Package 'siggenes'

October 13, 2025

Version 1.82.0

Date 2021-08-02

Title Multiple Testing using SAM and Efron's Empirical Bayes Approaches

Author Holger Schwender

Maintainer Holger Schwender <holger.schw@gmx.de>

Depends Biobase, multtest, splines, methods

Imports stats4, grDevices, graphics, stats, scrime (>= 1.2.5)

Suggests affy, annotate, genefilter, KernSmooth

Description Identification of differentially expressed genes and estimation of the False Discovery Rate (FDR) using both the Significance Analysis of Microarrays (SAM) and the Empirical Bayes Analyses of Microarrays (EBAM).

License LGPL (>= 2)

biocViews MultipleComparison, Microarray, GeneExpression, SNP, ExonArray, DifferentialExpression

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

git_branch RELEASE_3_21

git_last_commit bdda32e

git_last_commit_date 2025-04-15

Repository Bioconductor 3.21

Date/Publication 2025-10-12

Contents

`chisq.ebam` *EBAM Analysis for Categorical Data*

Description

Generates the required statistics for an Empirical Bayes Analysis of Microarrays (EBAM) of categorical data such as SNP data.

Should not be called directly, but via ebam(..., method = chisq.ebam).

This function replaces `cat.ebam`.

Usage

```
chisq.ebam(data, cl, approx = NULL, B = 100, n.split = 1,
   check.for.NN = FALSE, lev = NULL, B.more = 0.1, B.max = 50000,
   n.subset = 10, fast = FALSE, n.interval = NULL, df.ratio = 3,
   df.dens = NULL, knots.mode = NULL, type.nclass = "wand",
   rand = NA)
```

Arguments

data	a matrix, data frame, or list. If a matrix or data frame, then each row must correspond to a variable (e.g., a SNP), and each column to a sample (i.e.\ an observation). If the number of observations is huge it is better to specify data as a list consisting of matrices, where each matrix represents one group and summarizes how many observations in this group show which level at which variable. These matrices can be generated using the function rowTables from the package **scrime**. For details on how to specify this list, see the examples section on this man page, and the help for rowChisqMultiClass in the package **scrime**.
cl	a numeric vector of length ncol(data) indicating to which class a sample belongs. Must consist of the integers between 1 and c, where c is the number of different groups. Needs only to be specified if data is a matrix or a data frame.
approx	should the null distribution be approximated by a χ^2-distribution? Currently only available if data is a matrix or data frame. If not specified, approx = FALSE is used, and the null distribution is estimated by employing a permutation method.
B	the number of permutations used in the estimation of the null distribution, and hence, in the computation of the expected z-values.
n.split	number of chunks in which the variables are splitted in the computation of the values of the test statistic. Currently, only available if approx = TRUE and data is a matrix or data frame. By default, the test scores of all variables are calculated simultaneously. If the number of variables or observations is large, setting n.split to a larger value than 1 can help to avoid memory problems.
check.for.NN	if TRUE, it will be checked if any of the genotypes is equal to "NN". Can be very time-consuming when the data set is high-dimensional.
lev	numeric or character vector specifying the codings of the levels of the variables/SNPs. Can only be specified if data is a matrix or a data frame. Must only be specified if the variables are not coded by the integers between 1 and the number of levels. Can also be a list. In this case, each element of this list must be a numeric or character vector specifying the codings, where all elements must have the same length.
B.more	a numeric value. If the number of all possible permutations is smaller than or equal to (1+B.more)*B, full permutation will be done. Otherwise, B permutations are used.
B.max	a numeric value. If the number of all possible permutations is smaller than or equal to B.max, B randomly selected permutations will be used in the computation of the null distribution. Otherwise, B random draws of the group labels are used.

n.subset	a numeric value indicating in how many subsets the B permutations are divided when computing the permuted z-values. Please note that the meaning of n.subset differs between the SAM and the EBAM functions.
fast	if FALSE the exact number of permuted test scores that are more extreme than a particular observed test score is computed for each of the variables/SNPs. If TRUE, a crude estimate of this number is used.
n.interval	the number of intervals used in the logistic regression with repeated observations for estimating the ratio f_0/f (if approx = FALSE), or in the Poisson regression used to estimate the density of the observed z-values (if approx = TRUE). If NULL, n.interval is set to 139 if approx = FALSE, and estimated by the method specified by type.nclass if approx = TRUE.
df.ratio	integer specifying the degrees of freedom of the natural cubic spline used in the logistic regression with repeated observations. Ignored if approx = TRUE.
df.dens	integer specifying the degrees of freedom of the natural cubic spline used in the Poisson regression to estimate the density of the observed z-values. Ignored if approx = FALSE. If NULL, df.dens is set to 3 if the degrees of freedom of the appromimated null distribution, i.e.\ the χ^2-distribution, are less than or equal to 2, and otherwise df.dens is set to 5.
knots.mode	if TRUE the df.dens - 1 knots are centered around the mode and not the median of the density when fitting the Poisson regression model. Ignored if approx = FALSE. If not specified, knots.mode is set to TRUE if the degrees of freedom of the approximated null distribution, i.e.\ tht χ^2-distribution, are larger than or equal to 3, and otherwise knots.mode is set to FALSE. For details on this density estimation, see denspr.
type.nclass	character string specifying the procedure used to compute the number of cells of the histogram. Ignored if approx = FALSE or n.interval is specified. Can be either "wand" (default), "scott", or "FD". For details, see denspr.
rand	numeric value. If specified, i.e. not NA, the random number generator will be set into a reproducible state.

Details

For each variable, Pearson's Chi-Square statistic is computed to test if the distribution of the variable differs between several groups. Since only one null distribution is estimated for all variables as proposed in the original EBAM application of Efron et al. (2001), all variables must have the same number of levels/categories.

Value

A list containing statistics required by ebam.

Warning

This procedure will only work correctly if all SNPs/variables have the same number of levels/categories.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Efron, B., Tibshirani, R., Storey, J.D., and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment, *JASA*, 96, 1151-1160.

Schwender, H. and Ickstadt, K. (2008). Empirical Bayes Analysis of Single Nucleotide Polymorphisms. *BMC Bioinformatics*, 9, 144.

Schwender, H., Krause, A., and Ickstadt, K. (2003). Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany.

See Also

EBAM-class,ebam, chisq.stat

Examples

```
## Not run:
  # Generate a random 1000 x 40 matrix consisting of the values
  # 1, 2, and 3, and representing 1000 variables and 40 observations.

  mat <- matrix(sample(3, 40000, TRUE), 1000)

  # Assume that the first 20 observations are cases, and the
  # remaining 20 are controls.

  cl <- rep(1:2, e=20)

  # Then an EBAM analysis for categorical data can be done by

  out <- ebam(mat, cl, method=chisq.ebam, approx=TRUE)
  out

  # approx is set to TRUE to approximate the null distribution
  # by the ChiSquare-distribution (usually, for such a small
  # number of observations this might not be a good idea
  # as the assumptions behind this approximation might not
  # be fulfilled).

  # The same results can also be obtained by employing
  # contingency tables, i.e. by specifying data as a list.
  # For this, we need to generate the tables summarizing
  # groupwise how many observations show which level at
  # which variable. These tables can be obtained by

  library(scrime)
  cases <- rowTables(mat[, cl==1])
  controls <- rowTables(mat[, cl==2])
  ltabs <- list(cases, controls)

  # And the same EBAM analysis as above can then be
  # performed by
```

```
  out2 <- ebam(ltabs, method=chisq.ebam, approx=TRUE)
  out2

## End(Not run)
```

`chisq.stat` *SAM Analysis for Categorical Data*

Description

Generates the required statistics for a Significance Analysis of Microarrays of categorical data such as SNP data.

Should not be called directly, but via sam(..., method = chisq.stat).

