Package: gMWT (via r-universe)

October 25, 2026

Type Package

Title Generalized Mann-Whitney Type Tests

Version 1.4

Date 2023-04-14

Maintainer Daniel Fischer <daniel.fischer@luke.fi>

Depends R (>= 3.2), clinfun, Rcpp (>= 0.9.13)

LinkingTo Rcpp, RcppArmadillo

Description Generalized Mann-Whitney type tests based on probabilistic indices and new diagnostic plots, for the underlying manuscript see Fischer, Oja (2015) <doi:10.18637/jss.v065.i09>.

License GPL (>= 2)

LazyLoad yes

NeedsCompilation yes

Author Daniel Fischer [aut, cre] (<https://orcid.org/0000-0002-7513-683X>), Hannu Oja [aut, ths]

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

Date/Publication 2023-04-19 09:20:03 UTC

RemoteUrl https://github.com/cran/gMWT

RemoteRef HEAD

RemoteSha ccc32eefcb99f770922b303a214793d6e148613b

Contents

gMWT-package 2
createGroups 3
estPI 4
getSigTests 6
gmw 8
plot.estPI 11
plot.re 12

plotPI 13
print.estPI 14
print.re 15
rejectionPlot 16
summary.re 18

Index **19**

---

gMWT-package *Implementations of Generalized Mann-Whitney Type Tests*

Description

The package provides nonparametric tools for the comparison of several groups/treatments when the number of variables is large. The tools are the following.

(i) Computation of the so-called probabilistic index (PI) estimates for the group comparisons. The probabilistic indices here are (a) the probability $\hat{P}_t$ that a random observation from group t is smaller than a random observation from other groups, (b) the probability $\hat{P}_{tt'}$ that a random observation from group t is smaller than a random observation from group t', and (c) the probability $\hat{P}_{tt't''}$ that random observations from groups t, t', t'' appear in this same order. The tools are also given to produce the plots of PIs for set of variables.

(ii) Computation of the p-values of some classical and some new nonparametric tests for the comparison of several groups/treatments. The tests are based on the use of the probabilistic indices (PI) $\hat{P}_t$, $\hat{P}_{tt'}$ and $\hat{P}_{tt't''}$. Classical Wilcoxon-Mann-Whitney test, Kruskal-Wallis and Jonckheere-Terpstra tests are included.

(iii) Tools for the simultaneous testing problem. As the package is meant for the analysis of gene expression data, for example, the plots of the numbers of observed and expected rejected null hypotheses and their ratio as a function of alpha are provided. Different multiple testing adjustment methods are implemented, i.a. Bonferroni, Bejamini & Hochberg and Westfall & Young.

Details

Package:	GMW
Type:	Package
Version:	1.4
Date:	2023-04-14
License:	GPL
LazyLoad:	yes

Author(s)

Daniel Fischer, Hannu Oja

Maintainer: Daniel Fischer <daniel.fischer@luke.fi>

References

Daniel Fischer, Hannu Oja (2015). Mann-Whitney Type Tests for Microarray Experiments: The R Package gMWT. Journal of Statistical Software, 65(9), 1-19. URL http://www.jstatsoft.org/v65/i01/.

Fischer, D., Oja, H., Schleutker, J., Sen, P.K., Wahlfors, T. (2013): Generalized Mann-Whitney Type Tests for Microarray Experiments, Scandinavian Journal of Statistic. doi: 10.1111/sjos.12055.

`createGroups` *Renumbering the Group Labels*

Description

This function transforms the k group labels to integers 1,...,k in a specified order.

Usage

```
createGroups(g, desOrder)
```

Arguments

`g`	Vector with group labels.
`desOrder`	Specified order for the groups.

Details

This function helps to renumber the group labels in a specified way. The vector `desOrder` gives the old numbers as the function of the new numbers. The first element of `desOrder` gives the old number that is transformed to be one, and so on. The renumbering is used to specify alternative hypotheses.

The output is then a transformed vector g with new group numbers.

For non-numeric old group labels, the old number is the rank in the lexicographical order.

Value

A vector containing the new group numbers.

