
golub

Gene expression dataset from Golub et al. (1999)

Description

Gene expression data (3051 genes and 38 tumor mRNA samples) from the leukemia microarray study of Golub et al. (1999). Pre-processing was done as described in Dudoit et al. (2002). The R code for pre-processing is available in the file [../doc/golub.R](#).

Usage

```
data(golub)
```

Value

<code>golub</code>	matrix of gene expression levels for the 38 tumor mRNA samples, rows correspond to genes (3051 genes) and columns to mRNA samples.
<code>golub.cl</code>	numeric vector indicating the tumor class, 27 acute lymphoblastic leukemia (ALL) cases (code 0) and 11 acute myeloid leukemia (AML) cases (code 1).
<code>golub.gnames</code>	a matrix containing the names of the 3051 genes for the expression matrix <code>golub</code> . The three columns correspond to the gene <code>index</code> , <code>ID</code> , and <code>Name</code> , respectively.

Source

Golub et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science*, Vol. 286:531-537.

<http://www-genome.wi.mit.edu/MPR/> .

References

S. Dudoit, J. Fridlyand, and T. P. Speed (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, Vol. 97, No. 457, p. 77–87.

multtest-internal

Internal multtest functions and variables

Description

Internal multtest functions and variables

Usage

```
.mt.BLIM
.mt.RandSeed
.mt.naNUM
mt.number2na(x,na)
mt.na2number(x,na)
mt.getMaxB(classlabel,test,B, verbose)
mt.transformL(classlabel,test)
mt.transformV(V,classlabel,test,na,nonpara)
mt.transformX(X,classlabel,test,na,nonpara)
mt.checkothers(side="abs",fixed.seed.sampling="y", B=10000,
na=.mt.naNUM, nonpara="n")
mt.checkX(X,classlabel,test)
mt.checkV(V,classlabel,test)
mt.checkclasslabel(classlabel,test)
mt.niceres<-function(res,X,index)
```

Details

These are not to be called directly by the user.

<code>mt.maxT</code>	<i>Step-down maxT and minP multiple testing procedures</i>
----------------------	--

Description

These functions compute permutation adjusted p -values for step-down multiple testing procedures described in Westfall & Young (1993).

Usage

```
mt.maxT(X,classlabel,test="t",side="abs",fixed.seed.sampling="y",B=10000,na=.mt.naNUM,nonpara=
mt.minP(X,classlabel,test="t",side="abs",fixed.seed.sampling="y",B=10000,na=.mt.naNUM,nonpara=
```

Arguments

X	A data frame or matrix, with m rows corresponding to variables (hypotheses) and n columns to observations. In the case of gene expression data, rows correspond to genes and columns to mRNA samples. The data can be read using <code>read.table</code> .
classlabel	A vector of integers corresponding to observation (column) class labels. For k classes, the labels must be integers between 0 and $k - 1$. For the <code>blockf</code> test option, observations may be divided into n/k blocks of k observations each. The observations are ordered by block, and within each block, they are labeled using the integers 0 to $k - 1$.

