Package 'MSstatsShiny'

January 27, 2023

Type Package

Title MSstats GUI for Statistical Anaylsis of Proteomics Experiments

Version 1.0.7

Date 2022-12-04

Description MSstatsShiny is an R-Shiny graphical user interface (GUI) integrated with the R packages MSstats, MSstatsTMT, and MSstatsPTM. It provides a point and click end-to-end analysis pipeline applicable to a wide variety of experimental designs. These include data-dependedent acquisitions (DDA) which are label-free or tandem mass tag (TMT)-based, as well as DIA, SRM, and PRM acquisitions and those targeting post-translational modifications (PTMs). The application automatically saves users selections and builds an R script that recreates their analysis, supporting reproducible data analysis.

License Artistic-2.0

Depends R (>= 4.2)

Imports shiny, shinyBS, shinyjs, shinybusy, dplyr, ggplot2, data.table, Hmisc, MSstats, MSstatsTMT, MSstatsPTM, MSstatsConvert, gplots, marray, DT, ggrepel, uuid, utils, stats, htmltools, methods, tidyr, grDevices, graphics

Suggests rmarkdown, tinytest, sessioninfo, knitr

VignetteBuilder knitr

biocViews ImmunoOncology, MassSpectrometry, Proteomics, Software, ShinyApps, DifferentialExpression, OneChannel, TwoChannel, Normalization, QualityControl, GUI

BugReports https://github.com/Vitek-Lab/MSstatsShiny/issues

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.2

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

git_branch RELEASE_3_16

git_last_commit adf43d8

git_last_commit_date 2023-01-17

Date/Publication 2023-01-27

Author Devon Kohler [aut, cre],
Maanasa Kaza [aut],
Cristina Pasi [aut],
Ting Huang [aut],
Mateusz Staniak [aut],
Dhaval Mohandas [aut],
Eduard Sabido [aut],
Meena Choi [aut],
Olga Vitek [aut]

Maintainer Devon Kohler <kohler.d@northeastern.edu>

R topics documented:

apply_adj 2
dia_skyline_model 3
dia_skyline_summarized 4
example_dia_skyline 4
example_skyline_annotation 5
groupComparisonPlots2 5
launch_MSstatsShiny 8
lf_model 8
lf_summarization_loop 9
MSstatsShiny 10
QC_check 11
radioTooltip 11
tmt_model 12
tmt_pd_model 13
tmt_pd_summarized 14
tmt_summarization_loop 14
xy_str 15

Index **17**

`apply_adj` *Main PTM adjustment function*

Description

Main PTM function to model MSstatsShiny data.

Usage

```
apply_adj(ptm_model, protein_model)
```

Arguments

ptm_model	output of MSstats modeling function modeling PTMs
protein_model	output of MSstats modeling function modeling unmodified proteins

Value

list of PTM modeling results

Examples

```
model = MSstatsPTM::groupComparisonPTM(MSstatsPTM::summary.data,
                                       data.type = "LabelFree")
apply_adj(model$PTM.Model, model$PROTEIN.Model)
```

---

dia_skyline_model *Example of Sklyine DDA dataset modeled using MSstats* groupComparison *function.*

Description

Data includes one list with two data.tables named ComparisonResult and ModelQC and another list of model details named FittedModel. ComparisonResult shows an overview of all proteins modeled in the system. ModelQC provides a report on the quality control checks of each protein in the dataset.

Format

list

Examples

```
data(dia_skyline_model)
head(dia_skyline_model)
```

dia_skyline_summarized

Example of Sklyine DDA dataset processed using MSstats summarization function.

Description

Data includes one list with two data.tables named `FeatureLevelData` and `ProteinLevelData` and a string value `SummaryMethod`. `FeatureLevelData` shows the unsummarized feature level data. `ProteinLevelData` shows the data summarized up to the protein level and is used for modeling the data.