Author(s)

Daniel Fischer

Examples

```
g <- c(rep(1,5),rep(2,6),rep(3,4))
createGroups(g,c(2,3,1))

# 'Smallest' letter should become 3, second smallest 2 and largest 1:
g <- sample(c("A","B","C"),10,replace=TRUE)
createGroups(g,c(3,2,1))
```

`estPI` *Calculate Probabilistic Index*

Description

This function calculates the probabilistic indices $\hat{P}_t$, $\hat{P}_{tt'}$ and $\hat{P}_{tt't''}$ to compare the groups of observations.

Usage

```
estPI(X, g, type="pair", goi=NULL, mc=1, order=TRUE, alg="Cnaive")
```

Arguments

`X`	Matrix or vector with observations. The rows refer to individuals, the columns to variables.
`g`	Vector of group numbers for observations in `X`. Its length has to be the number of observations in `X`.
`type`	Type of probabilistic index, see details.
`goi`	Groups Of Interest, see details.
`mc`	Multiple Cores, set the amount of cores to use for parallel calculation (only available for Linux).
`order`	Boolean, calculate probabilistic index only for a specified order of groups or for all possible orders.
`alg`	Internal function, which algorithm should be used to calculate the probabilistic index.

Details

The matrix `X` contains the data. Each column refers to a variable, each row to an observation. The group memberships of the observations are given in `g`. In the case of one dimensional data, `X` is a vector.

There is also an option to calculate the probabilistic indices (PI) only for a subset of the groups. In that case those group labels can be specified with the `goi` option by giving the corresponding group labels.

Different types of PI can be calculated: `"single"` calculates the probability $\hat{P}_t$ for each group, `"pair"` produces the probabilistic indices $\hat{P}_{tt'}$ for all pairs of groups $t < t'$, and `"triple"` provides the probabilities $\hat{P}_{tt't''}$ for all triples of groups $t < t' < t''$. See Fischer et al. (2013) for more details.

Note that the PIs are calculated using the group numbering as given in `g`. See also the function `createGroups` for renumbering the group labels. By specifying the option `order=FALSE` the PIs for all possible group orders will be calculated. The default is that the PI is only calculated for the natural order given by `g`.

In case this code is executed on a Linux machine and `X` is a data matrix, the calculation can be parallelized using the option `mc` to specify the amount of used calculation cores.

Different algorithms are available for the calculation of the PIs. The default is the fastest possible option and the user should not change this option. Different algorithms are only provided for validation and testing purposes. Options here are currently `Cnaive`, `Rnaive`, `Rgrid`, `Rsub`, `Csub` but not all combinations of `type`/`alg` are available.

Value

A list with class 'estPI' containing the following components:

`probs`	Matrix or vector of the PIs.
`type`	String, storing the type of PI.
`goi`	Vector, the Groups Of Interest, as given in the function call.
`order`	Boolean, PI just for the specified order or for all orders.
`obs`	Matrix, the original data matrix.
`g`	Vector, the original group vector.
`alg`	String, the requested algorithm.

Author(s)

Daniel Fischer

References

Fischer, D., Oja, H., Schleutker, Sen, P.K., J., Wahlfors, T. (2013): Generalized Mann-Whitney Type Tests for Microarray Experiments, Scandinavian Journal of Statistics, to appear.

Fischer, D., Oja, H. (2013): Mann-Whitney Type Tests for Microarray Experiments: The R Package gMWT, submitted article.

See Also

`createGroups`

Examples

```
X <- c(sample(15))
g <- c(1,1,1,2,2,2,2,3,3,3,4,4,4,4,4)
estPI(X,g,type="single")

X <- matrix(c(rnorm(5000,1.5,2),rnorm(6000,2,2),rnorm(4000,3.5,1)),byrow=TRUE, ncol=10)
colnames(X) <- letters[1:10]
g <- c(rep(1,500),rep(2,600),rep(3,400))

estPI(X,g,type="single",mc=1)
```

`getSigTests` *Extract Significant Test Results for a Given Alpha or Different Multiple Testing Corrections.*

Description

This function finds the p-values less than or equal to `alpha`. Alpha is as given or corresponds to an adjusted alpha. Different multiple testing correction methods to adjust alpha are available.