Replaces `cat.stat`

Usage

```
chisq.stat(data, cl, approx = NULL, B = 100, n.split = 1,
   check.for.NN = FALSE, lev = NULL, B.more = 0.1,
   B.max = 50000, n.subset = 10, rand = NA)
```

Arguments

`data`	a matrix, data frame, or list. If a matrix or data frame, then each row must correspond to a variable (e.g., a SNP), and each column to a sample (i.e.\ an observation). If the number of observations is huge it is better to specify `data` as a list consisting of matrices, where each matrix represents one group and summarizes how many observations in this group show which level at which variable. These matrices can be generated using the function `rowTables` from the package **scrime**. For details on how to specify this list, see the examples section on this man page, and the help for `rowChisqMultiClass` in the package **scrime**.
`cl`	a numeric vector of length `ncol(data)` indicating to which class a sample belongs. Must consist of the integers between 1 and c, where c is the number of different groups. Needs only to be specified if `data` is a matrix or a data frame.
`approx`	should the null distribution be approximated by a χ^2-distribution? Currently only available if `data` is a matrix or data frame. If not specified, `approx = FALSE` is used, and the null distribution is estimated by employing a permutation method.
`B`	the number of permutations used in the estimation of the null distribution, and hence, in the computation of the expected d-values.
`n.split`	number of chunks in which the variables are splitted in the computation of the values of the test statistic. Currently, only available if `approx = TRUE` and `data` is a matrix or data frame. By default, the test scores of all variables are calculated simultaneously. If the number of variables or observations is large, setting `n.split` to a larger value than 1 can help to avoid memory problems.

check.for.NN	if TRUE, it will be checked if any of the genotypes is equal to "NN". Can be very time-consuming when the data set is high-dimensional.
lev	numeric or character vector specifying the codings of the levels of the variables/SNPs. Can only be specified if data is a matrix or a data frame. Must only be specified if the variables are not coded by the integers between 1 and the number of levels. Can also be a list. In this case, each element of this list must be a numeric or character vector specifying the codings, where all elements must have the same length.
B.more	a numeric value. If the number of all possible permutations is smaller than or equal to (1+B.more)*B, full permutation will be done. Otherwise, B permutations are used.
B.max	a numeric value. If the number of all possible permutations is smaller than or equal to B.max, B randomly selected permutations will be used in the computation of the null distribution. Otherwise, B random draws of the group labels are used.
n.subset	a numeric value indicating how many permutations are considered simultaneously when computing the expected d-values.
rand	numeric value. If specified, i.e. not NA, the random number generator will be set into a reproducible state.

Details

For each SNP (or more general, categorical variable), Pearson's Chi-Square statistic is computed to test if the distribution of the SNP differs between several groups. Since only one null distribution is estimated for all SNPs as proposed in the original SAM procedure of Tusher et al. (2001) all SNPs must have the same number of levels/categories.

Value

A list containing statistics required by sam.

Warning

This procedure will only work correctly if all SNPs/variables have the same number of levels/categories. Therefore, it is stopped when the number of levels differ between the variables.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Schwender, H. (2005). Modifying Microarray Analysis Methods for Categorical Data – SAM and PAM for SNPs. In Weihs, C. and Gaul, W. (eds.), *Classification – The Ubiquitous Challenge*. Springer, Heidelberg, 370-377.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116-5121.

See Also

SAM-class,sam, chisq.ebam, trend.stat

Examples

```
## Not run:
  # Generate a random 1000 x 40 matrix consisting of the values
  # 1, 2, and 3, and representing 1000 variables and 40 observations.

  mat <- matrix(sample(3, 40000, TRUE), 1000)

  # Assume that the first 20 observations are cases, and the
  # remaining 20 are controls.

  cl <- rep(1:2, e=20)

  # Then an SAM analysis for categorical data can be done by

  out <- sam(mat, cl, method=chisq.stat, approx=TRUE)
  out

  # approx is set to TRUE to approximate the null distribution
  # by the ChiSquare-distribution (usually, for such a small
  # number of observations this might not be a good idea
  # as the assumptions behind this approximation might not
  # be fulfilled).

  # The same results can also be obtained by employing
  # contingency tables, i.e. by specifying data as a list.
  # For this, we need to generate the tables summarizing
  # groupwise how many observations show which level at
  # which variable. These tables can be obtained by

  library(scrime)
  cases <- rowTables(mat[, cl==1])
  controls <- rowTables(mat[, cl==2])
  ltabs <- list(cases, controls)

  # And the same SAM analysis as above can then be
  # performed by

  out2 <- sam(ltabs, method=chisq.stat, approx=TRUE)
  out2

## End(Not run)
```

---

d.stat *SAM Analysis Using a Modified t-statistic*

Description

Computes the required statistics for a Significance Analysis of Microarrays (SAM) using either a (modified) t- or F-statistic.

Should not be called directly, but via the function sam.

Usage

```
d.stat(data, cl, var.equal = FALSE, B = 100, med = FALSE, s0 = NA,
    s.alpha = seq(0, 1, 0.05), include.zero = TRUE, n.subset = 10,
    mat.samp = NULL, B.more = 0.1, B.max = 30000, gene.names = NULL,
    R.fold = 1, use.dm = TRUE, R.unlog = TRUE, na.replace = TRUE,
    na.method = "mean", rand = NA)
```

Arguments

data	a matrix, data frame or ExpressionSet object. Each row of data (or exprs(data), respectively) must correspond to a variable (e.g., a gene), and each column to a sample (i.e.\ an observation).
cl	a numeric vector of length ncol(data) containing the class labels of the samples. In the two class paired case, cl can also be a matrix with ncol(data) rows and 2 columns. If data is an ExpressionSet object, cl can also be a character string. For details on how cl should be specified, see ?sam.
var.equal	if FALSE (default), Welch's t-statistic will be computed. If TRUE, the pooled variance will be used in the computation of the t-statistic.
B	numeric value indicating how many permutations should be used in the estimation of the null distribution.
med	if FALSE (default), the mean number of falsely called genes will be computed. Otherwise, the median number is calculated.
s0	a numeric value specifying the fudge factor. If NA (default), s0 will be computed automatically.
s.alpha	a numeric vector or value specifying the quantiles of the standard deviations of the genes used in the computation of s0. If s.alpha is a vector, the fudge factor is computed as proposed by Tusher et al. (2001). Otherwise, the quantile of the standard deviations specified by s.alpha is used as fudge factor.
include.zero	if TRUE, s0 = 0 will also be a possible choice for the fudge factor. Hence, the usual t-statistic or F statistic, respectively, can also be a possible choice for the expression score d. If FALSE, s0=0 will not be a possible choice for the fudge factor. The latter follows Tusher et al. (2001) definition of the fudge factor in which only strictly positive values are considered.
n.subset	a numeric value indicating how many permutations are considered simultaneously when computing the p-value and the number of falsely called genes. If med = TRUE, n.subset will be set to 1.
mat.samp	a matrix having ncol(data) columns except for the two class paired case in which mat.samp has ncol(data)/2 columns. Each row specifies one permutation of the group labels used in the computation of the expected expression

scores $\bar{d}$. If not specified (`mat.samp=NULL`), a matrix having `B` rows and `ncol(data)` is generated automatically and used in the computation of $\bar{d}$. In the two class unpaired case and the multiclass case, each row of `mat.samp` must contain the same group labels as `cl`. In the one class and the two class paired case, each row must contain -1's and 1's. In the one class case, the expression values are multiplied by these -1's and 1's. In the two class paired case, each column corresponds to one observation pair whose difference is multiplied by either -1 or 1. For more details and examples, see the manual of **siggenes**.

`B.more` a numeric value. If the number of all possible permutations is smaller than or equal to (1+`B.more`)*`B`, full permutation will be done. Otherwise, `B` permutations are used. This avoids that `B` permutations will be used – and not all permutations – if the number of all possible permutations is just a little larger than `B`.

`gene.names` a character vector of length `nrow(data)` containing the names of the genes.

`B.max` a numeric value. If the number of all possible permutations is smaller than or equal to `B.max`, `B` randomly selected permutations will be used in the computation of the null distribution. Otherwise, `B` random draws of the group labels are used. In the latter way of permuting it is possible that some of the permutations are used more than once.

`R.fold` a numeric value. If the fold change of a gene is smaller than or equal to `R.fold`, or larger than or equal to 1/`R.fold`,respectively, then this gene will be excluded from the SAM analysis. The expression score d of excluded genes is set to `NA`. By default, `R.fold` is set to 1 such that all genes are included in the SAM analysis. Setting `R.fold` to 0 or a negative value will avoid the computation of the fold change. The fold change is only computed in the two-class unpaired cases.

`use.dm` if `TRUE`, the fold change is computed by 2 to the power of the difference between the mean log2 intensities of the two groups, i.e.\ 2 to the power of the numerator of the test statistic. If `FALSE`, the fold change is determined by computing 2 to the power of `data` (if `R.unlog = TRUE`) and then calculating the ratio of the mean intensity in the group coded by 1 to the mean intensity in the group coded by 0. The latter is the definition of the fold change used in Tusher et al.\ (2001).

`R.unlog` if `TRUE`, the anti-log of `data` will be used in the computation of the fold change. Otherwise, `data` is used. This transformation should be done when `data` is log2-tranformed (in a SAM analysis it is highly recommended to use log2-transformed expression data). Ignored if `use.dm = TRUE`.