test	<p>A character string specifying the statistic to be used to test the null hypothesis of no association between the variables and the class labels.</p> <p>If test="t", the tests are based on two-sample Welch t-statistics (unequal variances).</p> <p>If test="t.equalvar", the tests are based on two-sample t-statistics with equal variance for the two samples. The square of the t-statistic is equal to an F-statistic for $k = 2$.</p> <p>If test="wilcoxon", the tests are based on standardized rank sum Wilcoxon statistics.</p> <p>If test="f", the tests are based on F-statistics.</p> <p>If test="pairt", the tests are based on paired t-statistics. The square of the paired t-statistic is equal to a block F-statistic for $k = 2$.</p> <p>If test="blockf", the tests are based on F-statistics which adjust for block differences (cf. two-way analysis of variance).</p>
side	<p>A character string specifying the type of rejection region.</p> <p>If side="abs", two-tailed tests, the null hypothesis is rejected for large absolute values of the test statistic.</p> <p>If side="upper", one-tailed tests, the null hypothesis is rejected for large values of the test statistic.</p> <p>If side="lower", one-tailed tests, the null hypothesis is rejected for small values of the test statistic.</p>
fixed.seed.sampling	<p>If fixed.seed.sampling="y", a fixed seed sampling procedure is used, which may double the computing time, but will not use extra memory to store the permutations. If fixed.seed.sampling="n", permutations will be stored in memory. For the blockf test, the option n was not implemented as it requires too much memory.</p>
B	<p>The number of permutations. For a complete enumeration, B should be 0 (zero) or any number not less than the total number of permutations.</p>
na	<p>Code for missing values (the default is .mt.naNUM=--93074815.62). Entries with missing values will be ignored in the computation, i.e., test statistics will be based on a smaller sample size. This feature has not yet fully implemented.</p>
nonpara	<p>If nonpara="y", nonparametric test statistics are computed based on ranked data.</p> <p>If nonpara="n", the original data are used.</p>

Details

These functions compute permutation adjusted p -values for the step-down maxT and minP multiple testing procedures, which provide strong control of the family-wise Type I error rate (FWER). The adjusted p -values for the minP procedure are defined in equation (2.10) p. 66 of Westfall & Young (1993), and the maxT procedure is discussed p. 50 and 114. The permutation algorithms for estimating the adjusted p -values are given in Ge et al. (In preparation). The procedures are for the simultaneous test of m null hypotheses, namely, the null hypotheses of no association between the m variables corresponding to the rows of the data frame **X** and the class labels **classlabel**. For gene expression data, the null hypotheses correspond to no differential gene expression across mRNA samples.

Value

A data frame with components

<code>index</code>	Vector of row indices, between 1 and <code>nrow(X)</code> , where rows are sorted first according to their adjusted p -values, next their unadjusted p -values, and finally their test statistics.
<code>teststat</code>	Vector of test statistics, ordered according to <code>index</code> . To get the test statistics in the original data order, use <code>teststat[order(index)]</code> .
<code>rawp</code>	Vector of raw (unadjusted) p -values, ordered according to <code>index</code> .
<code>adjp</code>	Vector of adjusted p -values, ordered according to <code>index</code> .
<code>plover</code>	For <code>mt.minP</code> function only, vector of "adjusted p -values", where ties in the permutation distribution of the successive minima of raw p -values with the observed p -values are counted only once. Note that procedures based on <code>plover</code> do not control the FWER. Comparison of <code>plover</code> and <code>adjp</code> gives an idea of the discreteness of the permutation distribution. Values in <code>plover</code> are ordered according to <code>index</code> .

Author(s)

Yongchao Ge, yongchao.ge@mssm.edu,
Sandrine Dudoit, <http://www.stat.berkeley.edu/~sandrine>.

References

S. Dudoit, J. P. Shaffer, and J. C. Boldrick (Submitted). Multiple hypothesis testing in microarray experiments.

Y. Ge, S. Dudoit, and T. P. Speed. Resampling-based multiple testing for microarray data hypothesis, Technical Report #633 of UCB Stat. <http://www.stat.berkeley.edu/~gyc>

P. H. Westfall and S. S. Young (1993). *Resampling-based multiple testing: Examples and methods for p-value adjustment*. John Wiley & Sons.

See Also

`mt.plot`, `mt.rawp2adjp`, `mt.reject`, `mt.sample.teststat`, `mt.teststat`, `golub`.

