Package 'unifiedWMWqPCR'

November 12, 2026

Type Package

Title Unified Wilcoxon-Mann Whitney Test for testing differential expression in qPCR data

Version 1.48.0

Date 2018-04-23

Author Jan R. De Neve & Joris Meys

Maintainer Joris Meys <Joris.Meys@UGent.be>

Description This packages implements the unified Wilcoxon-Mann-Whitney Test for qPCR data. This modified test allows for testing differential expression in qPCR data.

License GPL (>=2)

Encoding UTF-8

Depends methods

Imports BiocGenerics, limma, stats, graphics

Collate 'Datasets.R' 'S3Generics.R' 'uwmwEstimate_Class.R' 'uwmwRes_Class.R' 'allGenerics.R' 'as.matrix.uwmwEstimate.R' 'as.matrix.uwmwRes.R' 'forestplot_methods.R' 'getEstimate_methods.R' 'groups.message.R' 'outer.h.f.R' 'outer.o.f.R' 'plotMethoduwmwRes.R' 'prepdf.R' 'sandwich1.f.R' 'sandwich2.f.R' 'sandwich3.f.R' 'sort.uwmwRes.R' 'uWMW.H.R' 'uWMW.O.R' 'uWMW_Extract.R' 'uWMW_internal.R' 'uWMW_methods.R' 'unifiedWMWqPCR-package.R' 'uwmw_Accessors.R' 'uwmw_Showmethods.R' 'volcanoplot.R' 'volcano-internal.R' 'wmw.f.R'

biocViews DifferentialExpression, GeneExpression, MicrotitrePlateAssay, MultipleComparison, QualityControl, Software, Visualization, qPCR

RoxygenNote 7.3.2

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

git_branch RELEASE_3_23

git_last_commit 459e7d1

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

unifiedWMWqPCR-package . . . 2
as.matrix.uwmwEstimate . . . 3
as.matrix.uwmwRes . . . 4
forestplot . . . 5
getEstimate . . . 9
NBdata . . . 10
plot,uwmwRes-method . . . 11
type . . . 12
unorder . . . 15
uWMW,matrix-method . . . 16
uwmwEstimate-class . . . 18
uwmwRes-class . . . 19
uwmw_Extract . . . 20
volcanopar . . . 21
volcanoplot . . . 23

Index **25**

`unifiedWMWqPCR-package`

Unified Wilcoxon-Mann Whitney Test for qPCR data.

Description

This package allows you to carry out the unified Wilcoxon-Mann Whitney Test for qPCR data. If you use this package for research purposes, please cite the following article:

Details

De Neve, Jan; Meys, Joris; Ottoy, Jean-Pierre; Clement, Lieven and Thas, Olivier. "UnifiedWMWqPCR: The unified Wilcoxon-Mann-Whitney test for analyzing RT-qPCR data in R." Bioinformatics(2014): Vol.30, Issue 17, pp.2494-2495.

Or see `citation("unifiedWMWqPCR")`

`as.matrix.uwmwEstimate` *S3 method as.matrix for uwmwEstimate objects.*

Description

For `uwmwEstimate` objects, an `as.matrix` method is defined that transforms the object to a numeric matrix with the following columns:

`est`: The estimates

`se`: The standard errors on estimates.

`ll` lower limit of the confidence interval

`ul` upper limit of the confidence interval

The row names are the names of the tested genes/features. The matrix takes the ordering in the object into account if necessary.

Usage

```
## S3 method for class 'uwmwEstimate'
as.matrix(x, ...)

## S4 method for signature 'uwmwEstimate'
as.matrix(x, ...)
```

Arguments

`x`	a uwmwEstimate object
`...`	currently ignored

Value

A matrix with the columns specified above.

Author(s)

Joris Meys

Examples

```
data(NBmat)
NBtest <- uWMW(NBmat, groups=NBgroups)
NBest <- getEstimate(NBtest,"p")
as.matrix(NBtest)
```

`as.matrix.uwmwRes` *S3 method as.matrix for uwmwRes objects.*

Description

For `uwmwRes` objects, an `as.matrix` method is defined that transforms the object to a numeric matrix with the following columns:

`logor`: The log odds ratio values

`se`: The standard errors on the log OR values.

`or` The odds ratio values

`z.value` the z values related to the log OR values

`p.value` The p values related to the log OR values

The row names are the names of the tested genes/features. The matrix takes the ordering in the object into account if necessary.

Usage

```
## S3 method for class 'uwmwRes'
as.matrix(x, ...)

## S4 method for signature 'uwmwRes'
as.matrix(x, ...)
```

Arguments

`x`	a uwmwRes object
`...`	currently ignored

Value

A matrix containing the columns specified above.

Examples

```
data(NBmat)
NBtest <- uWMW(NBmat, groups=NBgroups)
as.matrix(NBtest)
```

forestplot *Making a forest plot of the results of uWMW*

Description

This function creates a forest plot indicating the (log) odds ratios, the (log) odds or the probabilities for the results of the unified Wilcoxon-Mann-Whitney test.