`na.replace` if `TRUE`, missing values will be removed by the genewise/rowwise statistic specified by `na.method`. If a gene has less than 2 non-missing values, this gene will be excluded from further analysis. If `na.replace=FALSE`, all genes with one or more missing values will be excluded from further analysis. The expression score d of excluded genes is set to `NA`.

`na.method` a character string naming the statistic with which missing values will be replaced if `na.replace=TRUE`. Must be either `"mean"` (default) or `median`.

`rand` numeric value. If specified, i.e. not `NA`, the random number generator will be set into a reproducible state.

Value

An object of class SAM.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Schwender, H., Krause, A. and Ickstadt, K. (2003). Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116-5121.

See Also

SAM-class,sam, z.ebam

`delta.plot` *Delta Plots*

Description

Generates both a plot of Δ vs. the FDR and a plot of Δ vs.\ the number of identified genes in a SAM analysis.

Usage

```
delta.plot(object, delta = NULL, helplines = FALSE)
```

Arguments

`object`	a object of class SAM.
`delta`	a vector of values for Δ. If `NULL`, a default set of Δ values will be used.
`helplines`	if `TRUE`, help lines will be drawn in the Δ plots.

Details

The Δ plots are a visualization of the table generated by `sam` that contains the estimated FDR and the number of identified genes for a set of Δ values.

Value

Two plots in one graphsheet: The plot of Δ vs. FDR and the plot of Δ vs. the number of identified genes.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Tusher, V., Tibshirani, R., and Chu, G. (2001). Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *PNAS*, 98, 5116-5121.

See Also

SAM-class,sam

Examples

```
## Not run:
  # Load the package multtest and the data of Golub et al. (1999)
  # contained in multtest.
  library(multtest)
  data(golub)

  # Perform a SAM analysis.
  sam.out<-sam(golub, golub.cl, B=100, rand=123)

  # Generate the Delta plots for the default set of Deltas computed by sam.
  delta.plot(sam.out)

  # Another way of generating the same plot.
  plot(sam.out)

  # Generate the Delta plots for Delta = 0.2, 0.4, ..., 2.
  plot(sam.out, seq(0.2, 2, 0.2))

## End(Not run)
```

---

denspr *Density Estimation*

Description

Estimates the density of a vector of observations by a Poisson regression fit to histogram counts.

Usage

```
denspr(x, n.interval = NULL, df = 5, knots.mode = TRUE,
    type.nclass = c("wand", "scott", "FD"), addx=FALSE)
```

Arguments

x	a numeric vector containing the observations for which the density should be estimated.
n.interval	an integer specifying the number of cells for the histogram. If NULL, n.interval is estimated by the method specified by type.nclass.
df	integer specifying the degrees of freedom of the natural cubic spline used in the Poisson regression fit.
knots.mode	if TRUE the df - 1 knots are centered around the mode and not the median of the density, where the mode is estimated by the midpoint of the cell of the histogram that contains the largest number of observations. If FALSE, the default knots are used in the function ns. Thus, if FALSE the basis matrix will be generated by ns(x, df = 5).
type.nclass	character string specifying the procedure used to compute the number of cells of the histogram. Ignored if n.interval is specified. By default, the method of Wand (1994) with level = 1 (see the help page of dpih in the package **KernSmooth**) is used. For the other choices, see nclass.scott.
addx	should x be added to the output? Necessary when the estimated density should be plotted by plot(out) or lines(out), where out is the output of denspr.

Value

An object of class denspr consisting of

y	a numeric vector of the same length as x containing the estimated density for each of the observations
center	a numeric vector specifying the midpoints of the cells of the histogram
counts	a numeric vector of the same length as center composed of the number of observations of the corresponding cells
x.mode	the estimated mode
ns.out	the output of ns
type	the method used to estimate the numbers of cells
x	the input vector x if addx = TRUE; otherwise, NULL.

Author(s)

Holger Schwender,<holger.schw@gmx.de>

References

Efron, B., and Tibshirani, R. (1996). Using specially designed exponential families for density estimation. *Annals of Statistics*, 24, 2431–2461.

Wand, M.P. (1997). Data-based choice of histogram bin width. *American Statistician*, 51, 59–64.

See Also

cat.ebam

Examples

```
## Not run:
# Generating some random data.
x <- rnorm(10000)
out <- denspr(x, addx=TRUE)
plot(out)

# Or for an asymmetric density.
x <- rchisq(10000, 2)
out <- denspr(x, df=3, addx=TRUE)
plot(out)

## End(Not run)
```

---

ebam *Empirical Bayes Analysis of Microarrays*

Description

Performs an Empirical Bayes Analysis of Microarrays (EBAM). It is possible to perform one and two class analyses using either a modified t-statistic or a (standardized) Wilcoxon rank statistic, and a multiclass analysis using a modified F-statistic. Moreover, this function provides a EBAM procedure for categorical data such as SNP data and the possibility to employ an user-written score function.

Usage

```
ebam(x, cl, method = z.ebam, delta = 0.9, which.a0 = NULL,
     control = ebamControl(), gene.names = dimnames(x)[[1]],
     ...)
```

Arguments

x	either a matrix, a data frame or an ExpressionSet object, or the output of find.a0, i.e.\ an object of class FindA0. Can also be a list (if method = chisq.ebam or method = trend.ebam). For the latter case, see chisq.ebam. If x is not a FindA0 object, then each row of x (or exprs(x), respectively) must correspond to a variable (e.g., a gene or a SNP), and each column to a sample.
cl	a specification of the class labels of the samples. Ignored if x is a FindA0 object. Needs not to be specified if x is a list. Typically, cl is specified by a vector of length ncol(x). In the two class paired case, cl can also be a matrix with ncol(x) rows and 2 columns. If x is an ExpressionSet object, cl can also be a character string naming the column of pData(x) that contains the class labels of the samples. In the one-class case, cl should be a vector of 1's. In the two class unpaired case, cl should be a vector containing 0's (specifying the samples of, e.g., the control group) and 1's (specifying, e.g., the case group).

In the two class paired case, cl can be either a numeric vector or a numeric matrix. If it is a vector, then cl has to consist of the integers between -1 and $-n/2$ (e.g., before treatment group) and between 1 and $n/2$ (e.g., after treatment group), where n is the length of cl and k is paired with $-k$, $k = 1, \ldots, n/2$. If cl is a matrix, one column should contain -1's and 1's specifying, e.g., the before and the after treatment samples, respectively, and the other column should contain integer between 1 and $n/2$ specifying the $n/2$ pairs of observations.

In the multiclass case and if method = chisq.ebam or method = trend.ebam, cl should be a vector containing integers between 1 and g, where g is the number of groups. In the two latter cases, cl needs not to be specified, if x is a list. For details, see chisq.ebam.

For examples of how cl can be specified, see the manual of **siggenes**.

method — a character string or name specifying the method or function that should be used in the computation of the expression score z.

If method = z.ebam, a modified t- or F-statistic, respectively, will be computed as proposed by Efron et al. (2001).

If method = wilc.ebam, a (standardized) Wilcoxon sum / signed rank statistic will be used as expression score.

For an analysis of categorical data such as SNP data, method can be set to chisq.ebam. In this case, Pearson's Chi-squared statistic is computed for each row.

If the variables are ordinal and a trend test should be applied (e.g., in the two-class case, the Cochran-Armitage trend test), method = trend.ebam can be employed.

It is also possible to employ an user-written function for computing an user-specified expression score. For details, see the vignette of **siggenes**.

delta — a numeric vector consisting of probabilities for which the number of differentially expressed genes and the FDR should be computed, where a gene is called differentially expressed if its posterior probability is larger than Δ.

which.a0 — an integer between 1 and the length of quan.a0 of find.a0. If NULL, the suggested choice of find.a0 is used. Ignored if x is a matrix, data frame or ExpressionSet object.

control — further arguments for controlling the EBAM analysis. For these arguments, see ebamControl.

gene.names — a vector of length nrow(x) specifying the names of the variables. By default, the row names of the matrix / data frame comprised by x are used.

... — further arguments of the specific EBAM methods. If method = z.ebam, see z.ebam. If method = wilc.ebam, see wilc.ebam. If method = chisq.ebam, see chisq.ebam.

Value

An object of class EBAM.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment. *JASA*, 96, 1151-1160.

Schwender, H., Krause, A., and Ickstadt, K. (2006). Identifying Interesting Genes with siggenes. *RNews*, 6(5), 45-50.

Storey, J.D. and Tibshirani, R. (2003). Statistical Significance for Genome-Wide Studies. *Proceedings of the National Academy of Sciences*, 100, 9440-9445.

