Package 'MSstatsPTM'

February 27, 2021

Type Package

Title Statistical Characterization of Post-translational Modifications

Version 1.0.0

Date 2020-09-28

Description MSstatsPTM provides general statistical methods for quantitative characterization of post-translational modifications (PTMs). Typically, the analysis involves the quantification of PTM sites (i.e., modified residues) and their corresponding proteins, as well as the integration of the quantification results. MSstatsPTM provides functions for summarization, estimation of PTM site abundance, and detection of changes in PTMs across experimental conditions.

License Artistic-2.0

Depends R (>= 4.0)

Imports broom, dplyr, rlang, stats, tibble, tidyr, tidyselect, Biostrings

Suggests knitr, rmarkdown, testthat (>= 2.1.0), BiocStyle

VignetteBuilder knitr

biocViews MassSpectrometry, Proteomics, Software, DifferentialExpression

BugReports https://github.com/tsunghengtsai/MSstatsPTM

Encoding UTF-8

LazyData true

ByteCompile true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

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

git_branch RELEASE_3_12

git_last_commit d8c642c

git_last_commit_date 2020-10-27

Date/Publication 2021-02-26

Author Tsung-Heng Tsai [aut, cre], Olga Vitek [aut]

Maintainer Tsung-Heng Tsai <tsai.tsungheng@gmail.com>

R topics documented:

adjustProteinLevel . . . 2
annotSite . . . 3
estimateAbundance . . . 4
extractMeanDiff . . . 5
fitLinearModel . . . 6
fixedGroup . . . 6
fixedGroupBatch . . . 7
locateMod . . . 8
MSstatsPTM . . . 8
PTMcompareMeans . . . 9
PTMestimate . . . 10
PTMlocate . . . 11
PTMnormalize . . . 11
PTMsimulateExperiment . . . 12
PTMsummarize . . . 13
simulatePeaks . . . 14
simulateSites . . . 15
simulateSummarization . . . 15
summarizeFeatures . . . 16
tidyEstimates . . . 17
tidyFasta . . . 17

Index **18**

`adjustProteinLevel` *Adjust differential analysis result with respect to protein abundance*

Description

`adjustProteinLevel` performs the adjustment with respect to protein abundance.

Usage

```
adjustProteinLevel(diffSite, diffProtein)
```

Arguments

`diffSite`	A data frame for the differential analysis result of PTMs, returned by the function `extractMeanDiff` with the option `perProtein=FALSE`. The data frame contains columns of `Protein`, `Site`, `Label`, `log2FC`, `SE`, `Tvalue`, `DF`, and `pvalue`.
`diffProtein`	A data frame for the differential analysis result of proteins, returned by the function `extractMeanDiff` with the option `perProtein=TRUE`. The data frame contains columns of `Protein`, `Label`, `log2FC`, `SE`, `Tvalue`, `DF`, and `pvalue`.

Value

A data frame.

Examples

```
sim <- PTMsimulateExperiment(
    nGroup=2, nRep=2, nProtein=1, nSite=1, nFeature=5,
    logAbundance=list(
        PTM=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05),
        PROTEIN=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05)
    )
)
summarized <- PTMsummarize(sim)
estimates <- PTMestimate(summarized)
res <- extractMeanDiff(estimates[["PTM"]], "G_1", "G_2", FALSE)
res_prot <- extractMeanDiff(estimates[["PROTEIN"]], "G_1", "G_2", TRUE)
adjustProteinLevel(res, res_prot)
```

---

annotSite *Annotate modification site*

Description

annotSite annotates modified sites as their residues and locations.

Usage

```
annotSite(aaIndex, residue, lenIndex = NULL)
```

Arguments

aaIndex	An integer vector. Location of the sites.
residue	A string vector. Amino acid residue.
lenIndex	An integer. Default is NULL

Value

A string.

Examples

```
annotSite(10, "K")
annotSite(10, "K", 3L)
```

`estimateAbundance` *Estimate log2-abundances of PTM sites or proteins*

Description

`estimateAbundance` takes as input the summarized log2-intensities for each PTM site, performs statistical modeling for the abundance of the site, and returns the estimates of model parameters for all sites in all experimental conditions.

Usage

```
estimateAbundance(df, fctBatch = FALSE, perProtein = FALSE)
```

Arguments

`df`	A data frame with columns of `protein`, `site`, `group`, `run`, `log2inty`, and possibly, `batch`.
`fctBatch`	A logical. `TRUE` considers a fixed batch effect, `FALSE` otherwise. Default is `FALSE`.
`perProtein`	A logical. `TRUE` ignores the site-level information for PTM and considers protein as a whole, `FALSE` otherwise. Default is `FALSE`.