Usage

```
getSigTests(pvalues, alpha=0.05, method="plain")
```

Arguments

`pvalues`	A matrix, vector or an object of class `gmw`.
`alpha`	Significance level.
`method`	Method of multiple testing adjustment.

Details

The argument `pvalues` is the result of a `gmw` test run or a matrix/vector of p-values. If `pvalues` is in matrix shape (several test methods applied to the same data), then each row corresponds to a different test method and the columns to different variables. The option `method` specifies the method of multiple testing correction. Typical options are `"plain"` for no correction, `"bonferroni"` for a Bonferroni correction, `"simes"` for an improved Bonferroni correction and `"BH"` for a Benjamini-Hochberg correction. Please note that the Simes method and the Benjamini-Hochberg correction lead to the same results. For permutation test results there is also the Westfall and Young method `"maxT"` available. In order to perform this correction the option `keepPM=TRUE` has to be set in the `gmw` call in order to keep the required permutation matrix.

Additional there are the two options `"csD"` and `"csR"`. Those calculate for each alpha between 0 and `alpha` the difference (`="csD"`) or the ratio (`="csR"`) between observed and expected rejections and report an optimal alpha (and corresponding test rejections) for which these criteria are maximal. Please keep in mind that this method might return unreasonable large 'optimal' cut-off points.

See also the function `rejectionPlot` for more details.

Value

A list object of class 're' with the values: (In case `pvalues` is a matrix the output is a list with length `nrow(X)` and each list item contains a single `re` object)

`sigTests`	Position of the significant tests.
`sigPvalues`	P-values of the significant tests.
`pvalues`	The original `pvalues` object.
`method`	Chosen method.

alpha	Chosen alpha.
multAlpha	Adjusted / Optimal alpha.
inputN	Rows of pvalues.

Author(s)

Daniel Fischer

References

Benjamini, Y. and Hochberg, Y. (1995): Controlling the false discovery rate: a practical and powerful approach to multiple testing. J.R. Statist. Soc. B, 57(1):289 - 300.

Simes, R. J. (1986): An improved bonferroni procedure for multiple tests of significance. Biometrika, 73:751 - 754.

Westfall, P.H. and Young, S.S. (1993): Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment. Wiley, New York.

See Also

rejectionPlot, gmw

Examples

```
X <- matrix(c(rnorm(500,2,1),rnorm(600,2,1),rnorm(400,2.2,1)),byrow=TRUE, ncol=10)
colnames(X) <- letters[1:10]
g <- c(rep(1,50),rep(2,60),rep(3,40))
test <- gmw(X,g,test="kw",type="external")
cs1 <- getSigTests(test)
cs1

simData <- runif(1000,0,1)
simData <- c(simData,runif(200,0,0.01))
simData2 <- runif(1000,0,1)
simData2 <- c(simData2,runif(200,0,0.01))

simDataMat <- rbind(simData,simData2)

getSigTests(simDataMat, method="bon")
getSigTests(simData, method="bon")
getSigTests(simData2, method="bon")
getSigTests(simData, method="sim")
getSigTests(pvalues=simData, method="bh",alpha=0.05)
getSigTests(pvalues=simData, method="csD")
getSigTests(pvalues=simData, method="csR")

set.seed(731)
X <- matrix(c(rnorm(50,2,1),rnorm(60,2,1),rnorm(40,2.2,1)),byrow=TRUE, ncol=10)
colnames(X) <- letters[1:10]
g <- c(rep(1,5),rep(2,6),rep(3,4))
X[12:15,1] <- X[12:15,1] + 5
```

```
# Keep the permutation matrix in order to perform the W&Y adjustment
  testPM <- gmw(X,g,test="kw",type="perm",keepPM=TRUE)
# Apply the Westfall& Young adjustment
  getSigTests(testPM,method="maxT")
```

---

gmw *Generalized Mann-Whitney Type Tests*

Description

The function calculates p-values for different tests as presented in the paper "Generalized Mann-Whitney Type Tests for Microarray Experiments".