Examples

```
# Gene expression data from Golub et al. (1999)
# To reduce computation time and for illustrative purposes, we consider only
# the first 100 genes and use the default of B=10,000 permutations.
# In general, one would need a much larger number of permutations
# for microarray data.

data(golub)
smallgd<-golub[1:100,]
classlabel<-golub.cl
```

```

# Permutation unadjusted p-values and adjusted p-values
# for maxT and minP procedures with Welch t-statistics
resT<-mt.maxT(smallgd,classlabel)
resP<-mt.minP(smallgd,classlabel)
rawp<-resT$rawp[order(resT$index)]
teststat<-resT$teststat[order(resT$index)]

# Plot results and compare to Bonferroni procedure
bonf<-mt.rawp2adjp(rawp, proc=c("Bonferroni"))
allp<-cbind(rawp, bonf$adjp[order(bonf$index),2], resT$adjp[order(resT$index)],resP$adjp[order(resP$index),2])

mt.plot(allp, teststat, plottype="rvsa", proc=c("rawp","Bonferroni","maxT","minP"),leg=c(0.7,50),lty=1,col="black")
mt.plot(allp, teststat, plottype="pvst", proc=c("rawp","Bonferroni","maxT","minP"),leg=c(60,0.2),lty=1,col="black")
mt.plot(allp, teststat, plottype="pvst", proc=c("rawp","Bonferroni","maxT","minP"),leg=c(-6,0.6),pch=16,col="black")

# Permutation adjusted p-values for minP procedure with F-statistics (like equal variance t-statistics)
mt.minP(smallgd,classlabel,test="f",fixed.seed.sampling="n")

# Note that the test statistics used in the examples below are not appropriate
# for the Golub et al. data. The sole purpose of these examples is to
# demonstrate the use of the mt.maxT and mt.minP functions.

# Permutation adjusted p-values for maxT procedure with paired t-statistics
classlabel<-rep(c(0,1),19)
mt.maxT(smallgd,classlabel,test="pairt")

# Permutation adjusted p-values for maxT procedure with block F-statistics
classlabel<-rep(0:18,2)
mt.maxT(smallgd,classlabel,test="blockf",side="upper")

```

mt.plot

Plotting results from multiple testing procedures

Description

This function produces a number of graphical summaries for the results of multiple testing procedures and their corresponding adjusted p -values.

Usage

```
mt.plot(adjp, teststat, plottype="rvsa", logscale=FALSE, alpha=seq(0, 1, length = 100), proc, 1
```

Arguments

adjp A matrix of adjusted p -values, with rows corresponding to hypotheses (genes) and columns to multiple testing procedures. This matrix could be obtained from the functions `mt.maxT`, `mt.minP`, or `mt.rawp2adjp`.

<code>teststat</code>	A vector of test statistics for each of the hypotheses. This vector could be obtained from the functions <code>mt.teststat</code> , <code>mt.maxT</code> , or <code>mt.minP</code> .
<code>plottype</code>	A character string specifying the type of graphical summary for the results of the multiple testing procedures. If <code>plottype="rvsa"</code> , the number of rejected hypotheses is plotted against the nominal Type I error rate for each of the procedures given in <code>proc</code> . If <code>plottype="pvst"</code> , the ordered adjusted p -values are plotted for each of the procedures given in <code>proc</code> . This can be viewed as a plot of the Type I error rate against the number of rejected hypotheses. If <code>plottype="pvst"</code> , the adjusted p -values are plotted against the test statistics for each of the procedures given in <code>proc</code> . If <code>plottype="pvsr"</code> , the adjusted p -values are plotted for each of the procedures given in <code>proc</code> using the original data order.
<code>logscale</code>	A logical variable for the <code>pvst</code> and <code>pvsr</code> plots. If <code>logscale</code> is <code>TRUE</code> , the negative decimal logarithms of the adjusted p -values are plotted against the test statistics or gene indices. If <code>logscale</code> is <code>FALSE</code> , the adjusted p -values are plotted against the test statistics or gene indices.
<code>alpha</code>	A vector of nominal Type I error rates for the <code>rvsa</code> plot.
<code>proc</code>	A vector of character strings containing the names of the multiple testing procedures, to be used in the legend.
<code>...</code>	Graphical parameters such as <code>col</code> , <code>lty</code> , <code>pch</code> , and <code>lwd</code> may also be supplied as arguments to the function (see <code>par</code>).
<code>leg</code>	A vector of coordinates for the legend.