Format

list

Examples

```
data(dia_skyline_summarized)
head(dia_skyline_summarized)
```

example_dia_skyline

Example of input Sklyine DDA dataset.

Description

Used as input data to MSstats workflow. Data includes one data.table which is the output of Skyline.

Format

data.frame

Details

The raw data (input data for MSstats) is required to contain variable of ProteinName, PeptideSequence, PrecursorCharge, FragmentIon, ProductCharge, IsotopeLabelType, Condition, BioReplicate, Run, Intensity. The variable names should be fixed. If the information of one or more columns is not available for the original raw data, please retain the column variables and type in fixed value. For example, the original raw data does not contain the information of PrecursorCharge and ProductCharge, we retain the column PrecursorCharge and ProductCharge and then type in NA for all transitions in RawData. Variable Intensity is required to be original signal without any log transformation and can be specified as the peak of height or the peak of area under curve.

Examples

```
data(example_dia_skyline)
head(example_dia_skyline)
```

`example_skyline_annotation` *Example annotation file*

Description

data.frame mapping Skyline run names to the corresponding bioreplicates and conditions. Used as input to preprocessing function, converting data into MSstats format.

Format

data.frame

Examples

```
data(example_skyline_annotation)
head(example_skyline_annotation)
```

`groupComparisonPlots2` *Plotting funcitonality for QC plots*

Description

General plotting code to produce all QC plots in the application

Usage

```
groupComparisonPlots2(
  data = data,
  type = type,
  sig = 0.05,
  FCcutoff = FALSE,
  logBase.pvalue = 10,
  ylimUp = FALSE,
  ylimDown = FALSE,
  xlimUp = FALSE,
  x.axis.size = 10,
  y.axis.size = 10,
  dot.size = 3,
```

```
  text.size = 4,
  legend.size = 13,
  ProteinName = TRUE,
  colorkey = TRUE,
  numProtein = 100,
  clustering = "both",
  width = 10,
  height = 10,
  which.Comparison = "all",
  which.Protein = "all",
  address = "",
  savePDF = FALSE
)
```

Arguments

data	'ComparisonResult' in testing output from function groupComparison.
type	choice of visualization. "VolcanoPlot" represents volcano plot of log fold changes and adjusted p-values for each comparison separately. "Heatmap" represents heatmap of adjusted p-values for multiple comparisons. "ComparisonPlot" represents comparison plot of log fold changes for multiple comparisons per protein.
sig	FDR cutoff for the adjusted p-values in heatmap and volcano plot. level of significance for comparison plot. 100(1-sig)% confidence interval will be drawn. sig=0.05 is default.
FCcutoff	for volcano plot or heatmap, whether involve fold change cutoff or not. FALSE (default) means no fold change cutoff is applied for significance analysis. FCcutoff = specific value means specific fold change cutoff is applied.
logBase.pvalue	for volcano plot or heatmap, (-) logarithm transformation of adjusted p-value with base 2 or 10(default).
ylimUp	for all three plots, upper limit for y-axis. FALSE (default) for volcano plot/heatmap use maximum of -log2 (adjusted p-value) or -log10 (adjusted p-value). FALSE (default) for comparison plot uses maximum of log-fold change + CI.
ylimDown	for all three plots, lower limit for y-axis. FALSE (default) for volcano plot/heatmap use minimum of -log2 (adjusted p-value) or -log10 (adjusted p-value). FALSE (default) for comparison plot uses minimum of log-fold change - CI.
xlimUp	for Volcano plot, the limit for x-axis. FALSE (default) for use maximum for absolute value of log-fold change or 3 as default if maximum for absolute value of log-fold change is less than 3.
x.axis.size	size of axes labels, e.g. name of the comparisons in heatmap, and in comparison plot. Default is 10.
y.axis.size	size of axes labels, e.g. name of targeted proteins in heatmap. Default is 10.
dot.size	size of dots in volcano plot and comparison plot. Default is 3.
text.size	size of ProteinName label in the graph for Volcano Plot. Default is 4.
legend.size	size of legend for color at the bottom of volcano plot. Default is 7.