Usage

```
## S4 method for signature 'ANY'
forestplot(x, ...)

## S4 method for signature 'uwmwRes'
forestplot(
  x,
  estimate = c("logor", "logodds", "or", "odds", "p"),
  level = 0.95,
  ...
)

## S4 method for signature 'uwmwEstimate'
forestplot(x, ...)

forestplot.internal(
  x,
  annotate = TRUE,
  addfit = TRUE,
  xlim = NULL,
  alim = NULL,
  ylim = NULL,
  at = NULL,
  steps = 5,
  level = 0.95,
  digits = 2,
  refline = NULL,
  xlab = NULL,
  slab = NULL,
  mlab = NULL,
  ilab = NULL,
  ilab.xpos = NULL,
  ilab.pos = NULL,
  order = NULL,
  transf = FALSE,
  atransf = FALSE,
  targs = NULL,
  rows = NULL,
```

```
  efac = 1,
  pch = 15,
  psize = 1,
  col = "darkgrey",
  border = "darkgrey",
  cex = NULL,
  cex.lab = NULL,
  cex.axis = NULL,
  refcol = "red",
  predcol = refcol,
  ...
)
```

Arguments

x	An object of class uwmwRes or uwmwEstimate.
...	parameters passed down to the internal functions. These can be any of the following.
estimate	An optional character string defining which measure should be plotted. It can take the values logor, or, logodds, odds or p (for displaying the probability of differential expression, not the p-value!). Defaults to logor. Note that this argument is ignored when plotting an uwmwEstimate object.
level	Numerical value between 0 and 1 to specify the width of the confidence interval. Defaults to 0.95 (95% confidence interval).
annotate	A logical value indicating whether the plot needs to be annotated, i.e. whether the values for the chosen measure and confidence interval should be displayed on the right of the plot. Defaults to TRUE
addfit	A logical value indicating whether the reference measure should be plotted. See Details.
xlim	The horizontal limits of the plot region. If unspecified, the function tries to set the horizontal plot limits to some sensible values. Should not be used by the user.
alim	the actual x axis limits. If unspecified, an educated guess is taken by the function.
ylim	The vertical limits of the plot. If unspecified, the function does what it thinks is best. Should not be used by the user.
at	Position of the x axis tick marks and corresponding labels are placed. If unspecified, the function tries to position them at sensible values.
steps	An integer indicating the number of tick marks on the X axis. Ignored when at is specified. Defaults to 5
digits	integer specyfing the number of decimal places for tick mark labels and annotations. Can also be a vector of two integers. In that case, the first value specifies the number of decimal places for the annotations, the second for the x axis labels.
refline	numerical value indicating where a reference line should be drawn. An NA value will prevent the line from being drawn. See Details.

xlab	title for the x axis. If unspecified, the function tries to figure out the fitting title.
slab	optional vector with names for the displayed genes.
mlab	optional character string giving a label to the intercept estimate. If unspecified, this is created in the function if necessary.
ilab	optional vector or matrix with character strings providing additional information that can be plotted next to the genes.
ilab.xpos	Vector of numerical values specifying the x axis positions of the character vectors given via ilab. This has to be specified when ilab is specified.
ilab.pos	integer from 1 to 4 specifying the alignment of the character vector(s) given via ilab (2 is right aligned, 4 is left aligned). Default is to center the labels.
order	optional character string, character vector or numerical vector specifying how the genes should be ordered.
transf	optional argument specifying the name of a function that should be used to transform the observed effect sizes, summary estimates, fitted values and confidence interval bounds (e.g., transf=exp). Defaults to FALSE, which means that no transformation is used.
atransf	optional argument specifying the name of a function that should be used to transform the x-axis labels and annotations (e.g., transf=exp). Defaults to FALSE, which means that no transformation is used.
targs	optional arguments needed by the function specified via transf or atransf.
rows	optional vector specifying the horizontal position for the plotted results. If unspecified, the layout happens automatically. See Details and Examples for more information.
efac	numerical value specifying the vertical expansion of the arrows, summary estimate symbols and Ci limits. Normally the default of 1 should work just fine.
pch	plotting symbol used for the observed effect sizes. By default, it's a filled square.
psize	optional vector with the point sizes for the observed effects. If set to NULL, the point sizes are drawn proportional to the value of the log of the test statistic. Defaults to 1.
col	character string specifying the color used for the individual estimates.
border	character string specifying the color used for the border of the individual estimates.
cex	optional numerical value for expansion of text and symbols. See also par.
cex.lab	Optional numerical value for expansion of the axis title.
cex.axis	Optional numerical value for expansion of the x axis labels.
refcol	Character string specyfying the color of the reference line. Defaults to red.
predcol	character string specifying the color of the estimated reference value. Ignored if estimate is logor or or.