Value

A list of two elements named `PTM` and `PROTEIN`. The `PTM` list has four elements: `protein` (a string vector of protein names), `site` (a string vector of PTM sites), `param` (a list of model parameter estimates for each site), and `df` (a numeric vector of degrees of freedom for each model). The `PROTEIN` list includes all as in `PTM`, except `site`.

Examples

```
sim <- PTMsimulateExperiment(
    nGroup=2, nRep=2, nProtein=1, nSite=1, nFeature=5,
    logAbundance=list(
        PTM=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05),
        PROTEIN=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05)
    )
)
s <- PTMsummarize(sim)
estimateAbundance(s[["PTM"]])
estimateAbundance(s[["PROTEIN"]], perProtein=TRUE)
```

`extractMeanDiff` *Compare mean abundances for PTM sites (or proteins) across conditions*

Description

`extractMeanDiff` performs significance analysis for detection of changes in PTM mean abundances between conditions.

Usage

```
extractMeanDiff(data, controls, cases, perProtein = FALSE)
```

Arguments

`data`	A list of abundance estimates with the following elements: `protein`, `site`, `param`, and `df`. `site` may be excluded when performing per-protein analysis (`perProtein` is `TRUE`).
`controls`	A string vector of control groups in the comparisons.
`cases`	A string vector of case groups.
`perProtein`	A logical. `TRUE` ignores the site-level information for PTM and considers protein as a whole, `FALSE` performs site-level analysis. Default is `FALSE`.

Value

A data frame.

Examples

```
sim <- PTMsimulateExperiment(
    nGroup=2, nRep=2, nProtein=1, nSite=1, nFeature=5,
    logAbundance=list(
        PTM=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05),
        PROTEIN=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05)
    )
)
summarized <- PTMsummarize(sim)
estimates <- PTMestimate(summarized)
extractMeanDiff(estimates[["PTM"]], controls="G_1", cases="G_2", FALSE)
extractMeanDiff(estimates[["PROTEIN"]], controls="G_1", cases="G_2", TRUE)
```

fitLinearModel *Fit linear model*

Description

fitLinearModel fits and returns a linear model with log2inty as response, and group and possibly batch as fixed effects.

Usage

```
fitLinearModel(df, fctBatch = FALSE)
```

Arguments

df	A data frame with columns log2inty, group, and batch for one PTM site.
fctBatch	A logical. TRUE considers batch effect, FALSE otherwise. Default is FALSE.

Value

An lm model object.

Examples

```
x1 <- data.frame(
    batch=rep(c("1", "2"), each=4),
    group=rep(c("1", "2"), 4),
    log2inty=rep(c(10, 12), 4) + rnorm(8)
)
fitLinearModel(x1, fctBatch=TRUE)

x2 <- data.frame(
    group=rep(c("1", "2"), 3),
    log2inty=rep(c(10, 12), 3) + rnorm(6)
)
fitLinearModel(x2)
```

fixedGroup *Linear model with group effect*

Description

fixedGroup fits and returns a linear model with group as a fixed effect.

Usage

```
fixedGroup(df)
```

Arguments

df	A data frame with columns log2inty and group for one PTM site.

Value

An `lm` model object.

Examples

```
x <- data.frame(
    group=rep(c("1", "2"), 3),
    log2inty=rep(c(10, 12), 3) + rnorm(6)
)
fixedGroup(x)
```

`fixedGroupBatch` *Linear model with group and batch effects*

Description

`fixedGroupBatch` fits and returns a linear model with `log2inty` as response, and `group` and `batch` as fixed effects.

Usage

```
fixedGroupBatch(df)
```

Arguments

`df`	A data frame with columns `log2inty`, `group`, and `batch` for one PTM site.

Value

An `lm` model object.

Examples

```
x <- data.frame(
    batch=rep(c("1", "2"), each=4),
    group=rep(c("1", "2"), 4),
    log2inty=rep(c(10, 12), 4) + rnorm(8)
)
fixedGroupBatch(x)
```

locateMod *Locate modified sites with a peptide*

Description

locateMod locates modified sites with a peptide.

Usage

```
locateMod(peptide, aaStart, residueSymbol)
```

Arguments

peptide	A string. Peptide sequence.
aaStart	An integer. Starting index of the peptide.
residueSymbol	A string. Modification residue and denoted symbol.

Value

A string.