`ProteinName` for volcano plot only, whether display protein names or not. TRUE (default) means protein names, which are significant, are displayed next to the points. FALSE means no protein names are displayed.

`colorkey` TRUE(default) shows colorkey.

`numProtein` The number of proteins which will be presented in each heatmap. Default is 100. Maximum possible number of protein for one heatmap is 180.

`clustering` Determines how to order proteins and comparisons. Hierarchical cluster analysis with Ward method(minimum variance) is performed. 'protein' means that protein dendrogram is computed and reordered based on protein means (the order of row is changed). 'comparison' means comparison dendrogram is computed and reordered based on comparison means (the order of comparison is changed). 'both' means to reorder both protein and comparison. Default is 'protein'.

`width` width of the saved file. Default is 10.

`height` height of the saved file. Default is 10.

`which.Comparison` list of comparisons to draw plots. List can be labels of comparisons or order numbers of comparisons from levels(data$Label), such as levels(testResultMultiComparisons$Compariso Default is "all", which generates all plots for each protein.

`which.Protein` Protein list to draw comparison plots. List can be names of Proteins or order numbers of Proteins from levels(testResultMultiComparisons$ComparisonResult$Protein). Default is "all", which generates all comparison plots for each protein.

`address` the name of folder that will store the results. Default folder is the current working directory. The other assigned folder has to be existed under the current working directory. An output pdf file is automatically created with the default name of "VolcanoPlot.pdf" or "Heatmap.pdf" or "ComparisonPlot.pdf". The command address can help to specify where to store the file as well as how to modify the beginning of the file name. If address=FALSE, plot will be not saved as pdf file but showed in window.

`savePDF` Boolean input passed from user on whether or not to save the plot to a PDF.

Value

PDF or console plot

Examples

```
data("dia_skyline_model")
groupComparisonPlots2(dia_skyline_model$ComparisonResult, type="VolcanoPlot",
                      address=FALSE)
```

launch_MSstatsShiny *Run MSstatsShiny Application*

Description

Main function to run MSstatsShiny. All other functions in this package are run automatically.

Usage

```
launch_MSstatsShiny(
  launch_app = TRUE,
  port = getOption("shiny.port"),
  host = getOption("shiny.host", "127.0.0.1")
)
```

Arguments

launch_app	One of TRUE or FALSE indicating whether or not to run application. Default is TRUE.
port	(optional) Specify port the application should list to.
host	(optional) The IPv4 address that the application should listen on.

Value

Running Shiny Application

Examples

```
## To run app set launch_app=TRUE
launch_MSstatsShiny(launch_app=FALSE)
```

lf_model *Main LF modeling function for MSstatsShiny application*

Description

Main LF function to model MSstatsShiny data.

Usage

```
lf_model(data, contrast.matrix, busy_indicator = TRUE)
```

Arguments

data	summarized data from output of MSstats summarization function.
contrast.matrix	contrast matrix specifying which conditions should be compared
busy_indicator	Boolean indicator indicating whether or not to display shiny waiting indicator.

Value

list of LF modeling results

Examples

```
data("dia_skyline_summarized")
comparison <- matrix(c(1, -1, 0, 0, 0, 0, 0, 0, 0, 0),nrow=1)
row.names(comparison) = "1 vs 128"
colnames(comparison) = c("1", "128", "16", "2", "256",
                         "32", "4", "512", "64", "8")
model_lf_test = lf_model(dia_skyline_summarized, comparison,
                         busy_indicator = FALSE)
```

---

lf_summarization_loop *Main LF calculation summarization function for MSstatsShiny application*

Description

Main LF function to calculate MSstatsShiny results.