Details

The function has methods for uwmwRes and uwmwEstimate objects. When called for an uwmwRes object, the requested estimate is first calculated using `getEstimate` and the result is passed on to the next method.

Note that in either case, it is not possible to use the function on a subset of either type of object. The subsetting functions for `uwmwRes` and `uwmwEstimate` objects return matrices, and hence necessary information on the reference value is lost. To plot a subset of your data, use the `order` argument as shown in the examples.

Adding a reference value to the plot only makes sense when plotting the log(odds), odds or probabilities. If log(OR) or OR are plotted, `addfit` is set to FALSE.

The default settings plot a reference line at a location depending on the plotted estimate. For log(OR), the line is plotted at `refline = 0`. For OR, the line is plotted at `refline = 1`.

Value

NULL invisibly

Warning

Although the internal function is shown here (merely for illustration of the arguments and defaults), the user shouldn't be calling this one directly. The function is not exported.

Note

Thanks to the work of W.Viechtbauer, `forestplot` provides many possibilities for tweaking and customizing the plots. Many of the arguments work the same as in the function `forest.rma` (`metafor` package). You can always check the help file of `forest.rma` for more illustrations on the different arguments.

This function is currently implemented using an internal function that expects an uwmwEstimate object. In a future version, the internal function will be rewritten to be more generic. This will enable the definition of methods for other classes without need to change the internal function itself.

Author(s)

This code is adapted by Joris Meys from the function `forest.rma` (`metafor` package). The original function is written by W. Viechtbauer.

Examples

```
data(NBmat)
NBtest <- uWMW(NBmat, groups=NBgroups)
sigid <- which(pval(NBtest) < 0.05)
forestplot(NBtest,"logodds",order=sigid)

nameid <- c("hsa-mir-30a-3p","hsa-mir-30a-5p")
forestplot(NBtest,"p",order=nameid,addfit=FALSE,
           refline=NA,main="Comparison 30a")
```

```
forestplot(NBtest,"p",order=nameid,addfit=FALSE,
           refline=0.5,main="Comparison 30a",
           alim=c(0,1),xlim=c(-1,2),at=c(0,0.5,1))
```

`getEstimate` *Extract logor, or, odds and percentages from a uwmwRes object*

Description

With this function you can extract the modelled (log) odds ratio, odds or percentages that represent the chance on differential expression as estimated by the uWMW function. It also allows to extract either the standard error of or the confidence interval around the estimates. See Details for more explanation.

Usage

```
## S4 method for signature 'uwmwRes'
getEstimate(
  x,
  esttype = c("logor", "or", "logodds", "odds", "p"),
  se.fit = (esttype %in% c("logor", "logodds")),
  ci = TRUE,
  drop = TRUE
)
```

Arguments

`x`	an object of the clas uwmwRes
`esttype`	a character string indicating the measure you want to extract. It can take the values `logor` for the log odds ratio, `or` for the odds ratios, `logodds` for the log odds, `odds` for the odds or `p` for the percentages.
`se.fit`	logical value indicating whether the standard errors of the logor or the log odds should be returned as well. Ignored when type has a value different from `logor` or `logodds`. Note that you can also use the accessor `se` to get only the standard errors.
`ci`	numerical value indicating the confidence interval (0.95 is 95% confidence interval). If set to TRUE, the 95 interval is returned. If set to NULL, no confidence interval is returned.
`drop`	a logical value. If set to `TRUE` and neither `se` nor `ci` is calculated, the function returns a vector instead of an uwmwEstimate object.

Details

The function can only calculate standard errors for the log OR and the log odds. In all other cases, `se.fit` is ignored. The function takes into account a possible ordering in the object (see also `sort`). So take into account that you get the estimates in the specified order. In case you want this different, either use the function `unorder` on the object first, or check if any of the `uwmw_Accessors` can help you out.

The argument `se.fit` is mainly to be used to save calculation time. Normally there's no need to set it to `FALSE`.

Value

In general, a `uwmwEstimate` object with the requested estimate. See `uwmwEstimate` for details. In case `drop=TRUE` and neither the standard error nor the confidence interval is calculated, a numeric named vector.

Examples

```
data(NBmat)
NBtest <- uWMW(NBmat, groups=NBgroups)
getEstimate(NBtest,'logodds')
getEstimate(NBtest,'odds',ci=0.9)
```

`NBdata` *Documentation for the dataset NBdata*

Description

The example data used in this package, are a subset of the data provided by Mestdagh et al. (2009). The subset contains quantification cycles of 323 microRNAs in 61 neuroblastoma (NB) tumor samples: 22 MYCN amplified (called MNA) and 39 MYCN single copy samples (called MNSC). The subset was selected so that all microRNAs with a least 85

Details

The data exists in different formats. `NBdata` gives you a data frame with following variables:

`subject`: variable of class `"factor"`, indicating the subject code.

`miRNA`: variable of class `"factor"`, indicating the miRNA code.