Examples

```
locateMod("P*EP*TIDE", 3, "\\*")
```

MSstatsPTM *MSstatsPTM: A package for statistical characterization of PTMs*

Description

The MSstatsPTM package provides four main functions for quantitative analysis of PTMs

Details

Quantitative analyses of PTMs are supported by four main functions of *MSstatsPTM*:

Normalization

PTMnormalize() normalizes the quantified peak intensities to correct systematic variation across MS runs.

Summarization

PTMsummarize() summarizes log2-intensities of spectral features (i.e., precursor ions in DDA, fragments in DIA, or transitions in SRM) into one value per PTM site per run or one value per protein per run.

Estimation

PTMestimate() takes as input the summarized log2-intensities for each PTM site, performs statistical modeling for the log2-abundance of the site, and returns the estimates of model parameters for all PTM sites in all experimental conditions.

Comparison

PTMcompareMeans() performs statistical testing for detecting changes in PTM mean abundances between conditions.

PTMcompareMeans *Compare mean abundances for all PTM sites across conditions*

Description

PTMcompareMeans performs significance analysis for detection of changes in PTM mean abundances between conditions.

Usage

```
PTMcompareMeans(data, controls, cases, adjProtein = FALSE)
```

Arguments

data	A list of two elements named PTM and PROTEIN. The PTM list has four elements: protein (a string vector of protein names), site (a string vector of PTM sites), param (a list of model parameter estimates for each site), and df (a numeric vector of degrees of freedom for each model). The PROTEIN list includes all as in PTM, except the element site.
controls	A string vector of control groups in the comparisons.
cases	A string vector of case groups.
adjProtein	A logical. TRUE performs protein-level adjustment, FALSE otherwise. Default is FALSE.

Value

A data frame.

Examples

```
sim <- PTMsimulateExperiment(
    nGroup=2, nRep=2, nProtein=1, nSite=1, nFeature=5,
    logAbundance=list(
        PTM=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05),
        PROTEIN=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05)
    )
)
summarized <- PTMsummarize(sim)
estimates <- PTMestimate(summarized)
PTMcompareMeans(estimates, controls="G_1", cases="G_2")
```

PTMestimate *Estimate log2-abundances of PTM sites and proteins*

Description

PTMestimate takes as input the summarized log2-intensities for each PTM site, performs statistical modeling for the abundance of the site, and returns the estimates of model parameters for all sites in all experimental conditions. If protein log2-intensities are availble, the same estimation procedure is applied to each protein as well.

Usage

```
PTMestimate(data, fctBatch = FALSE)
```

Arguments

data	A list of two data frames named PTM and PROTEIN. The PTM data frame includes columns of protein, site, group, run, log2inty, and possibly, batch. The PROTEIN data frame includes all columns as in PTM except site.
fctBatch	A logical defining the handling of batch effect for all data or two logicals for the PTM and PROTEIN (if provided) data separately. TRUE considers a fixed batch effect, FALSE otherwise. Default is FALSE.

Value

A list of two lists named PTM and PROTEIN. The PTM list has four elements: protein (a string vector of protein names), site (a string vector of PTM sites), param (a list of model parameter estimates for each site), and df (a numeric vector of degrees of freedom for each model). The PROTEIN list includes all as in PTM, except site.

Examples

```
sim <- PTMsimulateExperiment(
    nGroup=2, nRep=2, nProtein=1, nSite=1, nFeature=5,
    logAbundance=list(
        PTM=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05),
        PROTEIN=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05)
    )
)
s <- PTMsummarize(sim)
PTMestimate(s)
```

PTMlocate *Annotate modified sites with associated peptides*

Description

PTMlocate annotates modified sites with associated peptides.

Usage

```
PTMlocate(peptide, uniprot, fasta, modResidue, modSymbol, rmConfound = FALSE)
```

Arguments

peptide	A string vector of peptide sequences. The peptide sequence does not include its preceding and following AAs.
uniprot	A string vector of Uniprot identifiers of the peptides' originating proteins. UniProtKB entry isoform sequence is used.
fasta	A tibble with FASTA information. Output of tidyFasta.
modResidue	A string. Modifiable amino acid residues.
modSymbol	A string. Symbol of a modified site.
rmConfound	A logical. TRUE removes confounded unmodified sites, FALSE otherwise. Default is FALSE.

Value

A data frame with three columns: uniprot_iso, peptide, site.