Author(s)

Sandrine Dudoit, <http://www.stat.berkeley.edu/~sandrine>,
Yongchao Ge, (yongchao.ge@mssm.edu).

References

S. Dudoit, J. P. Shaffer, and J. C. Boldrick (Submitted). Multiple hypothesis testing in microarray experiments.

Y. Ge, S. Dudoit, and T. P. Speed. Resampling-based multiple testing for microarray data hypothesis, Technical Report #633 of UCB Stat. <http://www.stat.berkeley.edu/~gyc>

See Also

`mt.maxT`, `mt.minP`, `mt.rawp2adjp`, `mt.reject`, `mt.teststat`, `golub`.

Examples

```
# Gene expression data from Golub et al. (1999)
# To reduce computation time and for illustrative purposes, we consider only
# the first 100 genes and use the default of B=10,000 permutations.
# In general, one would need a much larger number of permutations
```

```

# for microarray data.

data(golub)
smallgd<-golub[1:100,]
classlabel<-golub.cl

# Permutation unadjusted p-values and adjusted p-values for maxT procedure
res1<-mt.maxT(smallgd,classlabel)
rawp<-res1$rawp[order(res1$index)]
teststat<-res1$teststat[order(res1$index)]

# Permutation adjusted p-values for simple multiple testing procedures
procs<-c("Bonferroni","Holm","Hochberg","SidakSS","SidakSD","BH","BY")
res2<-mt.rawp2adjp(rawp,procs)

# Plot results from all multiple testing procedures
allp<-cbind(res2$adjp[order(res2$index),],res1$adjp[order(res1$index)])
dimnames(allp)[[2]][9]<-"maxT"
procs<-dimnames(allp)[[2]]
procs[7:9]<-c("maxT","BH","BY")
allp<-allp[,procs]

cols<-c(1:4,"orange","brown","purple",5:6)
ltypes<-c(3,rep(1,6),rep(2,2))

# Ordered adjusted p-values
mt.plot(allp,teststat,plottype="pvsr",proc=procs,leg=c(80,0.4),lty=ltypes,col=cols,lwd=2)

# Adjusted p-values in original data order
mt.plot(allp,teststat,plottype="pvsi",proc=procs,leg=c(80,0.4),lty=ltypes,col=cols,lwd=2)

# Number of rejected hypotheses vs. level of the test
mt.plot(allp,teststat,plottype="rvsa",proc=procs,leg=c(0.05,100),lty=ltypes,col=cols,lwd=2)

# Adjusted p-values vs. test statistics
mt.plot(allp,teststat,plottype="pvst",logscale=TRUE,proc=procs,leg=c(0,4),pch=ltypes,col=cols)

```

`mt.rawp2adjp`

Adjusted p-values for simple multiple testing procedures

Description

This function computes adjusted p -values for simple multiple testing procedures from a vector of raw (unadjusted) p -values. The procedures include the Bonferroni, Holm (1979), Hochberg (1988), and Sidak procedures for strong control of the family-wise Type I error rate (FWER), and the Benjamini & Hochberg (1995) and Benjamini & Yekutieli (2001) procedures for (strong) control of the false discovery rate (FDR).

Usage

```
mt.rawp2adjp(rawp, proc=c("Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD", "BH", "BY"))
```

Arguments

- rawp** A vector of raw (unadjusted) p -values for each hypothesis under consideration. These could be nominal p -values, for example, from t -tables, or permutation p -values as given in `mt.maxT` and `mt.minP`. If the `mt.maxT` or `mt.minP` functions are used, raw p -values should be given in the original data order, `rawp[order(index)]`.
- proc** A vector of character strings containing the names of the multiple testing procedures for which adjusted p -values are to be computed. This vector should include any of the following: "Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD", "BH", "BY".

Details

Adjusted p -values are computed for simple FWER and FDR controlling procedures based on a vector of raw (unadjusted) p -values.

Bonferroni Bonferroni single-step adjusted p -values for strong control of the FWER.

Holm Holm (1979) step-down adjusted p -values for strong control of the FWER.