Usage

```
gmw(X,g,goi=NULL,test="mw",type="permutation",prob="pair",nper=2000,
alternative="greater",mc=1,output="min", cluster=NULL, order = TRUE,
keepPM= FALSE, mwAkw=FALSE, alg=NULL)
```

Arguments

X	Data matrix, each column corresponds to a variable, each row to an individual. Can also be a vector (one variable).
g	Vector of length nrow(X) (respective length(X)), assigning treatment groups (numbers) to observations, see details.
goi	Vector with elements of g, defining for which treatment groups the test should be performed.
test	Specifies the test statistic.
type	Permutation test ("permutation") or asymptotic tests("asymptotic") for the calculation of the p-values. Tests implemented in R-base "external" are also accessible, see details.
prob	This option is only for the Mann-Whitney test, see details.
nper	If type is "permutation" this option specifies how many permutations are used to calculate the p-value.
alternative	Specifies the alternative, the options are "smaller","greater" and "two.sided", see details.
mc	Multiple Cores, determines how many tests will be performed parallel (only available under Linux), see details.
output	Determines the level of the details in the output.
cluster	A vector of same length as g, giving possible cluster information for performing the permutation test.
order	Boolean, shall all orders be calculated or only increasing orders?

keepPM	Boolean, keep the permutation matrices, required for Westfall & Young multiple testing adjustment.
mwAkw	Boolean or numeric, if TRUE pairwise Mann-Whitney tests are performed after the Kruskal-Wallis test. If numeric MW tests are only performed for KW tests with smaller p-value than that value.
alg	Internal function, what permutation algorithm should be used. Shouln't be changed by the user.

Details

The object X is the data vector (one variable) or the data matrix. Each row refers to an observation, and each column to a variable. The tests are performed separately for all variables.

The vector g gives the group number. The directional tests are based on this numbering of the group.

The goi option defines, which treatment groups are used in the test constructions. If no groups are specified (default), all groups are used.

The test option specifies the test statistic. Possible options are 'uit'(union intersection test), 'triple' (test based on triple indicator functions), 'jt' for Jonckheere-Terpstra test, 'jt*' for a modified Jonckheere-Terpstra test, 'mw' for the Mann-Whitney / Wilcoxon test and 'kw' for the Kruskal-Wallis test. See also reference [1] for further details.

The option type is used to decide how the p-values are computed. For all tests are permutation type tests available and the option for that is type="permutation". In addition for test='mw', test='kw' or test='jt' also the option type='external' is available. This calls then the code from the base system or other, imported packages. For test='uit' there is also an asymptotic test (type="asymptotic") available. For test='triple' or test='jt*' asymptotic implementations are currently under development.

The prob option is only for the Mann-Whitney test. For the option "single", the tests are to compare a single group versus all the other groups. The option "pair" makes all pair-wise comparisons between the groups.

The option alternative is used to specify whether one-sided or two-sided alternatives are used. If the test is based on the PIs, the option "greater" for example means that, according to the alternative, the groups with larger group numbers tend to have larger observations as well. The function createGroups may be used to renumber the groups, if needed.

The mc option is only valid if X is a matrix and the used OS is Linux, because the parallelisation is based on the package parallel, and that again is based on the concept of forking, which is currently only supported under Linux.

The option output can be used to control how detailed the output is. The default "min" reports just the matrix of p-values in a matrix (columns=variables, rows=tests). If output="full", a list will be returned with items containing full test objects of class htest.

The option cluster is an additional object for the Kruskal-Wallis permutation test. For cluster-dependent observation, only the permutations within clusters are acceptable for the p-value calculation.

In the getSigTests function it is possible to apply the Westfall & Young multiple testing method. In this approach the permutation matrix is used to adjust for multiple testing, hence if one wishes to apply this method, the only option for type is "permutation". In addition the boolean flag keepPM

has to be set to TRUE. Is default is to drop the permutation matrix after each run in order to save memory.