`Cq`: variable of class `"numeric"`, containing the cycle information.

`group`: variable of class `"factor"`, indicating the group code.

The data matrix `NBmat` contains the same data in matrix format, where the rows are the different miRNA's and the columns the different subjects. The vector `NBgroups` specifies to which group every column of `NBmat` belongs.

References

Mestdagh, P., P. Van Vlierberghe, A. De Weer, D. Muth, F. Westermann, F. Speleman, and J. Vandesompele (2009) A novel and universal method for microRNA RT-qPCR data normalization. Genome Biology., 10, R64.

De Neve, J. Thas, O. Ottoy, J.P. and Clement L. (2013) An extension of the Wilcoxon-Mann-Whitney test for analyzing RT-qPCR data. Statistical Applications in Genetics and Molecular Biology. 12, 333-346.

Examples

```
# Look at the data frame
data(NBdata)
str(NBdata)

# Look at the matrix and grouping vector
data(NBmat)
str(NBmat)
str(NBgroups)
```

`plot,uwmwRes-method` *Quick forest plot of significantly up- and downregulated features.*

Description

This function plots a forest plot, normally used in meta analysis, to visualize the odds ratios (OR) and confidence intervals resulting from a call to `uWMW`. It is the default plot function for the `uwmwRes` objects. The results are ordered according to significance, and only the significant results are plotted. This function calls `forestplot` directly.

Usage

```
## S4 method for signature 'uwmwRes'
plot(x, y, ...)
```

Arguments

`x`	an uwmwRes object.
`y`	ignored for uwmwRes objects
`...`	arguments passed down to `forestplot`

Value

NULL invisibly

See Also

`forestplot`

Examples

```
data(NBmat)
NBtest <- uWMW(NBmat, groups=NBgroups)
plot(NBtest)
```

---

type *Extract info from uwmwRes and uwmwEstimate objects*

Description

This help file describes different ways to access the slots and values contained in uwmwRes objects resulting from calls to uWMW, and in uwmwEstimate objects resulting from calls to getEstimate.

Usage

```
type(x)

## S4 method for signature 'uwmwRes'
type(x)

## S4 method for signature 'uwmwEstimate'
type(x)

## S4 method for signature 'uwmwRes'
housekeeping(x)

## S4 method for signature 'uwmwEstimate'
housekeeping(x)

## S4 method for signature 'uwmwRes'
names(x)

## S4 method for signature 'uwmwEstimate'
names(x)

## S4 method for signature 'uwmwRes'
logor(x, ordered = TRUE)

## S4 method for signature 'uwmwRes'
se(x, ordered = TRUE)

## S4 method for signature 'uwmwEstimate'
se(x)

## S4 method for signature 'uwmwRes'
oddsRatio(x, ordered = TRUE)
```

```
## S4 method for signature 'uwmwRes'
zval(x, ordered = TRUE)

## S4 method for signature 'uwmwRes'
pval(x, ordered = TRUE)

## S4 method for signature 'uwmwRes'
coef(object)

## S4 method for signature 'uwmwRes'
vcov(object)

## S4 method for signature 'uwmwEstimate'
ref(x)

## S4 method for signature 'uwmwEstimate'
esttype(x)

## S4 method for signature 'uwmwRes'
getOrder(x)

is.unsorted(x, na.rm = FALSE, strictly = FALSE)

## S4 method for signature 'uwmwRes'
orderedBy(x)

## S4 method for signature 'uwmwEstimate'
orderedBy(x)

## S4 method for signature 'uwmwRes'
groupinfo(x)

## S4 method for signature 'uwmwEstimate'
groupinfo(x)

## S4 method for signature 'uwmwEstimate'
length(x)

## S4 method for signature 'uwmwRes'
length(x)
```

Arguments

x	a uwmwRes object or a uwmwEstimate object.
ordered	logical value. If set to FALSE, the extracted values are given in the original order, also after a uwmwRes object has been ordered using sort. Defaults to TRUE
object	see x

na.rm	for compatibility with the base function is.unsorted. Ignored for uwmwRes objects.
strictly	for compatibility with the base function is.unsorted. Ignored for uwmwRes objects.

Value

type(x) returns the type of uWMW carried out (i.e. O or H for using overall respectively housekeeping expression as a reference.)

housekeeping(x) returns the content of the housekeeping slot from the object. Or, in case of overall normalization, it returns NULL.

names(x) returns the names of the genes in the object, and in the order defined in the object.

logor(x) returns a numeric vector with the log OR values.

se(x) returns a numeric vector with the standard errors on the logor.

oddsRatio(x) returns a numeric vector with the odds ratios.

zval(x) returns a numerical vector with the Z values contained in the object.

pval(x) returns a numerical vector containing the p values in the object.

orderedBy(x) returns the slot by which the object is ordered (i.e. the value of the slot orderedBy)

groupinfo(x) returns the groupinfo slot, i.e. a character vector of length 2 that indicates in which order the groups are compared by uWMW.