Hochberg Hochberg (1988) step-up adjusted p -values for strong control of the FWER (for raw (unadjusted) p -values satisfying the Simes inequality).

SidakSS Sidak single-step adjusted p -values for strong control of the FWER (for positive orthant dependent test statistics).

SidakSD Sidak step-down adjusted p -values for strong control of the FWER (for positive orthant dependent test statistics).

BH adjusted p -values for the Benjamini & Hochberg (1995) step-up FDR controlling procedure (independent and positive regression dependent test statistics).

BY adjusted p -values for the Benjamini & Yekutieli (2001) step-up FDR controlling procedure (general dependency structures).

Value

A list with components

- adjp** A matrix of adjusted p -values, with rows corresponding to hypotheses and columns to multiple testing procedures. Hypotheses are sorted in increasing order of their raw (unadjusted) p -values.
- index** A vector of row indices, between 1 and `length(rawp)`, where rows are sorted according to their raw (unadjusted) p -values. To obtain the adjusted p -values in the original data order, use `adjp[order(index),]`.

Author(s)

Sandrine Dudoit, <http://www.stat.berkeley.edu/~sandrine>,
Yongchao Ge, <yongchao.ge@mssm.edu>.

References

- Y. Benjamini and Y. Hochberg (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc. B*. Vol. 57: 289-300.
- Y. Benjamini and D. Yekutieli (2001). The control of the false discovery rate in multiple hypothesis testing under dependency. *Annals of Statistics*. Accepted.
- S. Dudoit, J. P. Shaffer, and J. C. Boldrick (Submitted). Multiple hypothesis testing in microarray experiments.
- Y. Ge, S. Dudoit, and T. P. Speed. Resampling-based multiple testing for microarray data hypothesis, Technical Report #633 of UCB Stat. <http://www.stat.berkeley.edu/~gyc>
- Y. Hochberg (1988). A sharper Bonferroni procedure for multiple tests of significance, *Biometrika*. Vol. 75: 800-802.
- S. Holm (1979). A simple sequentially rejective multiple test procedure. *Scand. J. Statist.*. Vol. 6: 65-70.

See Also

[mt.maxT](#), [mt.minP](#), [mt.plot](#), [mt.reject](#), [golub](#).

Examples

```
# Gene expression data from Golub et al. (1999)
# To reduce computation time and for illustrative purposes, we consider only
# the first 100 genes and use the default of B=10,000 permutations.
# In general, one would need a much larger number of permutations
# for microarray data.

data(golub)
smallgd<-golub[1:100,]
classlabel<-golub.cl

# Permutation unadjusted p-values and adjusted p-values for maxT procedure
res1<-mt.maxT(smallgd,classlabel)
rawp<-res1$rawp[order(res1$index)]

# Permutation adjusted p-values for simple multiple testing procedures
procs<-c("Bonferroni","Holm","Hochberg","SidakSS","SidakSD","BH","BY")
res2<-mt.rawp2adjp(rawp,procs)
```

`mt.reject`

Identity and number of rejected hypotheses

Description

This function returns the identity and number of rejected hypotheses for several multiple testing procedures and different nominal Type I error rates.

Usage

```
mt.reject(adjp, alpha)
```

Arguments

<code>adjp</code>	A matrix of adjusted p -values, with rows corresponding to hypotheses and columns to multiple testing procedures. This matrix could be obtained from the function <code>mt.rawp2adjp</code> .
<code>alpha</code>	A vector of nominal Type I error rates.

Value

A list with components

<code>r</code>	A matrix containing the number of rejected hypotheses for several multiple testing procedures and different nominal Type I error rates. Rows correspond to Type I error rates and columns to multiple testing procedures.
<code>which</code>	A matrix of indicators for the rejection of individual hypotheses by different multiple testing procedures for a nominal Type I error rate <code>alpha[1]</code> . Rows correspond to hypotheses and columns to multiple testing procedures.