If a Kruskal-Wallis test is performed, there is also the option to perform afterwards paiwise Mann-Whitney tests to identify concrete, deviating groups. If one wishes to do that just for significant variables one can set the option mwAkw to the corresponding significance level. If mwAkw is set to TRUE (or respective 1) the Mann-Whitney tests are performed for all variables.

There is also a function to choose the used calculation algorithm, options here are "Rsubmat", "Rnaive", "Csubmat", "Cnaive". The purpose is just for validation.

Value

A matrix or vector of p-values of the underlying hypothesis test(s). In case of output="full" we give a list, and each list item contains the htest object for the column-wise performed test.

Author(s)

Daniel Fischer

References

Fischer, D., Oja, H., Schleutker, J., Sen, P.K., Wahlfors, T. (2013): Generalized Mann-Whitney Type Tests for Microarray Experiments, Scandinavian Journal of Statistic, doi: 10.1111/sjos.12055.

Daniel Fischer, Hannu Oja (2015). Mann-Whitney Type Tests for Microarray Experiments: The R Package gMWT. Journal of Statistical Software, 65(9), 1-19. URL http://www.jstatsoft.org/v65/i01/.

Westfall, P.H. and Young, S.S. (1993): Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment. Wiley, New York.

Examples

```
X <- c(sample(15))
X <- c(X,101,102,103)
g <- c(1,1,1,2,2,2,2,3,3,3,4,4,4,4,4,5,5,5)
cluster=c(rep(c(1,2),9))

gmw(X,g,test="kw",type="external")
gmw(X,g,test="kw",type="permutation")
gmw(X,g,test="kw",type="permutation",cluster=cluster)

gmw(X,g,test="jt",type="permutation")
```

`plot.estPI` *Plot an estPI Object*

Description

Plots an `estPI` object.

Usage

```
 ## S3 method for class 'estPI'
plot(x,col="black",highlight,hlCol="red",pch=20,zoom=FALSE,...)
```

Arguments

`x`	Object of class `estPI`.
`col`	Vector of colors of the scatterplot.
`highlight`	Vector with positions, which shall be marked in special color.
`hlCol`	Color of highlighted spots.
`pch`	Dot type of the plot.
`zoom`	Logical, shall the plots be zoomed to interesting areas?
`...`	Additional parameters.

Details

This function plots the probabilistic indices from an `estPI` object. Additional graphical parameters are `col`, that specifies the colors of the points used in the plot. Furthermore there is the option to highlight the probabilistic indices for specified variables. Those are defined with the `highlight` option and the `hlCol` color. One possible application of this is to highlight only the significant test results, or e.g. in the genetic context only those genes, which are located on a certain chromosome.

The boolean option `zoom` is meant for plotting the graph in a different scale and it adjusts the limits of the x and y axes to the minimum and maximum of the corresponding probabilistic index.

Value

A plot for `estPI` object

Author(s)

Daniel Fischer

Examples

```
X <- c(sample(15))
g <- c(1,1,1,2,2,2,2,3,3,3,4,4,4,4,4)
res <- estPI(X,g,type="single")
plot(res)

X <- matrix(c(rnorm(5000,1.5,2),rnorm(6000,2,2),rnorm(4000,3.5,1),rnorm(3000,2.5,1)),
         byrow=TRUE, ncol=100)
colnames(X) <- letters[1:100]
g <- c(rep(1,50),rep(2,60),rep(3,40),rep(4,30))

res <- estPI(X,g,type="single")
plot(res)
```

`plot.re` *Plot an re Object*

Description

Plots an `re` object returned from the function `getSigTests`.

Usage

```
 ## S3 method for class 're'
plot(x,...)
```

Arguments

`x`	Object of class `re`.
`...`	Additional parameters.

Details

This function plots a rejection plot for an existing `re` object.

Value

A plot from a `re` object

Author(s)

Daniel Fischer

See Also

`getSigTests`

Examples

```
X <- matrix(c(rnorm(500,2,1),rnorm(600,2,1),rnorm(400,2.2,1)),byrow=TRUE, ncol=10)
colnames(X) <- letters[1:10]
g <- c(rep(1,50),rep(2,60),rep(3,40))

test <- gmw(X,g,test="kw",type="external")

re1 <- getSigTests(test)

plot(re1)
```

---

plotPI *Creates Scatterplot Matrices for Probabilistic Indices.*

Description

This function creates scatterplot matrices for the different probabilistic indices (PI) P_t , $P_{tt'}$ and $P_{tt't''}$ computed for different variables.