See Also

uwmw_Extract for matrix like extraction of data.

Examples

```
data(NBmat)
NBtest <- uWMW(NBmat,NBgroups)
coef(NBtest)[1:10]
type(NBtest)

# With a sorted object
NBsort <- sort(NBtest, which="p")
is.unsorted(NBtest)
is.unsorted(NBsort)
orderedBy(NBsort)

# On an Estimate object
NBlogodds <- getEstimate(NBsort,"logodds")
se(NBlogodds)
# and so on...
```

unorder *Sort and order method for uWMWRes objects*

Description

This functions provide sorting functionality for uwmwRes objects. It allows to sort the values in the object in order to get the genes with the highest OR, lowest p value, ... The function uses order underneath

Usage

```
unorder(x, ...)

## S4 method for signature 'uwmwRes'
sort(
  x,
  decreasing = FALSE,
  which = c("or", "p", "se", "name", "z"),
  na.last = TRUE,
  ...
)

## S4 method for signature 'ANY'
unorder(x, ...)

## S4 method for signature 'uwmwRes'
unorder(x, ...)

## S4 method for signature 'uwmwRes'
order(
  ...,
  na.last = TRUE,
  decreasing = FALSE,
  method = c("auto", "shell", "radix")
)
```

Arguments

x	a uwmwRes object
...	currently ignored
decreasing	a logical value indicating whether values should be sorted in increasing or decreasing order.
which	a character value indicating on which values should be used to sort on. The possible values are: "or" for sorting on the odds ratio, "p" for sorting on the p value, "se" for sorting on the standard error or "name" for sorting on the gene names.

na.last	a logical value indicating whether NA results should be sorted at the end. See order for more information.
method	Argument of the generic, ignored here. See also order

Details

The function does not change the internal order, but changes the slot id in the object. This slot is used by other functions to give the requested values

Value

sort returns a sorted uwmwRes object.

See Also

The functions is.unsorted, orderedBy and getOrder to check on the ordered state of a uwmwRes object.

order returns the order of the uwmwRes object.

unorder returns a uwmwRes object with the order removed.

Examples

```
data(NBmat)
NBtest <- uWMW(NBmat, groups=NBgroups)
NBsort <- sort(NBtest,which="name")
NBsort[1:10]
```

---

uWMW,matrix-method *The unified Wilcoxon-Mann-Whitney test for qPCR data*

Description

This function carries out the unified Wilcoxon-Mann-Whitney test for qPCR data. See De Neve et al. (2013) for more details.

Usage

```
## S4 method for signature 'matrix'
uWMW(
  x,
  groups,
  housekeeping.names = NULL,
  transpose = FALSE,
  feat.names = NULL
)

## S4 method for signature 'data.frame'
uWMW(x, groups, feat.names, subjects, value, ...)
```

Arguments

x	An object containing the qPCR measurements. See details.
groups	A vector indicating the groups that need comparing, or a single character telling which variable in the data frame contains the groups. Make sure this vector is as long as the number of replicates in the data set.
housekeeping.names	an (optional) vector with the names of one or more housekeeping features. Make sure those names are spelled exactly as in the object.
transpose	In case a matrix is used, should the matrix be transposed? A matrix needs to be transposed when the columns do not represent the replicates. The function expects the columns to be replicates and the rows to be the different features.
feat.names	An (optional) character vector with the names of the features (typically genes or microRNAs) or a single character giving the name of the feature variable. If not specified, the feature names are derived from the row names of the matrix, or from the feature names of the qPCRset object.
subjects	An (optional) character string indicating which variable of the data frame contains the subject id's. Ignored if x is not a data frame.
value	An (optional) character string indicating which variable of the data frame contains the values. Ignored if x is not a data frame.
...	For passing arguments between methods and to internal functions.

Details

This function carries out the unified Wilcoxon-Mann-Whitney test using either Overall normalization (O) or Housekeeping normalization (H) as reference (see De Neve et al., 2013). If the argument `housekeeping.names` is specified, housekeeping normalization is considered. Otherwise overall normalization is considered.

The function uWMW can deal with data frames and matrices. When using a data frame, you need to specify the arguments `groups`, `feat.names`, `subjects` and `value`; each one should contain the name of the related variable in the data frame.

When using a matrix, each column is assumed to be a subject and each row a feature. The argument `groups` should contain as much values as there are columns in the matrix.

Value

An object of the class `uwmwRes`, containing the results of the unified Wilcoxon-Mann-Whitney test. See the help page of the class `uwmwRes` for more information.

Author(s)

Wrapper methods are written by Joris Meys. Internal functions are written by Jan De Neve.

References

De Neve, J. Thas, O. Ottoy, J.P. and Clement L. (2013) An extension of the Wilcoxon-Mann-Whitney test for analyzing RT-qPCR data. Statistical Applications in genetics and Molecular Biology. 12, 333-346

See Also

uwmw_Accessors and uwmw_Extract for accessing the results, and volcanoplot and forestplot for plotting them.