See Also

EBAM-class, find.a0, z.ebam, wilc.ebam, chisq.ebam

Examples

```
## Not run:
  # Load the data of Golub et al. (1999) contained in the package multtest.
  data(golub)

  # golub.cl contains the class labels.
  golub.cl

  # Perform an EBAM analysis for the two class unpaired case assuming
  # unequal variances. Specify the fudge factor a0 by the suggested
  # choice of find.a0
  find.out <- find.a0(golub, golub.cl, rand = 123)
  ebam.out <- ebam(find.out)
  ebam.out

  # Since a0 = 0 leads to the largest number of genes (i.e. the suggested
  # choice of a0), the following leads to the same results as the above
  # analysis (but only if the random number generator, i.e. rand, is set
  # to the same number).
  ebam.out2 <- ebam(golub, golub.cl, a0 = 0, fast = TRUE, rand = 123)
  ebam.out2

  # If fast is set to TRUE in ebam, a crude estimate of the number of
  # falsely called genes is used (see the help file for z.ebam). This
  # estimate is always employed in find.a0.
  # The exact number is used in ebam when performing
  ebam.out3 <- ebam(golub, golub.cl, a0 = 0, rand = 123)
  ebam.out3

  # Since this is the recommended way, we use ebam.out3 at the end of
  # the Examples section for further analyses.



  # Perform an EBAM analysis for the two class unpaired case assuming
  # equal group variances. Set a0 = 0, and use B = 50 permutations
  # of the class labels.
  ebam.out4 <- ebam(golub, golub.cl, a0 = 0, var.equal = TRUE, B = 50,
```

```
   rand = 123)
ebam.out4

# Perform an EBAM analysis for the two class unpaired cased assuming
# unequal group variances. Use the median (i.e. the 50% quantile)
# of the standard deviations of the genes as fudge factor a0. And
# obtain the number of genes and the FDR if a gene is called
# differentially when its posterior probability is larger than
# 0.95.
ebam.out5 <- ebam(golub, golub.cl, quan.a0 = 0.5, delta = 0.95,
   rand = 123)
ebam.out5

# For the third analysis, obtain the number of differentially
# expressed genes and the FDR if a gene is called differentially
# expressed if its posterior probability is larger than 0.8, 0.85,
# 0.9, 0.95.
print(ebam.out3, c(0.8, 0.85, 0.9, 0.95))

# Generate a plot of the posterior probabilities for delta = 0.9.
plot(ebam.out3, 0.9)

# Obtain the list of genes called differentially expressed if their
# posterior probability is larger than 0.99, and gene-specific
# statistics for these variables such as their z-value and their
# local FDR.
summary(ebam.out3, 0.99)

## End(Not run)
```

`EBAM-class` *Class EBAM*

Description

This is a class representation for the Empirical Bayes Analysis of Microarrays (EBAM) proposed by Efron et al. (2001).

Objects from the Class

Objects can be created using the function `ebam`.

Slots

`z`: Object of class `"numeric"` representing the expression scores of the genes.

`posterior`: Object of class `"numeric"` representing the posterior probabilities of the genes.

`p0`: Object of class `"numeric"` specifying the prior probability that a gene is not differentially expressed.

`local`: Object of class `"numeric"` consisting of the local FDR estimates for the genes.

`mat.fdr`: Object of class `"matrix"` containing general statistics such as the number of differentially expressed genes and the estimated FDR for the specified values of `delta`.

`a0`: Object of class `"numeric"` specifying the used value of the fudge factor. If not computed, `a0` will be set to `numeric(0)`.

`mat.samp`: Object of class `"matrix"` containing the permuted group labels used in the estimation of the null distribution. Each row represents one permutation, each column one observation (pair). If no permutation procedure has been used, `mat.samp` will be set to `matrix(numeric(0))`.

`vec.pos`: Object of class `"numeric"` consisting of the number of positive permuted test scores that are absolutely larger than the test score of a particular gene for each gene. If not computed `vec.pos` is set to `numeric(0)`.

`vec.neg`: Object of class `"numeric"` consisting of the number of negative permuted test scores that are absolutely larger than the test score of a particular gene for each gene. If not computed `vec.neg` is set to `numeric(0)`.

`msg`: Object of class `"character"` containing information about, e.g., the type of analysis. `msg` is printed when the functions `print` and `summary` are called.

`chip`: Object of class `"character"` naming the microarray used in the analysis. If no information about the chip is available, `chip` will be set to `""`.

Methods

plot `signature(object = "EBAM")`: Generates a plot of the posterior probabilities of the genes for a specified value of Δ. For details, see `help.ebam(plot)`. For the arguments, see `args.ebam(plot)`.

print `signature(object = "EBAM")`: Prints general information such as the number of differentially expressed genes and the estimated FDR for several values of Δ. For details, see `help.ebam(print)`. Arguments can be listed by `args.ebam(print)`.

show `signature(object = "EBAM")`: Shows the output of an EBAM analysis.

summary `signature(object = "EBAM")`: Summarizes the results of an EBAM analysis for a specified value of Δ. For details, see `help.ebam(summary)`. For the arguments, see `args.ebam(summary)`.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment, *JASA*, 96, 1151-1160.

Schwender, H., Krause, A. and Ickstadt, K. (2003). Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany.

See Also

`ebam`, `find.a0`, `FindA0-class`

Examples

```
## Not run:
  # Load the data of Golub et al. (1999) contained in the package multtest.
  data(golub)

  # golub.cl contains the class labels.
  golub.cl

  # Perform an EBAM analysis for the two class unpaired case assuming
  # unequal variances. Specify the fudge factor a0 by the suggested
  # choice of find.a0
  find.out <- find.a0(golub, golub.cl, rand = 123)
  ebam.out <- ebam(find.out)
  ebam.out

  # Obtain the number of differentially
  # expressed genes and the FDR if a gene is called differentially
  # expressed if its posterior probability is larger than 0.8, 0.85,
  # 0.9, 0.95.
  print(ebam.out, c(0.8, 0.85, 0.9, 0.95))

  # Generate a plot of the posterior probabilities for delta = 0.9.
  plot(ebam.out, 0.9)

  # Obtain the list of genes called differentially expressed if their
  # posterior probability is larger than 0.99, and gene-specific
  # statistics for these variables such as their z-value and their
  # local FDR.
  summary(ebam.out, 0.9)


## End(Not run)
```

`ebamControl` *Further EBAM Arguments*

Description

Specifies most of the optional arguments of `ebam` and `find.a0`.

Usage

```
ebamControl(p0 = NA, p0.estimation = c("splines", "interval", "adhoc"),
   lambda = NULL, ncs.value = "max", use.weights = FALSE)

find.a0Control(p0.estimation = c("splines", "adhoc", "interval"),
   lambda = NULL, ncs.value = "max", use.weights = FALSE,
   n.chunk = 5, n.interval = 139, df.ratio = NULL)
```

Arguments

p0	a numeric value specifying the prior probability p_0 that a gene is not differentially expressed. If NA, p0 will be estimated automatically.
p0.estimation	either "splines" (default), "interval", or "adhoc". If "splines", the spline based method of Storey and Tibshirani (2003) is used to estimate p_0. If "adhoc" ("interval"), the adhoc (interval based) method proposed by Efron et al.\ (2001) is used to estimate p_0.
lambda	a numeric vector or value specifying the λ values used in the estimation of p_0. If NULL, lambda is set to seq(0, 0.95, 0.05) if p0.estimation = "splines", and to 0.5 if p0.estimation = "interval". Ignored if p0.estimation = "adhoc". For details, see pi0.est.
ncs.value	a character string. Only used if p0.estimation = "splines" and lambda is a vector. Either "max" or "paper". For details, see pi0.est.
use.weights	should weights be used in the spline based estimation of p_0? If TRUE, 1 - lambda is used as weights. For details, see pi0.est.
n.chunk	an integer specifying in how many subsets the B permutations should be split when computing the permuted test scores.
n.interval	the number of intervals used in the logistic regression with repeated observations for estimating the ratio f_0/f.
df.ratio	integer specifying the degrees of freedom of the natural cubic spline used in the logistic regression with repeated observations.

Details

These parameters should only be changed if they are fully understood.

Value

A list containing the values of the parameters that are used in ebam or find.a0, respectively.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment. *JASA*, 96, 1151-1160.

Storey, J.D. and Tibshirani, R. (2003). Statistical Significance for Genome-Wide Studies. *Proceedings of the National Academy of Sciences*, 100, 9440-9445.

See Also

limma2ebam, ebam, find.a0

`find.a0` *Computation of the Fudge Factor*

Description

Suggests an optimal value for the fudge factor in an EBAM analysis as proposed by Efron et al. (2001).

