Package ‘sistmr’

May 9, 2026

Type Package

Title A Collection of Utility Function from the Inserm/Inria SISTM Team

Version 0.1.1

Author Boris Hejblum [aut], Mélanie Huchon [aut, cre]

Maintainer Mélanie Huchon <`melanie.huchon@u-bordeaux.fr`>

Description Functions common to members of the SISTM team.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Imports BlandAltmanLeh, dplyr, ggbeeswarm, ggplot2, ggrepel, rlang, scales, stats

NeedsCompilation no

Repository CRAN

Date/Publication 2022-03-24 08:30:02 UTC

Contents

`BlandAltmanPlot` *Bland-Altman plot function*

Description

Bland-Altman plot function

Usage

```
BlandAltmanPlot(
  var1,
  var2,
  with_gradient = FALSE,
  line_color = c("blue", "lightblue"),
  extremum_pctg = TRUE
)
```

Arguments

`var1`	a vector of numerics for the 1rst group to be compared.
`var2`	a vector of numerics for the 2nd group to be compared.
`with_gradient`	a logical indicating if you have a lot of measures, use `with_gradient=TRUE` to have gradient scale and not points. Default value is FALSE.
`line_color`	a vector of color for the three lines : average difference and upper and lower limits of the confidence interval for the average difference.
`extremum_pctg`	a logical indicating if you want to add the percentage of points outside the confidence interval for the upper and lower limits. Default is TRUE.

Value

a ggplot2 object

Examples

```
library(ggplot2)

#Small sample
#Generate data
x <- rnorm(30)
y <- rnorm(30, mean = 5, sd = 3)
#Plotting
BlandAltmanPlot(var1 = x, var2 = y)
#Add color by group
gr  <- c(rep("G1", 15), rep("G2", 15))
BlandAltmanPlot(var1 = x, var2 = y) + geom_point(aes(color = gr))

#High sample
```

```
#Generate data
x <- rnorm(10000)
y <- rnorm(10000, mean = 5, sd = 3)
#Plotting with gradient
BlandAltmanPlot(var1 = x, var2 = y, with_gradient = TRUE)
```

`multipleBoxplots` *Multiple boxplots for many times*

Description

Multiple boxplots for many times

Usage

```
multipleBoxplots(data, x_var, y_var, add_points = TRUE)
```

Arguments

`data`	a dataset from which the variable `x_var` and `y_var` should be taken.
`x_var`	corresponding to the x coordinates for the plot, it must be a factor to obtain multiple boxplots.
`y_var`	corresponding to the y coordinates for the plot.
`add_points`	if you want to add points on boxplots. Default value is `TRUE`.

Value

a `ggplot2` object

Examples

```
library(ggplot2)

#Generate data
x_ex <- factor(c(rep("J0", 10), rep("J7", 10), rep("J14", 10)), levels = c("J0", "J7", "J14"))
y_ex <- rnorm(30)

data_ex <- cbind.data.frame(x_ex, y_ex)

#Plotting
multipleBoxplots(data = data_ex, x_var = x_ex, y_var = y_ex)

multipleBoxplots(data = data_ex, x_var = x_ex, y_var = y_ex) +
labs(x = "Time", y = "Value") +
theme(legend.position = "none")
```

`normal_distribution` *Functions*

Description

Functions

Usage

```
normal_distribution(vec)
```

Arguments

`vec`	a `vector`

Value

a `vector`

`sistmr` *sistmr.*

Description

This package contains functions common to members of the SISTM team.

`volcanoPlot` *Volcano plot function*

Description

Volcano plot function

Usage

```
volcanoPlot(
  log2fc,
  pValue,
  data,
  FDR_threshold = 0.05,
  LFC_threshold = log2(1.5),
  color = c("red", "black"),
  geneNames = NULL,
  nb_geneTags = 20,
  logTransformPVal = TRUE
)
```

Arguments

`log2fc`	a magnitude of change (fold-change) in base log 2 corresponding to the x-axis.
`pValue`	a statistical significance (p-value) corresponding to the y-axis.
`data`	a data.frame of differentially expressed results from which the variable `log2fc`, `pValue` and `geneNames` (if it is used) should be taken.
`FDR_threshold`	a threshold of false discovery rate.
`LFC_threshold`	a threshold of log fold change.
`color`	a vector of two colors for significant or not significant points.
`geneNames`	a vector of gene names if you want to put gene tags on the volcano plot. Default is NULL.
`nb_geneTags`	number of tags for the significant genes if `geneNames` is not NULL. Default is 20 to obtain the 20 first significant genes.
`logTransformPVal`	If TRUE, the p-values will have a negative logarithm transformation (base 10). Default is TRUE.

Value

a `ggplot2` object

Examples

```
genes <- paste0("G", 1:500)
pval <- runif(500, max = 0.5)
log2FC <- runif(500, min = -4, max = 4)

data <- cbind.data.frame(genes, pval, log2FC)

rm(genes, pval, log2FC)
volcanoPlot(log2FC, pval, data, geneNames = genes)
```

Index