Examples

```
fasta <- tidyFasta("https://www.uniprot.org/uniprot/O13297.fasta")
PTMlocate("DRVSYIHNDSC*TR", "O13297", fasta, "C", "\\*")
```

PTMnormalize *Normalization of log2-intensities across MS runs*

Description

PTMnormalize normalizes log2-intensities of spectral features across MS runs using a reference, or by equalizing a chosen summary (the log2 intensity summation, median, or mean of log2-intensities) from all features, features of modified peptides or features of unmodified peptides.

Usage

```
PTMnormalize(data, method = "median", refs)
```

Arguments

data	A list of two data frames named PTM and PROTEIN. Both the PTM data frame and the PROTEIN data frame include columns of run, feature, and log2inty.
method	A string defining the normalization method. Default is "median", which equalizes the medians of log2-intensities across MS runs. Other methods include to equalize log2 of intensity summation ("logsum"), to equalize the means of log2-intensities ("mean"), and to adjust the log2-intensities based on a reference ("ref") given by (refs).
refs	A list of two data frames named PTM and PROTEIN. Each defines the adjustment of log2-intensities for the MS runs in its corresponding data.

Value

Normalized data stored as in data.

Examples

```
sim <- PTMsimulateExperiment(
    nGroup=2, nRep=2, nProtein=1, nSite=1, nFeature=5,
    logAbundance=list(
        PTM=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05),
        PROTEIN=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05)
    )
)
PTMnormalize(sim)
```

PTMsimulateExperiment *Simulate PTM quantification experiments*

Description

PTMsimulateExperiment simulates a PTM quantification experiment with a list of log2-intensities of multiple spectral features, PTM sites and the corresponding proteins, in multiple MS runs and conditions.

Usage

```
PTMsimulateExperiment(nGroup, nRep, nProtein, nSite, nFeature, logAbundance)
```

Arguments

nGroup	An integer to specify the number of conditions.
nRep	An integer to specify the number of replicates per condition.
nProtein	An integer to specify the number of protein.
nSite	An integer to specify the number of PTM sites per protein.
nFeature	An integer to specify the number of features per site.

logAbundance A list of two lists named PTM and PROTEIN. Each contains four elements: mu (a numeric representing the overall mean log2-abundance), delta (a numeric vector for the deviation of the mean log2-abundance for each group from the overall mean), sRep (a numeric representing the standard deviation for run-to-run variation), and sPeak (a numeric representing the standard deviation in peak log2-intensities).

Value

A tibble with columns of protein, site, group, run, feature, log2inty.

Examples

```
PTMsimulateExperiment(
    nGroup=2, nRep=2, nProtein=1, nSite=1, nFeature=5,
    logAbundance=list(
        PTM=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05),
        PROTEIN=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05)
    )
)
```

PTMsummarize *Site-level summarization*

Description

PTMsummarize summarizes the peak log2-intensities for each PTM site into one value per run. If protein peak-intensities are availble, the same summarization procedure is applied to each protein as well.

Usage

```
PTMsummarize(data, method = "tmp")
```

Arguments

data A list of two data frames named PTM and PROTEIN. The PTM data frame includes columns of protein, site, group, run, feature, log2inty, and possibly, batch. The PROTEIN data frame includes all columns as in PTM except site.

method A string defining the summarization method. Default is "tmp", which applies Tukey's median polish. Other methods include log2 of the summation of peak intensities ("logsum"), and mean ("mean"), median ("median") and max ("max") of the log2-intensities.

Value

A list of two data frames named PTM and PROTEIN. The PTM data frame has columns of protein, site, group, run, log2inty, and possibly, batch. The PROTEIN data frame includes all as in PTM, except site.

Examples

```
sim <- PTMsimulateExperiment(
    nGroup=2, nRep=2, nProtein=1, nSite=1, nFeature=5,
    logAbundance=list(
        PTM=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05),
        PROTEIN=list(mu=25, delta=c(0, 1), sRep=0.2, sPeak=0.05)
    )
)
PTMsummarize(sim)
```

`simulatePeaks` *Simulate peak log2-intensities*

Description

`simulateSites` simulates a list of log2-intensities of multiple spectral features of a PTM site, in multiple MS runs and conditions.

Usage

```
simulatePeaks(nGroup, nRep, nFeature, mu, delta, sRep, sPeak)
```

Arguments

`nGroup`	An integer to specify the number of conditions.
`nRep`	An integer to specify the number of replicates per condition.
`nFeature`	An integer to specify the number of features per site.
`mu`	A numeric to specify the overall mean log2-intensity.
`delta`	A numeric to specify the deviation of the mean log2-abundance of each group from the overall mean.
`sRep`	A numeric to specify the standard deviation for run-to-run variation.
`sPeak`	A numeric to specify the standard deviation in peak log2-intensities.