Usage

```
plotPI(X,g,type="pair",goi=NULL,mc=1,alg="Cnaive",col="black",
         highlight=NULL,hlCol="red",pch=20,zoom=FALSE,order=NULL,...)
```

Arguments

X	Matrix or vector with observations. Each column is a variable, each row an individual.
g	Vector of group labels for observations in X. Has to be the same length as X has observations.
type	Type of probabilistic index, see details.
goi	Groups of Interest, see details.
mc	Set the amount of cores to use for parallel calculation (only available for Linux).
order	Boolean, calculate PI only for natural order or for all combinations.
alg	Internal function, which implementation should be used to calculate the PI.
col	Vector of colors of the scatterplot.
highlight	Vector with positions, which are marked in special color.
hlCol	Color of highlighted spots.
pch	Dot type of the plot.
zoom	Logical, shall the plots be zoomed to interesting areas?
...	Additional plotting arguments.

Details

This function creates the scatterplot matrices for the PI, in case that they haven't been calculated previoulsy. This means that all arguments of the `estPI` are valid here, since this function is called first and the results will then be passed to the plot function of the `estPI` object.

See also `plot.estPI` for further details on the specific plot parameters.

Value

A plot of probalistic indices

Author(s)

Daniel Fischer

References

Fischer, D., Oja, H., Schleutker, J., Sen, P.K., Wahlfors, T. (2013): Generalized Mann-Whitney Type Tests for Microarray Experiments, Scandinavian Journal of Statistic, to appear.

Fischer, D., Oja, H. (2013): Mann-Whitney Type Tests for Microarray Experiments: The R Package gMWT, submitted article.

See Also

`estPI`, `plot.estPI`

Examples

```
X <- c(sample(15))
g <- c(1,1,1,2,2,2,2,3,3,3,4,4,4,4,4)
estPI(X,g,type="single")

X <- matrix(c(rnorm(5000,1.5,2),rnorm(6000,2,2),rnorm(4000,3.5,1)),byrow=TRUE, ncol=10)
colnames(X) <- letters[1:10]
g <- c(rep(1,500),rep(2,600),rep(3,400))

plotPI(X,g,type="single",mc=1)
```

---

print.estPI *Print an estPI Object*

Description

Prints an `estPI` object.

Usage

```
 ## S3 method for class 'estPI'
print(x,...)
```

Arguments

x	Object of class estPI.
...	Additional parameters.

Details

The print function displays the matrix of probabilistic indices.

Value

Content of an estPI object

Author(s)

Daniel Fischer

Examples

```
X <- c(sample(15))
g <- c(1,1,1,2,2,2,2,3,3,3,4,4,4,4,4)
res <- estPI(X,g,type="single")
print(res)
```

---

print.re *Print an re Object*

Description

Prints an re object.

Usage

```
 ## S3 method for class 're'
print(x,...)
```

Arguments

x	Object of class re.
...	Additional parameters.

Details

The print function displays in a list the significant variables for a chosen alpha and the optimal alpha in terms of the best ratio between expected and observed significant tests. For more details see also rejectionPlot.

Value

Content of an re object

Author(s)

Daniel Fischer

See Also

rejectionPlot

Examples

```
X <- matrix(c(rnorm(500,2,1),rnorm(600,2,1),rnorm(400,2.2,1)),byrow=TRUE, ncol=10)
colnames(X) <- letters[1:10]
g <- c(rep(1,50),rep(2,60),rep(3,40))

test <- gmw(X,g,test="kw",type="external")

re1 <- getSigTests(test)

re1
```

---

rejectionPlot *Creates a Rejection Plot.*

Description

This function plots the observed and expected numbers of rejected tests and their ratio as a function of alpha.