Usage

```
lf_summarization_loop(data, input, busy_indicator = TRUE)
```

Arguments

data	Data converted into MSstats format.
input	options for data processing input by the user
busy_indicator	Boolean indicator indicating whether or not to display shiny waiting indicator.

Value

list of LF Summarization results

Examples

```
data("example_dia_skyline")
data("example_skyline_annotation")
testdata = MSstats::SkylinetoMSstatsFormat(example_dia_skyline,
                                           annotation = example_skyline_annotation,
                                           filter_with_Qvalue = TRUE,
                                           qvalue_cutoff = 0.01,
                                           fewMeasurements="remove",
                                           removeProtein_with1Feature = TRUE,
                                           use_log_file = FALSE)

## Source app functionality
input = list()
input$norm = "equalizeMedians"
input$log = 2
input$names = NULL
input$features_used = "all"
code_n_feat=3
input$censInt = "NA"
input$features_used = "all"
input$MBi = TRUE
input$remove50 = FALSE
input$maxQC = 0.999
input$null = FALSE
input$null1 = FALSE
input$DDA_DIA = "LF"

lf_summarization_loop(testdata, input, busy_indicator=FALSE)
```

`MSstatsShiny` *MSstatsShiny: An R-shiny based package for detecting differencially abundant proteins, integrated with the MSstats family of packages.*

Description

A set of tools for detecting differentially abundant proteins in shotgun mass spectrometry-based proteomic experiments. The package can handle a variety of acquisition types, including label free, DDA, DIA, and TMT. The package includes tools to convert raw data from different spectral processing tools, summarize feature intensities, and fit a linear mixed effects model. The GUI supports different biological queries including those targeting the global proteome and post translational modifications. Additionally the package includes functionality to plot a variety of data visualizations.

functions

- `launch_MSstatsShiny` : Main function to launch the application.
- `groupComparisonPlots2` : Generates MSstatsShiny plots.

- lf_summarization_loop : Summarization for LF experiments.
- tmt_summarization_loop : Summarization for TMT experiments.
- lf_model : Modeling for LF experiments.
- tmt_model : Modeling for TMT experiments.

QC_check *Quick QC value check*

Description

Quick QC value check for LF vs TMT

Usage

```
QC_check(input)
```

Arguments

input	options for data processing input by the user

Value

string

Examples

```
input = list(null=TRUE)
QC_check(input)
```

---

radioTooltip *Custom function to create radio tool tips*

Description

Used in UI files to create HTML vizualizations

Usage

```
radioTooltip(
  id,
  choice,
  title,
  placement = "bottom",
  trigger = "hover",
  options = NULL
)
```

Arguments

id	input id
choice	user selection
title	title of object
placement	where should tooltip be shown
trigger	how should prompt be shown
options	additional options to pass to function

Value

HTML object

Examples

```
radioTooltip("testid", "test_choice", "test_title")
```

---

tmt_model *Main TMT modeling function for MSstatsShiny application*

Description

Main TMT function to model MSstatsShiny data.

Usage

```
tmt_model(data, input, contrast.matrix, busy_indicator = TRUE)
```

Arguments

data	summarized data from output of MSstats summarization function.
input	options for data processing input by the user
contrast.matrix	contrast matrix specifying which conditions should be compared
busy_indicator	Boolean indicator indicating whether or not to display shiny waiting indicator.