Value

A tibble with columns of `group`, `run`, `feature`, and `log2inty`.

Examples

```
simulatePeaks(nGroup=2, nRep=3, nFeature=5, 25, c(0, 1), 0.2, 0.05)
```

simulateSites *Simulate peak log-intensities for PTM sites*

Description

simulateSites simulates a list of log2-intensities of multiple spectral features and PTM sites of one protein, in multiple MS runs and conditions.

Usage

```
simulateSites(nGroup, nRep, nSite, nFeature, mu, delta, sRep, sPeak)
```

Arguments

nGroup	An integer to specify the number of conditions.
nRep	An integer to specify the number of replicates per condition.
nSite	An integer to specify the number of PTM sites per protein.
nFeature	An integer to specify the number of features per site.
mu	A numeric to specify the overall mean log2-intensity.
delta	A numeric to specify the deviation of the mean log2-abundance of each group from the overall mean.
sRep	A numeric to specify the standard deviation for run-to-run variation.
sPeak	A numeric to specify the standard deviation in peak log2-intensities.

Value

A tibble with columns of site, group, run, feature, log2inty.

Examples

```
simulateSites(nGroup=2, nRep=2, nSite=2, nFeature=5, 25, c(0, 1), 0.2, 0.05)
```

simulateSummarization *Simulate site-level summarization for PTM experiment*

Description

simulateSummarization simulates the summarized log2-intensity value of a PTM site in each MS run. The value is randomly generated based on a normal distribtuion, where the average log2-intensity is defined for each condition

Usage

```
simulateSummarization(nGroup, nRep, mu, delta, sRep)
```

Arguments

nGroup	An integer to specify the number of conditions.
nRep	An integer to specify the number of replicates per condition.
mu	A numeric value of the overall mean log2-abundance.
delta	A numeric vector to specify the deviation of the mean log2-abundance of each group from the overall mean.
sRep	A numeric. Standard deviation of the log2-intensities.

Value

A tibble with columns of group, run and log2inty.

Examples

```
simulateSummarization(nGroup=2, nRep=3, 25, c(0, 1), 0.2)
```

---

summarizeFeatures *Summarization for one site*

Description

summarizeFeatures summarizes feature log2-intensities for a PTM site and returns one summarized value per run. Tukey's median polish is used by default.

Usage

```
summarizeFeatures(df, method = "tmp")
```

Arguments

df	A data frame with columns of run, feature, and log2inty.
method	A string defining the summarization method. Default is "tmp", which applies Tukey's median polish. Other methods include log2 of the sum of intensity ("logsum"), and mean ("mean"), median ("median") and max ("max") of the log2-intensities.

Value

A tibble restoring one summarized value per MS run.

Examples

```
df <- data.frame(
    run=c("a", "a", "a", "b", "b"),
    feature=c("F1", "F2", "F3", "F1", "F3"),
    log2inty=rnorm(5)
)
summarizeFeatures(df, method="tmp")
```

tidyEstimates *Extract estimate of group effect*

Description

tidyEstimates extracts the estimate of group effect from a fitted linear model.

Usage

```
tidyEstimates(fit, data)
```

Arguments

fit	An lm model object.
data	A data frame used to derive the model object fit.

Value

A data frame restoring the estimated model parameters.

Examples

```
x <- data.frame(
    group=rep(c("1", "2"), 3),
    log2inty=rep(c(10, 12), 3) + rnorm(6)
)
fit <- fitLinearModel(x)
tidyEstimates(fit, x)
```

tidyFasta *Read and tidy a FASTA file*

Description

tidyFasta reads and tidys FASTA file.

Usage

```
tidyFasta(path)
```

Arguments

path	A string of path to a FASTA file.

Value

A tibble with columns named header, sequence, uniprot_ac, uniprot_iso, entry_name.

Examples

```
tidyFasta("https://www.uniprot.org/uniprot/O13297.fasta")
```

Index

adjustProteinLevel, 2
annotSite, 3

estimateAbundance, 4
extractMeanDiff, 5

fitLinearModel, 6
fixedGroup, 6
fixedGroupBatch, 7

locateMod, 8

MSstatsPTM, 8

PTMcompareMeans, 9
PTMestimate, 10
PTMlocate, 11
PTMnormalize, 11
PTMsimulateExperiment, 12
PTMsummarize, 13

simulatePeaks, 14
simulateSites, 15
simulateSummarization, 15
summarizeFeatures, 16

tidyEstimates, 17
tidyFasta, 17