Usage

```
find.a0(data, cl, method = z.find, B = 100, delta = 0.9,
    quan.a0 = (0:5)/5, include.zero = TRUE,
    control = find.a0Control(), gene.names = dimnames(data)[[1]],
    rand = NA, ...)
```

Arguments

`data`	a matrix, data frame or an ExpressionSet object. Each row of `data` (or `exprs(data)`, respectively) must correspond to a variable (e.g., a gene), and each column to a sample (i.e.\ an observation).
`cl`	a numeric vector of length `ncol(data)` containing the class labels of the samples. In the two class paired case, `cl` can also be a matrix with `ncol(data)` rows and 2 columns. If `data` is an ExpressionSet object, `cl` can also be a character string naming the column of `pData(data)` that contains the class labels of the samples. In the one-class case, `cl` should be a vector of 1's. In the two class unpaired case, `cl` should be a vector containing 0's (specifying the samples of, e.g., the control group) and 1's (specifying, e.g., the case group). In the two class paired case, `cl` can be either a numeric vector or a numeric matrix. If it is a vector, then `cl` has to consist of the integers between -1 and $-n/2$ (e.g., before treatment group) and between 1 and $n/2$ (e.g., after treatment group), where n is the length of `cl` and k is paired with $-k$, $k = 1, \ldots, n/2$. If `cl` is a matrix, one column should contain -1's and 1's specifying, e.g., the before and the after treatment samples, respectively, and the other column should contain integer between 1 and $n/2$ specifying the $n/2$ pairs of observations. In the multiclass case and if `method = cat.stat`, `cl` should be a vector containing integers between 1 and g, where g is the number of groups. For examples of how `cl` can be specified, see the manual of **siggenes**.
`method`	the name of a function for computing the numerator and the denominator of the test statistic of interest, and for specifying other objects required for the identification of the fudge factor. The default function `z.find` provides these objects for t- and F-statistics. It is, however, also possible to employ an user-written function. For how to write such a function, see the vignette of **siggenes**.
`B`	the number of permutations used in the estimation of the null distribution.

delta	a probability. All genes showing a posterior probability that is larger than or equal to delta are called differentially expressed.
quan.a0	a numeric vector indicating over which quantiles of the standard deviations of the genes the fudge factor a_0 should be optimized.
include.zero	should $a_0 = 0$, i.e. the not-modified test statistic also be a possible choice for the fudge factor?
control	further arguments for controlling the EBAM analysis with find.a0. For these arguments, see find.a0Control.
gene.names	a character vector of length nrow(data) containing the names of the genes. By default, the row names of data are used.
rand	integer. If specified, i.e. not NA, the random number generator will be set into a reproducible state.
...	further arguments for the function specified by fun. For further arguments of fun = z.find, see z.find.

Details

The suggested choice for the fudge factor is the value of a_0 that leads to the largest number of genes showing a posterior probability larger than delta.

Actually, only the genes having a posterior probability larger than delta are called differentially expressed that do not exhibit a test score less extreme than the score of a gene whose posterior probability is less than delta. So, let's say, we have done an EBAM analysis with a t-test and we have ordered the genes by their t-statistic. Let's further assume that Gene 1 to Gene 5 (i.e. the five genes with the lowest t-statistics), Gene 7 and 8, Gene 3012 to 3020, and Gene 3040 to 3051 are the only genes that show a posterior probability larger than delta. Then, Gene 1 to 5, and 3040 to 3051 are called differentially expressed, but Gene 7 and 8, and 3012 to 3020 are not called differentially expressed, since Gene 6 and Gene 3021 to 3039 show a posterior probability less than delta.

Value

An object of class FindA0.

Note

The numbers of differentially expressed genes can differ between find.a0 and ebam, even though the same value of the fudge factor is used, since in find.a0 the observed and permuted test scores are monotonically transformed such that the observed scores follow a standard normal distribution (if the test statistic can take both positive and negative values) and an F-distribution (if the test statistic can only take positive values) for each possible choice of the fudge factor.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment, *JASA*, 96, 1151-1160.

See Also

ebam, FindA0-class, find.a0Control

Examples

```
## Not run:
  # Load the data of Golub et al. (1999) contained in the package multtest.
  data(golub)

  # golub.cl contains the class labels.
  golub.cl

  # Obtain the number of differentially expressed genes and the FDR for the
  # default set of values for the fudge factor.
  find.out <- find.a0(golub, golub.cl, rand = 123)
  find.out

  # Obtain the number of differentially expressed genes and the FDR when using
  # the t-statistic assuming equal group variances
  find.out2 <- find.a0(golub, golub.cl, var.equal = TRUE, rand = 123)

  # Using the Output of the first analysis with find.a0, the number of
  # differentially expressed genes and the FDR for other values of
  # delta, e.g., 0.95, can be obtained by
  print(find.out, 0.95)

  # The logit-transformed posterior probabilities can be plotted by
  plot(find.out)

  # To avoid the logit-transformation, set logit = FALSE.
  plot(find.out, logit = FALSE)

## End(Not run)
```

FindA0class *Class FindA0*

Description

This is a class representation for the specification of the fudge factor in an EBAM analysis as proposed by Efron et al. (2001).

Objects from the Class

Objects can be created using the function find.a0.

Slots

`mat.z`: Object of class `"matrix"` containing the expression scores of the genes for each of the possible values for the fudge factor, where each row corresponds to a gene, and each column to one of the values for the fudge factor a_0.

`mat.posterior`: Object of class `"matrix"` consisting of the posterior probabilities of the genes for each of the possible values for the fudge factor, where each row of `mat.posterior` corresponds to a gene, and each column to one of the values for a_0. The probabilities in `mat.posterior` are computed using the monotonically transformed test scores (see the Details section of `find.a0`).

`mat.center`: Object of class `"matrix"` representing the centers of the `nrow(mat.center)` intervals used in the logistic regression with repeated observations for estimating f/f_0 for each of the `ncol(mat.center)` possible values for the fudge factor.

`mat.success`: Object of class `"matrix"` consisting of the numbers of observed test scores in the `nrow(mat.success)` intervals used in the logistic regression with repeated observations for each of the `ncol(mat.success)` possible values for the fudge factor.

`mat.failure`: Object of class `"matrix"` containing the numbers of permuted test scores in the `nrow(mat.failure)` intervals used in the logistic regression with repeated observations for each of the `ncol(mat.failure)` possible values for the fudge factor.

`z.norm`: Object of class `"numeric"` comprising the values of the `nrow(mat.z)` quantiles of the standard normal distribution (if any `mat.z<0`) or an F-distribution (if all `mat.z >= 0`).

`p0`: Object of class `"numeric"` specifying the prior probability that a gene is not differentially expressed.

`mat.a0`: Object of class `"data.frame"` comprising the number of differentially expressed genes and the estimated FDR for the possible choices of the fudge factor specified by `vec.a0`.

`mat.samp`: Object of class `"matrix"` consisting of the `nrow{mat.samp}` permutations of the class labels.

`vec.a0`: Object of class `"numeric"` representing the possible values of the fudge factor a_0.

`suggested`: Object of class `"numeric"` revealing the suggested choice for the fudge factor, i.e. the value of `vec.a0` that leads to the largest number of differentially expressed genes.

`delta`: Object of class `"numeric"` specifying the minimum posterior probability that a gene must have to be called differentially expressed.

`df.ratio`: Object of class `"numeric"` representing the degrees of freedom of the natural cubic spline used in the logistic regression with repeated observations.

`msg`: Object of class `"character"` containing information about, e.g., the type of analysis. `msg` is printed when `print` is called.

`chip`: Object of class `"character"` naming the microarray used in the analysis. If no information about the chip is available, `chip` will be set to `""`.

Methods

plot `signature(object = "FindA0")`: Generates a plot of the (logit-transformed) posterior probabilities of the genes for a specified value of Δ and a set of possible values for the fudge factor. For details, see `help.finda0(plot)`. For the arguments, see `args.finda0(plot)`.

print `signature(object = "FindA0")`: Prints the number of differentially expressed genes and the estimated FDR for each of the possible values of the fudge factor specified by `vec.a0`. For details, see `help.finda0(print)`. For arguments, see `args.finda0(print)`.

show `signature(object = "FindA0")`: Shows the output of an analysis with `find.a0`.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment, *JASA*, 96, 1151-1160.

Schwender, H., Krause, A. and Ickstadt, K. (2003). Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany.

See Also

`find.a0`, `ebam`, `EBAM-class`

---

findDelta *Finding the Threshold Delta*

Description

Computes the value of the threshold Delta for a given FDR or number of genes/variables in a SAM or EBAM analysis.