Examples

```
data(NBmat)
NBtest <- uWMW(NBmat, groups=NBgroups)

data(NBdata)
NBtest <- uWMW(x = NBdata, groups = "group", sub="subject",feat="miRNA",val="Cq")
head(NBtest)
as.matrix(NBtest)
```

---

uwmwEstimate-class *The class uwmwEstimate*

Description

This class represents an estimate object resulting from a call to getEstimate. It contains all information about the estimate, including standard errors and confidence intervals if requested. For this class a number of methods is foreseen, including the accessors for the slots. The class is sortable and can be indexed, so you can use this for making custom forest plots using the function forestplot.

Details

#' @section Slots:

esttype: object of class "character", containing the estimate type. This can be logor for log odds ratio, or for odds ratio, odds, logodds for the log odds or p for the probability.

names: object of class "character", containing the names of the genes for which the estimates are calculated.

est: object of class "numeric", containing the estimates itself.

se: object of class "numeric", containing the estimates for the standard error, if applicable.

ll: object of class "numeric", containing the lower limit of the confidence interval.

ul: object of class "numeric", containing the upper limit of the confidence interval.

refest: object of class "numeric", containing the estimate for the reference used in the analysis. Note that this only makes sense for log odds, odds and probabilities.

refse: object of class "numeric", containing the se estimate for the reference if applicable.

refll: object of class "numeric", containing the lower limit for the reference if applicable.

reful: object of class "numeric", containing the upper limit for the reference if applicable.

type: vector of class "character", containing the type of reference used in the original analysis. This can be either "O" or "H" for Overall respectively Housekeeping Expression as reference.

`confint`: vector of class `"numeric"`, indicating the limit used for the confidence interval. 0.95 represents the 95% confidence interval.

`housekeeping`: object of class `"character"`, containing either NULL or the names of the housekeeping genes used in the H version of `uWMW`.

`groupinfo`: character vector of length 2, indicating the groups. This slot is mainly used to show how the probabilistic indices are calculated.

Note

For this class, `show` and `length` methods are defined. `length` will give you the number of features.

Author(s)

Joris Meys

`uwmwRes-class` *Class uwmwRes*

Description

This class represents the results of the unified Wilcoxon-Mann-Whitney test. It contains all necessary information for the vulcano and forest plots. For this class a number of methods is foreseen, among which accessors for every slot.

Slots

`type`: object of class `"character"`, containing the type of analysis (either H or O, see `uWMW` for more details.)

`housekeeping`: object of class `"character"`, containing either NULL or the names of the housekeeping features used in the HME version of `uWMW`.

`names`: object of class `"character"`, containing the names of the features that were used in the test.

`logOR`: object of class `"numeric"`, containing the estimated log odds ratios from the uWMW test.

`se`: object of class `"numeric"`, containing the standard errors on the estimated log odds ratios.

`OR`: object of class `"numeric"`, containing the odds ratios estimated by the uWMW test. This slot is accessed using the function `oddsRatio()`

`z.value`: object of class `"numeric"`, containing the Z values related to the odds ratios estimated by the uWMW test. These z-values relate to the chance that a specific feature is up- or down-regulated, and are used as the basis for determining the p values.

`p.value`: object of class `"numeric"`, containing the p values related to the odds ratios estimated by the uWMW test. These p-values relate to the chance that a specific feature is up- or down-regulated.

`coef`: object of class `"numeric"`, containing the estimated coefficient of the PIM model that's used in the uWMW test.

`vcov`: matrix of class `"numeric"`, containing the variance-covariance matrix related to the estimated coefficients.

`id`: vector of class `"numeric"`, containing the sorting order of the features. This slot is set using the function `sort`

`orderedBy`: character value, indicating whether the object contains an order and if so, based on which slot. Possible values are "none", "p", "z", "or", "se" or "name". Defaults to "none".

`groupinfo`: character vector of length 2, indicating the groups. This slot is mainly used to show how the probabilistic indices are calculated.

Note

For this class, `show` and `length` methods are defined. `length` will give you the number of features.

Author(s)

Joris Meys

See Also

`uwmw_Accessors`

`uwmw_Extract` *Extract data from uwmwRes and uwmwEstimate objects.*

Description

For both uwmwRes and uwmwEstimate objects, you can use the square bracket operators to extract information much like you would do for a matrix.

Usage

```
## S4 method for signature 'uwmwRes,character'
x[i, j, drop = TRUE]

## S4 method for signature 'uwmwEstimate,character'
x[i, j, drop = TRUE]

## S4 method for signature 'uwmwRes,ANY'
x[i, j, drop = TRUE]

## S4 method for signature 'uwmwEstimate,ANY'
x[i, j, drop = TRUE]
```

Arguments

x	a uwmwRes object
i	numeric or character vector
j	optional numeric or character vector
drop	logical value. If set to FALSE and the result of the extraction is a single row or column, dimensions are dropped. Defaults to TRUE.