Usage

```
rejectionPlot(X,lCol="red",xlim=NULL,crit=NULL,rejLine = NULL,
              alpha = 0.01, rej.lty = c("dotted"), rej.col = "black",
incLegend=NULL)
```

Arguments

X	Matrix or vector of p-values, as obtained from gmw function.
lCol	Vector of of line colors
xlim	An interval for alpha values (for the plotting). Default is [0,1].
crit	Shall the distance or the ratio be plotted, see details.
rejLine	Type of rejection line. Options are "bh","bonferroni" and "simes".
alpha	Alpha for the multiple testing adjusted p-value- determines the slope of the rejection line.
rej.lty	Line type of the rejection line.
rej.col	Colour of the rejection line.
incLegend	Position of a possible legend.

Details

The matrix X contains p-values as given from the gmw function. The rows correspond to different tests, and the columns to different variables.

The function plots, for each test separately, the observed and expected numbers of rejected tests and their ratio (crit="ratio") or distance (crit="distance") as a function of alpha.

Value

A rejection plot

Author(s)

Daniel Fischer

References

Fischer, D., Oja, H., Schleutker, J., Sen, P.K., Wahlfors, T. (2013): Generalized Mann-Whitney Type Tests for Microarray Experiments, Scandinavian Journal of Statistic, to appear.

Fischer, D., Oja, H. (2013): Mann-Whitney Type Tests for Microarray Experiments: The R Package gMWT, submitted article.

See Also

gmw

Examples

```
X <- matrix(c(rnorm(50,2,1),rnorm(60,2,1),rnorm(40,2.5,1)),byrow=TRUE, ncol=10)
colnames(X) <- letters[1:10]
g <- c(rep(1,5),rep(2,6),rep(3,4))


test1 <- gmw(X,g,test="mw",type="external",alternative="greater",goi=c(2,3))
```

```
test2 <- gmw(X,g,test="kw",type="external")
test3 <- gmw(X,g,test="jt",type="external",alternative="greater")

tests <- rbind(test1$p.values,test2$p.values,test3$p.values)

rejectionPlot(tests,lCol=c("green","red","blue"))
```

---

summary.re *Summarize an re Object*

Description

Summarizes and prints an re object in an informative way.

Usage

```
 ## S3 method for class 're'
summary(object, ...)
```

Arguments

object	Object of class re.
...	Additional parameters.

Details

Summary for a re object.

Value

Summary for an re object

Author(s)

Daniel Fischer

Examples

```
X <- matrix(c(rnorm(500,2,1),rnorm(600,2,1),rnorm(400,2.2,1)),byrow=TRUE, ncol=10)
colnames(X) <- letters[1:10]
g <- c(rep(1,50),rep(2,60),rep(3,40))

test <- gmw(X,g,test="kw",type="external")

re1 <- getSigTests(test)

summary(re1)
```

Index

∗ **hplot**
    plotPI, 13
∗ **methods**
    createGroups, 3
    estPI, 4
    getSigTests, 6
    gmw, 8
    plot.estPI, 11
    plot.re, 12
    plotPI, 13
    print.estPI, 14
    print.re, 15
    rejectionPlot, 16
    summary.re, 18
∗ **package**
    gMWT-package, 2
∗ **plot**
    plot.estPI, 11
    plot.re, 12
∗ **print**
    print.estPI, 14
    print.re, 15
    summary.re, 18
∗
    gmw, 8

createGroups, 3, *4, 5*

estPI, 4, *14*

getSigTests, 6, *12*
gmw, *7*, 8, *17*
gMWT *(*gMWT-package*)*, 2
gMWT-package, 2

plot,estPI-method *(*plot.estPI*)*, 11
plot,re-method *(*plot.re*)*, 12
plot-method *(*plot.estPI*)*, 11
plot.estPI, 11, *14*
plot.re, 12
plotPI, 13
print,estPI-method *(*print.estPI*)*, 14
print,re-method *(*print.re*)*, 15
print-method *(*print.estPI*)*, 14
print.estPI, 14
print.re, 15

rejectionPlot, *6, 7*, 16, *16*

summary,re-method *(*summary.re*)*, 18
summary.re, 18