Usage

```
findDelta(object, fdr = NULL, genes = NULL, prec = 6, initial = NULL,
   verbose = FALSE)
```

Arguments

`object`	either a SAM or an EBAM object.
`fdr`	numeric value between 0 and 1 for which the threshold Delta and thus the number of genes/variables should be obtained. Only one of `fdr` and `genes` can be specified.
`genes`	integer specifying the number of genes/variables for which the threshold Delta and thus the estimated FDR should be obtained. Only one of `fdr` and `genes` can be specified.
`prec`	integer indicating the precision of the considered Delta values.

initial	a numeric vector of length two containing the minimum and the maximum value of Delta that is initially used in the search for Delta. Both values must be larger than 0. If object is an EBAM object, both values must also be smaller than or equal to 1. If not specified, the minimum is set to 0.1, and the maximum to either the maximum posterior (EBAM) or the maximum absolute distance between the observed and the corresponding expected values of the test statistic (SAM).
verbose	should more information about the search process be shown?

Value

If a value of Delta is found for the exact value of fdr or genes, then a vector of length 3 consisting of Delta and the corresponding number of genes and the estimated FDR. If such a value is not found, then a matrix with two rows and three columns, where the two rows contain the number of genes/variables and the estimated FDR for the two considered values of Delta that provide the closest upper and lower bounds to the desired FDR (if fdr is specified) or number of genes/variables (if genes is specified.)

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

See Also

sam, ebam

---

fudge2 *Fudge Factor*

Description

Computes the fudge factor as described by Tusher et al. (2001).

Usage

```
fudge2(r, s, alpha = seq(0, 1, 0.05), include.zero = TRUE)
```

Arguments

r	a numeric vector. The numerator of the test statistic computed for each gene is represented by one component of this vector.
s	a numeric vector. Each component of this vector corresponds to the denominator of the test statistic of a gene.
alpha	a numeric value or vector specifying quantiles of the s values. If alpha is numeric, this quantile of s will be used as fudge factor. Otherwise, the alpha quantile of the s values is computed that is optimal following the criterion of Tusher et al.\ (2001).
include.zero	if TRUE, $s_0 = 0$ is also a possible choice for the fudge factor.

Value

s.zero	the value of the fudge factor s_0.
alpha.hat	the optimal quantile of the s values. If $s_0 = 0$, alpha.hat will not be returned.
vec.cv	the vector of the coefficients of variations. Following Tusher et al. (2001), the optimal alpha quantile is given by the quantile that leads to the smallest CV of the modified test statistics.
msg	a character string summarizing the most important information about the fudge factor.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Tusher, V., Tibshirani, R., and Chu, G. (2001). Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *PNAS*, 98, 5116-5121.

See Also

SAM-class,sam

---

fuzzy.ebam *EBAM and SAM for Fuzzy Genotype Calls*

Description

Computes the required statistics for an Empirical Bayes Analysis of Microarrays (EBAM; Efron et al., 2001) or a Significant Analysis of Microarrays (SAM; Tusher et al., 2001), respectively, based on the score statistic proposed by Louis et al. (2010) for fuzzy genotype calls or approximate Bayes Factors (Wakefield, 2007) determined using this score statistic.

Should not be called directly, but via ebam(..., method = fuzzy.ebam) or sam(..., method = fuzzy.stat), respectively.

Usage

```
fuzzy.ebam(data, cl, type = c("asymptotic", "permutation", "abf"), W = NULL,
    logbase = exp(1), addOne = TRUE, df.ratio = NULL, n.interval = NULL,
    df.dens = 5, knots.mode = TRUE, type.nclass = c("FD", "wand", "scott"),
    fast = FALSE, B = 100, B.more = 0.1, B.max = 30000, n.subset = 10, rand = NA)

fuzzy.stat(data, cl, type = c("asymptotic", "permutation", "abf"), W = NULL,
    logbase = exp(1), addOne = TRUE, B = 100, B.more = 0.1, B.max = 30000,
    n.subset = 10, rand = NA)
```

Arguments

data	a matrix containing fuzzy genotype calls. Such a matrix can, e.g., be generated by the function getMatFuzzy from the R package scrime based on the confidences for the three possible genotypes computed by preprocessing algorithms such as CRLMM.
cl	a vector of zeros and ones specifying which of the columns of data contains the fuzzy genotype calls for the cases (1) and which the controls (0). Thus, the length of cl must be equal to the number of columns of data.
type	a character string specifying how the analysis should be performed. If "asymptotic", the trend statistic of Louis et al. (2010) is used directly, and EBAM or SAM are performed assuming that under the null hypothesis this test statistic follows am asymptotic standard normal distribution. If "permutation", a permutation procedure is employed to estimate the null distribution of this test statistic. If "abf", Approximate Bayes Factors (ABF) proposed by Wakefield (2007) are determined from the trend statistic, and EBAM or SAM are performed on these ABFs or transformations of these ABFs (see in particular logbase and addOne). In the latter case, again, a permutation procedure is used in EBAM and SAM to, e.g., compute posterior probabilities of association.
W	the prior variance. Must be either a positive value or a vector of length nrow(data) consisting of positive values. Ignored if type = "asymptotic" or type = "permutation". For details, see abf.
logbase	a numeric value larger than 1. If type = "abf", then the ABFs are not directly used in the analysis, but a log-transformation (with base logbase) of the ABFs. If the ABFs should not be transformed, logbase can be set to NA. Ignored if type = "asymptotic" or type = "permutation".
addOne	should 1 be added to the ABF before it is log-transformed? If TRUE, log(ABF + 1, base=logbase) is used as test score in EBAM or SAM. If FALSE, log(ABF, base = logbase) is considered. Only taken into account when type = "abf" and logbase is not NA.
df.ratio	integer specifying the degrees of freedom of the natural cubic spline used in the logistic regression with repeated observations for estimating the ratio f_0/f. Ignored if type = "asymptotic". If not specified, df.ratio is set to 3 if type = "abf", and to 5 if type = "permutation"
n.interval	the number of intervals used in the logistic regression with repeated observations (if type = "permutation" or type = "abf"), or in the Poisson regression used to estimate the density of the observed z-values (if type = "asymptotic"). If NULL, n.interval is estimated by the method specified by type.nclass, where at least 139 intervals are considered if type = "permutation" or type = "abf".
df.dens	integer specifying the degrees of freedom of the natural cubic spline used in the Poisson regression to estimate the density of the observed z-values in an application of ebam with type = "asymptotic". Otherwise, ignored.
knots.mode	logical specifying whether the df.dens - 1 knots are centered around the mode and not the median of the density when fitting the Poisson regression model to estimate the density of the observed z-values in an application of ebam with type = "asymptotic" (for details on this density estimation, see denspr). Ignored if type = "permutation" or type = "abf".

type.nclass	character string specifying the procedure used to compute the number of cells of the histogram. Ignored if type = "permutation", type = "abf", or n.interval is specified. Can be either "FD" (default), "wand", or "FD". For details, see denspr.
fast	if FALSE the exact number of permuted test scores that are more extreme than a particular observed test score is computed for each of the variables/SNPs. If TRUE, a crude estimate of this number is used.
B	the number of permutations used in the estimation of the null distribution, and hence, in the computation of the expected z-values. Ignored if type = "asymptotic".
B.more	a numeric value. If the number of all possible permutations is smaller than or equal to (1+B.more)*B, full permutation will be done. Otherwise, B permutations are used.
B.max	a numeric value. If the number of all possible permutations is smaller than or equal to B.max, B randomly selected permutations will be used in the computation of the null distribution. Otherwise, B random draws of the group labels are used.
n.subset	a numeric value indicating in how many subsets the B permutations are divided when computing the permuted z-values. Please note that the meaning of n.subset differs between the SAM and the EBAM functions.
rand	numeric value. If specified, i.e. not NA, the random number generator will be set into a reproducible state.

Value

A list containing statistics required by ebam or sam.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Efron, B., Tibshirani, R., Storey, J.D., and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment, *JASA*, 96, 1151-1160.

Louis, T.A., Carvalho, B.S., Fallin, M.D., Irizarry, R.A., Li, Q., and Ruczinski, I. (2010). Association Tests that Accommodate Genotyping Errors. In Bernardo, J.M., Bayarri, M.J., Berger, J.O., Dawid, A.P., Heckerman, D., Smith, A.F.M., and West, M. (eds.), *Bayesian Statistics 9*, 393-420. Oxford University Press, Oxford, UK. With Discussion.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *PNAS*, 98, 5116-5121.

Wakefield, J. (2007). A Bayesian Measure of Probability of False Discovery in Genetic Epidemiology Studies. *AJHG*, 81, 208-227.

See Also

ebam, sam, EBAM-class, SAM-class

help.ebam *Help files or argument list for EBAM-specific methods*

Description

Displays the help page or the argument list, respectively, for a EBAM-specific method.