Details

The features can be extracted using the row index, and the estimates as

Value

mostly a matrix, unless drop=TRUE and a single row or column is selected. Then a vector.

Examples

```
# With an uwmwRes object
data(NBmat)
NBtest <- uWMW(NBmat,NBgroups)
# These two lines are the same
NBtest["hsa-mir-1"]
NBtest["hsa-mir-1", ]
# These two not
str(NBtest["hsa-mir-1",,drop=FALSE])
str(NBtest["hsa-mir-1",])
# These two give the same data, but in a different way:
se(NBtest) # unnamed
NBtest[,"se"] #

# With an uwmwEstimate object
NBodds <- getEstimate(NBtest,"odds")
gnames <- grep("let",names(NBodds),value=TRUE)
NBodds[gnames]
NBodds[gnames,c("ll","ul")]
```

---

volcanopar *Parameters used for the function volcanoplot*

Description

The parameters described can all be used in the different volcanoplot methods.

Arguments

highlight	an integer, indicating how many of the top features should be highlighted. Defaults to 0.
names	an optional character vector giving the names of the genes to be highlighted. If not specified, the features are numbered from most significant to least significant. Default value is NULL.
xlab	see par. Defaults to "Fit"
ylab	see par. Defaults to "Significance"
pch	see par. Defaults to 16
cex	see par. Defaults to 0.35 as in the limma package.
transf.x	a function used to transpose the values on the X axis. Defaults to NULL, unless otherwise specified in the specific methods.
transf.y	a function used to transpose the values on the Y axis. Defaults to NULL, unless otherwise specified in the specific method.
add.ref	a charater value indicating if reference lines for x, y or both axes should be drawn. It takes the values "none", "x", "y" or "both". Default value is "none". If not specified but ref.x or ref.y is, then the reference line is drawn for the X and/or the Y axis.
ref.x	a numerical vector, indicating the value or values at which the reference line(s) should be drawn on the X axis. Defaults to NULL.
ref.y	a numerical value, indicating the value at which the reference line should be drawn on the Y axis. Defaults to NULL unless otherwise specified in the specific methods.
col.x	The color for the reference line on the X axis. The argument is passed to the col argument of lines, so check that help page for possible values. Defaults to "darkgrey"
col.y	see col.x, but for the Y axis. If col.x is specified and col.y is not, then the value of col.x is used for col.y.
lwd.x	The line width for the reference line on the X axis. The argument is passed to the lwd argument of lines, so check that help page for possible values. Defaults to 1.
lwd.y	see lwd.x, but for the Y axis. If lwd.y is not specified and lwd.x is, that value is used for lwd.y as well.
lty.x	The line type for the reference line on the X axis. The argument is passed to the lty argument of lines, so check that help page for possible values. Defaults to 1.
lty.y	see lty.x, but for the Y axis. If lty.y is not specified and lty.x is, that value is used for lty.y as well.

Details

The arguments highlight and names are chosen in such a way that the function can mimick the behaviour of the volcanoplot function in the package limma.

Note

These parameters belong to the internal function, eventually called by the different S4 methods. This internal function is not exported, and should not be called directly.

Examples

```
# see the help page of volcanoplot
```

`volcanoplot` *Make a volcano plot of the outcome of a uWMW test*

Description

This function creates a volcano plot of the outcome of a uWMW test. In this plot, the p-value is plotted against the odd ratio or the log-odds ratio, depending on what interests you most. It allows to quickly see what proportion of results is significant, and what proportion of significant results has a biologically significant chance of being upregulated.

Usage

```
## S4 method for signature 'uwmwRes'
volcanoplot(
  fit,
  esttype = c("logor", "or", "logodds", "odds", "p"),
  transf.y = function(i) -log10(i),
  ref.y = -log10(0.05),
  ...
)

## S4 method for signature 'matrix'
volcanoplot(fit, ...)

## S4 method for signature 'numeric'
volcanoplot(fit, ...)

## S4 method for signature 'MArrayLM'
volcanoplot(fit, coef = 1, highlight = 0, ...)
```

Arguments

`fit`	a uwmwRes object, a matrix with 2 columns or a numeric vector with either odds ratios or the log of odds ratios.
`esttype`	The estimate that should be plotted from the uwmwRes object. See also `getEstimate`.
`transf.y`	a function used to transpose the values on the Y axis. For uwmwRes objects, defaults to -log10(y). Passed down to internal function.

ref.y	The location on the Y axis for the reference line.
...	arguments passed to the internal function. See volcanopar
coef	for compatibility with limma package, see volcanopar
highlight	for compatibiity with limma package, see volcanopar

Details

The methods described here all use the same internal function to create the plot

Value

invisible NULL

Author(s)