Author(s)

Sandrine Dudoit, <http://www.stat.berkeley.edu/~sandrine>,
Yongchao Ge, yongchao.ge@mssm.edu.

See Also

`mt.maxT`, `mt.minP`, `mt.rawp2adjp`, `golub`.

Examples

```
# Gene expression data from Golub et al. (1999)
# To reduce computation time and for illustrative purposes, we consider only
# the first 100 genes and use the default of B=10,000 permutations.
# In general, one would need a much larger number of permutations
# for microarray data.
```

```

data(golub)
smallgd<-golub[1:100,]
classlabel<-golub.cl

# Permutation unadjusted p-values and adjusted p-values for maxT procedure
res<-mt.maxT(smallgd,classlabel)
mt.reject(cbind(res$rawp,res$adjp),seq(0,1,0.1))$r

```

<code>mt.sample.teststat</code>	<i>Permutation distribution of test statistics and raw (unadjusted) p-values</i>
---------------------------------	--

Description

These functions provide tools to investigate the permutation distribution of test statistics, raw (unadjusted) p -values, and class labels.

Usage

```

mt.sample.teststat(V,classlabel,test="t",fixed.seed.sampling="y",B=10000,na=.mt.naNUM,nonpara=
mt.sample.rawp(V,classlabel,test="t",side="abs",fixed.seed.sampling="y",B=10000,na=.mt.naNUM,n
mt.sample.label(classlabel,test="t",fixed.seed.sampling="y",B=10000)

```

Arguments

V	A numeric vector containing the data for one of the variables (genes).
classlabel	A vector of integers corresponding to observation (column) class labels. For k classes, the labels must be integers between 0 and $k - 1$. For the blockf test option, observations may be divided into n/k blocks of k observations each. The observations are ordered by block, and within each block, they are labeled using the integers 0 to $k - 1$.
test	<p>A character string specifying the statistic to be used to test the null hypothesis of no association between the variables and the class labels.</p> <p>If test="t", the tests are based on two-sample Welch t-statistics (unequal variances).</p> <p>If test="t.equalvar", the tests are based on two-sample t-statistics with equal variance for the two samples. The square of the t-statistic is equal to an F-statistic for $k = 2$.</p> <p>If test="wilcoxon", the tests are based on standardized rank sum Wilcoxon statistics.</p> <p>If test="f", the tests are based on F-statistics.</p> <p>If test="pairt", the tests are based on paired t-statistics. The square of the paired t-statistic is equal to a block F-statistic for $k = 2$.</p> <p>If test="blockf", the tests are based on F-statistics which adjust for block differences (cf. two-way analysis of variance).</p>

side	<p>A character string specifying the type of rejection region.</p> <p>If side="abs", two-tailed tests, the null hypothesis is rejected for large absolute values of the test statistic.</p> <p>If side="upper", one-tailed tests, the null hypothesis is rejected for large values of the test statistic.</p> <p>If side="lower", one-tailed tests, the null hypothesis is rejected for small values of the test statistic.</p>
fixed.seed.sampling	<p>If fixed.seed.sampling="y", a fixed seed sampling procedure is used, which may double the computing time, but will not use extra memory to store the permutations. If fixed.seed.sampling="n", permutations will be stored in memory. For the blockf test, the option n was not implemented as it requires too much memory.</p>
B	<p>The number of permutations. For a complete enumeration, B should be 0 (zero) or any number not less than the total number of permutations.</p>
na	<p>Code for missing values (the default is .mt.naNUM=--93074815.62). Entries with missing values will be ignored in the computation, i.e., test statistics will be based on a smaller sample size. This feature has not yet fully implemented.</p>
nonpara	<p>If nonpara="y", nonparametric test statistics are computed based on ranked data.</p> <p>If nonpara="n", the original data are used.</p>

Value

For **mt.sample.teststat**, a vector containing **B** permutation test statistics.

For **mt.sample.rawp**, a vector containing **B** permutation unadjusted *p*-values.

For **mt.sample.label**, a matrix containing **B** sets of permuted class labels. Each row corresponds to one permutation.