Usage

```
help.ebam(method)
args.ebam(method)
```

Arguments

method	a name or a character string specifying the method for which the arguments or the help page, respectively, should be shown. Currently available are print, plot, and summary.

Value

The arguments of the specified method are displayed or a html page containing the help for the specified method is opened, respectively.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

EBAM-class, ebam

Examples

```
## Not run:
   # Displays the arguments of the function summary
   args.ebam(summary)

   # Opens the help page in the browser
   help.ebam(summary)

## End(Not run)
```

help.finda0 *Help files or argument list for FindA0-specific methods*

Description

Displays the help page or the argument list, respectively, for a FindA0-specific method.

Usage

```
help.finda0(method)
args.finda0(method)
```

Arguments

method	a name or a character string specifying the method for which the arguments or the help page, respectively, should be shown. Currently available are print and plot.

Value

The arguments of the specified method are displayed or a html page containing the help for the specified method is opened, respectively.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

FindA0-class, find.a0

Examples

```
## Not run:
   # Displays the arguments of the function summary
   args.finda0(summary)

   # Opens the help page in the browser
   help.finda0(summary)

## End(Not run)
```

help.sam *Help files or argument list for SAM-specific methods*

Description

Displays the help page or the argument list, respectively, for a SAM-specific method.

Usage

```
help.sam(method)
args.sam(method)
```

Arguments

method	a name or a character string specifying the method for which the arguments or the help page, respectively, should be shown. Currently available are print, plot, summary and identify.

Value

The arguments of the specified method are displayed or a html page containing the help for the specified method is opened, respectively.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

SAM-class,sam

Examples

```
## Not run:
   # Displays the arguments of the function summary
   args.sam(summary)

   # Opens the help page in the browser
   help.sam(summary)

## End(Not run)
```

limma2sam *limma to SAM or EBAM*

Description

Transforms the output of an analysis with **limma** into a SAM or EBAM object, such that a SAM or EBAM analysis, respectively, can be performed using the test statistics provided by **limma**.

Usage

```
limma2sam(fit, coef, moderate = TRUE, sam.control = samControl())

limma2ebam(fit, coef, moderate = TRUE, delta = 0.9,
   ebam.control = ebamControl())
```

Arguments

fit	an object of class MArrayLM, i.e.\ the output of the functions eBayes and lmFit from the **limma** package.
coef	column number or name corresponding to the coefficient or contrast of interest. For details, see the argument coef of the function topTable in **limma**.
moderate	should the **limma** t-statistic be considered? If FALSE, the ordinary t-statistic is used in the trasnsformation to a SAM or EBAM object. If TRUE, it is expected that fit is the output of eBayes. Otherwise, fit can be the result of lmFit or eBayes.
sam.control	further arguments for the SAM analysis. See samControl for these arguments, which should only be changed if they are fully understood.
delta	the minimum posterior probability for a gene to be called differentially expressed (or more generally, for a variable to be called significant) in an EBAM analysis. For details, see ebam. Please note that the meaning of delta differs substantially between sam and ebam
ebam.control	further arguments for an EBAM analysis. See ebamControl for these arguments, which should only be changed if their meaning is fully understood.

Value

An object of class SAM or EBAM.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment. *JASA*, 96, 1151-1160.

Smyth, G.K. (2004). Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology*, 3(1), Article 3.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance Analysis of Microarrays Applied to the Ionizing Radiation Response. *PNAS*, 98, 5116-5121.

See Also

sam, ebam, SAM-class, EBAM-class, samControl, ebamControl

---

link.genes *Links for a list of genes*

Description

Generates a htmlpage with links to several public repositories for a list of genes.

Usage

```
link.genes(genenames, filename, entrez = TRUE, refseq = TRUE, symbol = TRUE,
  omim = FALSE, ug = FALSE, fullname = FALSE, which.refseq = "NM",
  chipname = "", cdfname = NULL, refsnp = NULL, max.associated = 2,
  dataframe = NULL, title = NULL, bg.col = "white", text.col = "black",
  link.col = "blue", tableborder = 1, new.window = TRUE, load = TRUE)
```

Arguments

genenames	a character vector containing the names of the interesting genes.
filename	a character string naming the file in which the output should be stored. Must have the suffix ".html".
entrez	logical indicating if Entrez links should be added to the output.
refseq	logical indicating if RefSeq links should be added to the output.
symbol	logical indicating if the gene symbols should be added to the output.
omim	logical indicating if OMIM links should be added to the output.
ug	logical indicating if UniGene links should be added to the output.
fullname	logical indicating whether the full gene names should be added to the output
which.refseq	character string or vector naming the first two letters of the RefSeq links that should be displayed in the html file.

chipname	character string specifying the chip type used in the analysis. Must be specified as in the metadata section of Bioconductor (e.g., "hgu133a" for the Affymetrix HG-U133A chip). Needs not to be specified if cdfname is specified. For Affymetrix SNP chips (starting with the 500k array set), chipname can be specified by the metadata package name, i.e.\ either by "pd.genomewidesnp.5", by "pd.genomewidesnp.6", by "pd.mapping250k.nsp", or by "pd.mapping250k.sty", to add links to the Affymetrix webpage of the SNPs to the html output.
cdfname	character string specifying the cdf name of the used chip. Must exactly follow the nomenclatur of the Affymetrix chips (e.g., "HG-U133A" for the Affymetrix HG-U133A chip). If specified, links to the Affymetrix webpage for the interesting genes will be added to the output. If SNP chips are considered, chipname instead of cdfname must be specified for obtaining these links.
refsnp	either a character vector or a data frame. If the former, refsnp containis the RefSNP IDs of the SNPs used in the SAM/EBAM analysis, where names(refsnp) specifies the names of these SNPs, i.e.\ their probe set IDs. If a data frame, then one column of refsnp must contain the RefSNP IDs of the SNPs, and the name of this column must be RefSNP. The other columns can contain additional annotations such as the chromosome or the physical position of each SNPs. The row names of refsnp must specify the SNPs, i.e.\ must be the probe set IDs of the SNPs. Using buildSNPannotation from the package **scrime** such a data frame can be generated automatically from the metadata package corresponding to the considered SNP chip.
max.associated	integer specifying the maximum number of genes associated with the respective SNP displayed in the html output. If all entries should be shown, set max.associated = 0. This however might result in a very large html output. For details, see shortenGeneDescription in the package **scrime**.
dataframe	data frame having one row for each interesting gene, i.e.\ nrow(dataframe) must be equal to length(genenames). The row names of dataframe must be equal to genenames. This matrix contains additional information on the list of genes that should be added to the output. If NULL (default) no information will be added to the link list.
title	character string naming the title that should be used in the html page.
bg.col	specification of the background color of the html page. See ?par for how colors can be specified.
text.col	specification of the color of the text used in the html page. See ?par for how colors can be specified.
link.col	specification of the color of the links used in the html file. See ?par for how colors can be specified.
tableborder	integer specifying the thickness of the border of the table.
new.window	logical indicating if the links should be opened in a new window.
load	logical value indicating whether to attempt to load the required annotation data package if it is not already loaded. For details, see the man page of lookUp in the package **annotate**.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

SAM-class, sam, link.siggenes, sam2html

---

link.siggenes *Links for a SAM or an EBAM object*

Description

Generates a html page with links to several public repositories for a list of genes called differentially expressed when using a specific Delta value in a SAM or an EBAM analysis.

