.scale = TRUE,
  TS.Lasso = TRUE)
```

Arguments

input_data	Raw abundance matrix with missing value, with features in rows and samples in columns.
alpha	Weights for L1 penalty in Elastic Net. The default and suggested value is alpha=1, which is for Lasso.
nfolds	The number of folds used in parameter (lambda) tuning.
log.scale	Whether the input_data needs log scale transform.The default is log.scale=T, assuming input_data is the raw abundance matrix. If input_data is log abundance matrix, log.scale=F.
TS.Lasso	Whether to use TS.Lasso or the minimum per compound for imputation.

Value

imputed.final	The imputed abundance matrix at the scale of input_data.

Examples

```
data('tcga.bc')
# tcga.bc contains mass specturm abundance of 150 metabolites for 30 breast cancer
# tumor and normal tissue samples with missing values.

imputed.compound.min=GMS.Lasso(tcga.bc,log.scale=TRUE,TS.Lasso=FALSE)
# Impute raw abundance matrix tcga.bc with compound minimum

imputed.tslasso=GMS.Lasso(tcga.bc,log.scale=TRUE,TS.Lasso=TRUE)
# Impute raw abundance matrix tcga.bc with TS.Lasso
```

GTS.Lasso *Generalized Two-Step Lasso for missing peaks imputation*

Description

GTS.Lasso recovers the abundance of missing peaks via either TS.Lasso or the minimum abundance per compound.

Usage

```
GTS.Lasso(input_data, alpha = 1, nfolds = 10, log.scale = TRUE,
  TS.Lasso = TRUE)
```

Arguments

input_data	Raw abundance matrix with missing value, with features in rows and samples in columns.
alpha	Weights for L1 penalty in Elastic Net. The default and suggested value is alpha=1, which is for Lasso.
nfolds	The number of folds used in parameter (lambda) tuning.
log.scale	Whether the input_data needs log scale transform.The default is log.scale=T, assuming input_data is the raw abundance matrix. If input_data is log abundance matrix, log.scale=F.
TS.Lasso	Whether to use TS.Lasso or the minimum per compound for imputation.

Value

imputed.final	The imputed abundance matrix at the scale of input_data.

Examples

```
data('tcga.bc')
# tcga.bc contains mass specturm abundance of 150 metabolites for 30 breast cancer
# tumor and normal tissue samples with missing values.

imputed.compound.min=GTS.Lasso(tcga.bc,log.scale=TRUE,TS.Lasso=FALSE)
# Impute raw abundance matrix tcga.bc with compound minimum

imputed.tslasso=GTS.Lasso(tcga.bc,log.scale=TRUE,TS.Lasso=TRUE)
# Impute raw abundance matrix tcga.bc with TS.Lasso
```

MAR.est *Missing At Random (MAR) proportion estimation based on technical replicates.*

Description

MAR.est estimates the proportion of missing peaks at random (MAR) caused by preprocessing tools with two technical replicates per sample.

Usage

```
MAR.est(abundance, sample, log.scale = TRUE, violin.plot = FALSE)
```

Arguments

abundance	The full abundance matrix without missing value, with features in rows and samples in columns.
sample	A vector of characters or integers. It is the sample name for each pair of replicates.
log.scale	A scalar or vector of proportions. It is the total percentage of missing peaks throughout the full matrix.
violin.plot	Logical, whether to generate violin and box plots to visualize abundance distribution of missing and nonmissing peaks.

Value

MAR.Proportion	Estimated MAR proportion
plot	Violin and box plots generated by ggplot2

Examples

```
data('replicates')
# replicates contains mass specturm log abundance of 85 peptides
# with missing values for 4 pairs of technical replicates.


MAR=MAR.est(replicates,sample=rep(1:4,each=2),log.scale=FALSE,violin.plot=TRUE)
# Estimates the MAR proportion in the 4 pairs of replicates and output violin/box plots object.

print(MAR$plot)
# Print violin/box plots
```

missing.sim *Missing peaks generating procedure for simulation study*

Description

missing.sim generates various types of missing peaks based on specified missing proportion.

Usage

```
missing.sim(complete.data, total.missing, random, pct.full,
  seednum = 365)
```

Arguments

complete.data	The full abundance matrix without missing value, with features in rows and samples in columns.
total.missing	A scalar or vector of proportions. It is the total percentage of missing peaks throughout the full matrix.
random	A scalar or vector of proportions. It is the percentage of random missing in all the missing peaks.
pct.full	A scalar for the percentage of alighned features (metabolites or peptides) without missing peaks.
seednum	The seed set for generating missing peaks index. Default seed is seednum=365.

Value

simulated.data	The list of all simulated scenarios
Labels	The description for each simulated scenario

Examples

```
data('tcga.bc.full')
# tcga.bc.full contains mass specturm abundance of 100 metabolites for 30 breast cancer
# tumor and normal tissue samples without missing values.


simulated.data=missing.sim(tcga.bc.full,total.missing=c(0.2,0.4),random=c(0.3,0.5,0.7),pct.full=0.4)
# Generate missing (NA) values in full abundance matrix tcga.bc.full permuting all scenarios
```

`replicates` *Raw mass spectrum proteomics log abundance for 4 pairs of technical replicates.*

Description

Raw mass spectrum proteomics log abundance for 4 pairs of technical replicates.

Usage

```
replicates
```

Format

A data frame of 85 rows and 8 columns with missing peaks' abundance as NA.

`tcga.bc` *Raw mass spectrum metabolomics data for TCGA breast cancer study.*

Description

Raw mass spectrum metabolomics data for TCGA breast cancer study.

Usage

```
tcga.bc
```

Format

A data frame of 150 rows and 30 columns with missing peaks' abundance as NA.

`tcga.bc.full` *A subset of mass spectrum metabolomics data for TCGA breast cancer study without missing peaks.*

Description

A subset of mass spectrum metabolomics data for TCGA breast cancer study without missing peaks.

Usage

```
tcga.bc.full
```

Format

A data frame of 100 rows and 30 columns without missing value (NA).

TS.Lasso *Two-Step Lasso for missing peaks imputation*

Description

TS.Lasso recovers the abundance of various types of missing peaks.

Usage

```
TS.Lasso(input_data, alpha = 1, nfolds = 10, log.scale = TRUE)
```

Arguments

input_data	Raw abundance matrix with missing value, with features in rows and samples in columns.
alpha	Weights for L1 penalty in Elastic Net. The default and suggested value is alpha=1, which is for Lasso.
nfolds	The number of folds used in parameter (lambda) tuning.
log.scale	Whether the input_data needs log scale transform.The default is log.scale=T, assuming input_data is the raw abundance matrix. If input_data is log abundance matrix, set log.scale=F.

Value

imputed.final	The imputed abundance matrix at the scale of input_data.

Examples

```
data('tcga.bc')
# tcga.bc contains mass specturm abundance of 150 metabolites for 30 breast cancer
# tumor and normal tissue samples with missing values.

imputed=TS.Lasso(tcga.bc,log.scale=TRUE)
# Impute raw abundance matrix tcga.bc
```

Index