Author(s)

Yongchao Ge, <yongchao.ge@mssm.edu>,
 Sandrine Dudoit, <http://www.stat.berkeley.edu/~sandrine>.

See Also

mt.maxT, **mt.minP**, **golub**.

Examples

```
# Gene expression data from Golub et al. (1999)
data(golub)

mt.sample.label(golub.cl,B=10)
```

```

permt<-mt.sample.teststat(golub[1,],golub.c1,B=1000)
qqnorm(permt)
qqline(permt)

permt<-mt.sample.teststat(golub[50,],golub.c1,B=1000)
qqnorm(permt)
qqline(permt)

permp<-mt.sample.rawp(golub[1,],golub.c1,B=1000)
hist(permp)

```

`mt.teststat`

Computing test statistics for each row of a data frame

Description

These functions provide a convenient way to compute test statistics, e.g., two-sample Welch t-statistics, Wilcoxon statistics, F-statistics, paired t-statistics, block F-statistics, for each row of a data frame.

Usage

```

mt.teststat(X,classlabel,test="t",na=.mt.naNUM,nonpara="n")
mt.teststat.num.denum(X,classlabel,test="t",na=.mt.naNUM,nonpara="n")

```

Arguments

X	A data frame or matrix, with m rows corresponding to variables (hypotheses) and n columns to observations. In the case of gene expression data, rows correspond to genes and columns to mRNA samples. The data can be read using <code>read.table</code> .
classlabel	A vector of integers corresponding to observation (column) class labels. For k classes, the labels must be integers between 0 and $k - 1$. For the <code>blockf</code> test option, observations may be divided into n/k blocks of k observations each. The observations are ordered by block, and within each block, they are labeled using the integers 0 to $k - 1$.
test	<p>A character string specifying the statistic to be used to test the null hypothesis of no association between the variables and the class labels.</p> <p>If <code>test="t"</code>, the tests are based on two-sample Welch t-statistics (unequal variances).</p> <p>If <code>test="t.equalvar"</code>, the tests are based on two-sample t-statistics with equal variance for the two samples. The square of the t-statistic is equal to an F-statistic for $k = 2$.</p> <p>If <code>test="wilcoxon"</code>, the tests are based on standardized rank sum Wilcoxon statistics.</p> <p>If <code>test="f"</code>, the tests are based on F-statistics.</p> <p>If <code>test="pairt"</code>, the tests are based on paired t-statistics. The square</p>

of the paired t-statistic is equal to a block F-statistic for $k = 2$.
If `test="blockf"`, the tests are based on F-statistics which adjust for block differences (cf. two-way analysis of variance).

`na` Code for missing values (the default is `.mt.naNUM=--93074815.62`). Entries with missing values will be ignored in the computation, i.e., test statistics will be based on a smaller sample size. This feature has not yet fully implemented.

`nonpara` If `nonpara="y"`, nonparametric test statistics are computed based on ranked data.
If `nonpara="n"`, the original data are used.

Value

For `mt.teststat`, a vector of test statistics for each row (gene).

For `mt.teststat.num.denum`, a data frame with

`teststat.num` the numerator of the test statistics for each row, depending on the specific `test` option.

`teststat.denum` the denominator of the test statistics for each row, depending on the specific `test` option.

Author(s)

Yongchao Ge, yongchao.ge@mssm.edu,
Sandrine Dudoit, <http://www.stat.berkeley.edu/~sandrine>.

See Also

`mt.maxT`, `mt.minP`, `golub`.

Examples

```
# Gene expression data from Golub et al. (1999)
data(golub)

teststat<-mt.teststat(golub,golub.cl)
qqnorm(teststat)
qqline(teststat)

tmp<-mt.teststat.num.denum(golub,golub.cl,test="t")
num<-tmp$teststat.num
denum<-tmp$teststat.denum
plot(sqrt(denum),num)

tmp<-mt.teststat.num.denum(golub,golub.cl,test="f")
```