Value

list of TMT modeling results

Examples

```
data(raw.pd, package = "MSstatsTMT")
data(annotation.pd, package = "MSstatsTMT")

testdata <- MSstatsTMT::PDtoMSstatsTMTFormat(raw.pd,
                                             annotation.pd,
                                             use_log_file = FALSE
                                             )#'
input = list()
input$summarization = "msstats"
input$norm = "equalizeMedians"
input$log = 2
input$names = NULL
input$features_used = "all"
code_n_feat=3
input$censInt = "NA"
input$features_used = "all"
input$MBi = TRUE
input$remove50 = FALSE
input$maxQC = 0.999
input$null = FALSE
input$null1 = FALSE
input$DDA_DIA = "LF"
input$global_norm = TRUE
input$reference_norm = TRUE
input$remove_norm_channel = TRUE
input$maxQC1 = NULL
input$moderated = FALSE

summarization_tmt_test = tmt_summarization_loop(testdata, input,
                                                busy_indicator = FALSE)

comparison=matrix(c(-1,0,0,1),nrow=1)
row.names(comparison) = "1-0.125"
colnames(comparison) = c("0.125", "0.5", "0.667", "1")

model_tmt_test = tmt_model(summarization_tmt_test, input, comparison,
                           busy_indicator = FALSE)
```

---

tmt_pd_model *Example of TMT dataset modeled using MSstatsTMT* groupComparisonTMT *function.*

Description

Data includes one list with two data.tables named ComparisonResult and ModelQC and another list of model details named FittedModel. ComparisonResult shows an overview of all proteins modeled in the system. ModelQC provides a report on the quality control checks of each protein in the dataset.

Format

list

Examples

```
data(tmt_pd_model)
head(tmt_pd_model)
```

---

tmt_pd_summarized *Example of TMT dataset processed using MSstatsTMT summarization function.*

Description

Data includes one list with two data.tables named `FeatureLevelData` and `ProteinLevelData`. `FeatureLevelData` shows the unsummarized feature level data. `ProteinLevelData` shows the data summarized up to the protein level and is used for modeling the data.

Format

list

Examples

```
data(tmt_pd_summarized)
head(tmt_pd_summarized)
```

---

tmt_summarization_loop *Main TMT summarization calculation function for MSstatsShiny application*

Description

Main TMT function to calculate MSstatsShiny results.

Usage

```
tmt_summarization_loop(data, input, busy_indicator = TRUE)
```

Arguments

`data`	Data converted into MSstats format.
`input`	options for data processing input by the user
`busy_indicator`	Boolean indicator indicating whether or not to display shiny waiting indicator.

Value

list of TMT summarization results

Examples

```
data(raw.pd, package = "MSstatsTMT")
data(annotation.pd, package = "MSstatsTMT")

testdata <- MSstatsTMT::PDtoMSstatsTMTFormat(raw.pd,
                                             annotation.pd,
                                             use_log_file = FALSE
                                             )

input = list()
input$summarization = "msstats"
input$norm = "equalizeMedians"
input$log = 2
input$names = NULL
input$features_used = "all"
code_n_feat=3
input$censInt = "NA"
input$features_used = "all"
input$MBi = TRUE
input$remove50 = FALSE
input$maxQC = 0.999
input$null = FALSE
input$null1 = FALSE
input$DDA_DIA = "LF"
input$global_norm = TRUE
input$reference_norm = TRUE
input$remove_norm_channel = TRUE
input$maxQC1 = NULL
input$moderated = FALSE

summarization_tmt_test = tmt_summarization_loop(testdata, input,
                                                busy_indicator = FALSE)
```

---

xy_str *Simple function to return coordinates*

Description

Used in experimental design to create vizualization

Usage

```
xy_str(e)
```

Arguments

e	input function provided by user

Value

Character with x and y coordinates

Examples

```
xy_str(list(x=5.0,y=2.0))
```

Index

apply_adj, 2

dia_skyline_model, 3
dia_skyline_summarized, 4

example_dia_skyline, 4
example_skyline_annotation, 5

groupComparisonPlots2, 5, *10*

launch_MSstatsShiny, 8, *10*
lf_model, 8, *11*
lf_summarization_loop, 9, *11*

MSstatsShiny, 10

QC_check, 11

radioTooltip, 11

tmt_model, *11*, 12
tmt_pd_model, 13
tmt_pd_summarized, 14
tmt_summarization_loop, *11*, 14

xy_str, 15