Joris Meys

Examples

```
data(NBmat)
housekeeping.id <- grep("let",rownames(NBmat),value=TRUE)
NB.Htest <- uWMW(NBmat,NBgroups,housekeeping.id)

volcanoplot(NB.Htest)
volcanoplot(NB.Htest,"p")
```

Index

∗ **data**
NBdata, 10
[,uwmwEstimate,ANY-method
(uwmw_Extract), 20
[,uwmwEstimate,character-method
(uwmw_Extract), 20
[,uwmwRes,ANY-method (uwmw_Extract), 20
[,uwmwRes,character-method
(uwmw_Extract), 20

as.matrix,uwmwEstimate-method
(as.matrix.uwmwEstimate), 3
as.matrix,uwmwRes-method
(as.matrix.uwmwRes), 4
as.matrix.uwmwEstimate, 3
as.matrix.uwmwRes, 4

coef (type), 12
coef,uwmwRes-method (type), 12

esttype (type), 12
esttype,uwmwEstimate-method (type), 12

forestplot, 5, *11*, *18*
forestplot,ANY-method (forestplot), 5
forestplot,uwmwEstimate-method
(forestplot), 5
forestplot,uwmwRes-method (forestplot),
5
forestplot.internal (forestplot), 5

getEstimate, *8*, 9, *12*, *18*, *23*
getEstimate,uwmwRes-method
(getEstimate), 9
getOrder, *16*
getOrder (type), 12
getOrder,uwmwRes-method (type), 12
groupinfo (type), 12
groupinfo,uwmwEstimate-method (type), 12
groupinfo,uwmwRes-method (type), 12

housekeeping (type), 12
housekeeping,uwmwEstimate-method
(type), 12
housekeeping,uwmwRes-method (type), 12

is.unsorted, *14*, *16*
is.unsorted (type), 12
is.unsorted,uwmwRes-method (type), 12

length, *19*, *20*
length,uwmwEstimate-method (type), 12
length,uwmwRes-method (type), 12
lines, *22*
logor (type), 12
logor,uwmwRes-method (type), 12

names,uwmwEstimate-method (type), 12
names,uwmwRes-method (type), 12
NBdata, 10
NBgroups (NBdata), 10
NBmat (NBdata), 10

oddsRatio (type), 12
oddsRatio,uwmwRes-method (type), 12
order, *15*, *16*
order,uwmwRes-method (unorder), 15
orderedBy, *16*
orderedBy (type), 12
orderedBy,uwmwEstimate-method (type), 12
orderedBy,uwmwRes-method (type), 12

par, *7*, *22*
plot,uwmwRes,ANY-method
(plot,uwmwRes-method), 11
plot,uwmwRes-method, 11
pval (type), 12
pval,uwmwRes-method (type), 12

ref (type), 12
ref,uwmwEstimate-method (type), 12

se, *9*
se *(*type*)*, 12
se,uwmwEstimate-method *(*type*)*, 12
se,uwmwRes-method *(*type*)*, 12
show, *19, 20*
sort, *10*, *13*
sort *(*unorder*)*, 15
sort,uwmwRes-method *(*unorder*)*, 15

type, 12
type,uwmwEstimate-method *(*type*)*, 12
type,uwmwRes-method *(*type*)*, 12

unifiedWMWqPCR
*(*unifiedWMWqPCR-package*)*, 2
unifiedWMWqPCR-package, 2
unorder, *10*, 15
unorder,ANY-method *(*unorder*)*, 15
unorder,uwmwRes-method *(*unorder*)*, 15
uWMW, *11, 12*, *19*
uWMW *(*uWMW,matrix-method*)*, 16
uWMW,data.frame-method
*(*uWMW,matrix-method*)*, 16
uWMW,matrix-method, 16
uwmw_Accessors, *10*, *18*, *20*
uwmw_Accessors *(*type*)*, 12
uwmw_Extract, *14*, *18*, 20
uwmwEstimate, *3*, *10*, *12*
uwmwEstimate *(*uwmwEstimate-class*)*, 18
uwmwEstimate-class, 18
uWMWqPCR *(*unifiedWMWqPCR-package*)*, 2
uwmwRes, *4*, *11, 12*, *15–17*, *21*
uwmwRes *(*uwmwRes-class*)*, 19
uwmwRes-class, 19

vcov *(*type*)*, 12
vcov,uwmwRes-method *(*type*)*, 12
volcano-par *(*volcanopar*)*, 21
volcanopar, 21, *24*
volcanoplot, *18*, *21*, 23
volcanoplot,MArrayLM-method
*(*volcanoplot*)*, 23
volcanoplot,matrix-method
*(*volcanoplot*)*, 23
volcanoplot,numeric-method
*(*volcanoplot*)*, 23
volcanoplot,uwmwRes-method
*(*volcanoplot*)*, 23
volcanoplot-par *(*volcanopar*)*, 21
volcanoplotpar *(*volcanopar*)*, 21

zval *(*type*)*, 12
zval,uwmwRes-method *(*type*)*, 12