Usage

```
link.siggenes(object, delta, filename, gene.names = NULL, addDataFrame = TRUE,
      entrez = TRUE, refseq = TRUE, symbol = TRUE, omim = FALSE, ug = FALSE,
      fullname = FALSE, which.refseq = "NM", chipname = "", cdfname = NULL,
      refsnp = NULL, max.associated = 2, n.digits = 3, title = NULL,
     bg.col = "white", text.col = "black", link.col = "blue", tableborder = 1,
      new.window = TRUE, load = TRUE)
```

Arguments

object	a SAM or an EBAM object.
delta	a numerical value specifying the Delta value.
filename	character string naming the file in which the output should be stored. Must have the suffix ".html".
gene.names	a character vector of the same length as object@d (or object@z) containing the names of the genes. Must only be specified if it is not specified in object, i.e. if it has not been specified in sam (or ebam).
addDataFrame	logical indicating if gene-specific information on the differentially expressed genes should be added to the output.
entrez	logical indicating if Entrez links should be added to the output.
refseq	logical indicating if RefSeq links should be added to the output.
symbol	logical indicating if the gene symbols should be added to the output.
omim	logical indicating if OMIM links should be added to the output.
ug	logical indicating if UniGene links should be added to the output.
fullname	logical indicating whether the full gene names should be added to the output.
which.refseq	character string or vector naming the first two letters of the RefSeq links that should be displayed in the html file.

chipname	character string specifying the chip type used in the analysis. Must be specified as in the meta-data section of Bioconductor (e.g., "hgu133a" for the Affymetrix HG-U133A chip). Needs not to be specified if cdfname is specified. For Affymetrix SNP chips (starting with the 500k array set), chipname can be specified by the metadata package name, i.e.\ either by "pd.genomewidesnp.5", by "pd.genomewidesnp.6", by "pd.mapping250k.nsp", or by "pd.mapping250k.sty", to add links to the Affymetrix webpage of the SNPs to the html output.
cdfname	character string specifying the cdf name of the used chip. Must exactly follow the nomenclatur of the Affymetrix chips (e.g., "HG-U133A" for the Affymetrix HG-U133A chip). If specified, links to the Affymetrix webpage for the interesting genes will be added to the output. If SNP chips are considered, chipname instead of cdfname must be specified for obtaining these links.
refsnp	either a character vector or a data frame. If the former, refsnp containis the RefSNP IDs of the SNPs used in the SAM/EBAM analysis, where names(refsnp) specifies the names of these SNPs, i.e.\ their probe set IDs. If a data frame, then one column of refsnp must contain the RefSNP IDs of the SNPs, and the name of this column must be RefSNP. The other columns can contain additional annotations such as the chromosome or the physical position of each SNPs. The row names of refsnp must specify the SNPs, i.e.\ must be the probe set IDs of the SNPs. Using buildSNPannotation from the package **scrime** such a data frame can be generated automatically from the metadata package corresponding to the considered SNP chip.
max.associated	integer specifying the maximum number of genes associated with the respective SNP displayed in the html output. If all entries should be shown, set max.associated = 0. This however might result in a very large html output. For details, see shortenGeneDescription in the package **scrime**.
n.digits	integer specifying the number of decimal places used in the output.
title	character string naming the title that should be used in the html page.
bg.col	specification of the background color of the html page. See ?par for how colors can be specified.
text.col	specification of the color of the text used in the html page. See ?par for how colors can be specified.
link.col	specification of the color of the links used in the html file. See ?par for how colors can be specified.
tableborder	integer specifying the thickness of the border of the table.
new.window	logical indicating if the links should be opened in a new window.
load	logical value indicating whether to attempt to load the required annotation data package if it is not already loaded. For details, see the man page of lookUp in the package **annotate**.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

sam, ebam, link.genes, sam2html, ebam2html

`list.siggenes` *List of the significant genes*

Description

Lists the genes called differentially expressed by the SAM or the EBAM analysis for a specified value of the threshold Δ.

Usage

```
list.siggenes(object, delta, file = "", gene.names = NULL, order = TRUE,
text = NULL, append = FALSE)
```

Arguments

`object`	either a SAM- or an EBAM-object.
`delta`	a numeric value specifying the threshold Δ in the SAM or EBAM analysis. Note that the meaning of Δ differs between SAM and EBAM: In SAM, it is a strictly positive value, whereas in EBAM it is a probability.
`file`	a character string naming a file in which the output is stored. If `""`, the significant genes will be shown in the console.
`gene.names`	a character vector containing the names of the genes. Needs only to be specified, if the gene names were not specified in `sam` or `ebam`, respectively.
`order`	if `TRUE`, the gene names will be ordered by their "significance".
`text`	a character string specifying the heading of the gene list. By default, the header specifies the type of analysis and the used value of Δ. To avoid a header, set `text = ""`.
`append`	If `TRUE`, the output will be appended to `file`. If `FALSE`, any existing file having the name `file` will be destroyed.

Value

A list of significant genes either shown in the console or stored in a file.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

`sam`, `ebam`

Examples

```
## Not run:
  # Load the package multtest and the data of Golub et al. (1999)
  # contained in \pkg{multtest}.
  library(multtest)
  data(golub)

  # Perform a SAM analysis.
  sam.out<-sam(golub, golub.cl, B=100, rand=123)

  # List the genes called significant by SAM using Delta = 3.1.
  list.siggenes(sam.out, 3.1, gene.names=golub.gnames[,2])

## End(Not run)
```

`md.plot` *MD Plot*

Description

Generates an MD plot for a specified value of Delta.

Contrary to a SAM plot in which the observed values of the test statistic D are plotted against the expected ones, the difference M between the observed and the expected values are plotted against the observed values in an MD plot.

Usage

```
md.plot(object, delta, pos.stats = 1, sig.col = 3, xlim = NULL, ylim = NULL,
      main = NULL, xlab = NULL, ylab = NULL, xsym = NULL, ysym = NULL,
      forceDelta = FALSE, includeZero = TRUE, lab = c(10, 10, 7), pch = NULL,
      sig.cex = 1, ...)
```

Arguments

`object`	an object of class SAM.
`delta`	a numeric value specifying the value of Δ for which the SAM plot should be generated.
`pos.stats`	an integer between 0 and 2. If `pos.stats = 1`, general information as the number of significant genes and the estimated FDR for the specified value of `delta` will be plotted in the upper left corner of the plot. If `pos.stats = 2`, these information will be plotted in the lower right corner. If `pos.stats = 0`, no information will be plotted.
`sig.col`	a specification of the color of the significant genes. If `sig.col` has length 1, all the points corresponding to significant genes are marked in the color specified by `sig.col`. If `length(sig.col) == 2`, the down-regulated genes, i.e. the genes with negative expression score d, are marked in the color specified

by sig.col[1], and the up-regulated genes, i.e. the genes with positive d, are marked in the color specified by sig.col[2]. For a description of how colors are specified, see par.

xlim — a numeric vector of length 2 specifying the x limits (minimum and maximum) of the plot.

ylim — a numeric vector of length 2 specifying the y limits of the plot.

main — a character string naming the main title of the plot.

xlab — a character string naming the label of the x axis.

ylab — a character string naming the label of the y axis.

xsym — should the range of the plotted x-axis be symmetric about the origin? Ignored if xlim is specified. If NULL, xsym will be set to TRUE, if some of the observed values of the test statistic are negative. Otherwise, xsym will be set to FALSE.

ysym — should the range of the plotted y-axis be symmetric about the origin? Ignored if ylim is specified.If NULL, ysym will be set to TRUE, if some of the observed values of the test statistic are negative. Otherwise, ysym will be set to FALSE.

forceDelta — should the two horizontal lines at delta and -delta be within the plot region, no matter whether they are out of the range of the observed d values? Ignored if ylim is specified.

includeZero — should $D = 0$ and $M = 0$ be included in the plot, although all observed values of D (or M) are larger than zero?

lab — a numeric vector of length 3 specifying the approximate number of tickmarks on the x axis and on the y axis and the label size.

pch — either an integer specifying a symbol or a single character to be used as the default in plotting points. For a description of how pch can be specified, see par.

sig.cex — a numerical value giving the amount by which the symbols of the significant genes should be scaled relative to the default.

... — further graphical parameters. See plot.default and par.

Value

A MD plot.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

sam, sam.plot2

Examples

```
## Not run:
  # Load the package multtest and the data of Golub et al. (1999)
  # contained in multtest.
  library(multtest)
  data(golub)

  # Perform a SAM analysis for the two class unpaired case assuming
  # unequal variances.
  sam.out <- sam(golub, golub.cl, B=100, rand=123)

  # Generate a SAM plot for Delta = 2
  plot(sam.out, 2)

  # As an alternative, the MD plot can be generated.
  md.plot(sam.out, 2)


## End(Not run)
```

`nclass.wand` *Number of cells in a histogram*

Description

Computes the number of cells in a histogram using the method of Wand (1994).

Usage

```
nclass.wand(x, level = 1)
```

Arguments

`x`	numeric vector of observations.
`level`	integer specifying the number of levels of functional estimation used in the estimation. For details, see the help page of `dpih` from the package **KernSmooth**.

Details

`nclass.wand` calls `dpih`, and then computes the number of cells corresponding to the optimal bin width returned by `dpih`.

Value

A numeric value specifying the number of cells for the histogram of `x`.

References

Wand, M.P. (1997). Data-based choice of histogram bin width. *American Statistician*, 51, 59–64.

See Also

densprr

---

pi0.est *Estimation of the prior probability*

Description

Estimates the prior probability that a gene is not differentially expressed by the natural cubic splines based method of Storey and Tibshirani (2003).

Usage

```
pi0.est(p, lambda = seq(0, 0.95, 0.05), ncs.value = "max",
    ncs.weights = NULL)
```

Arguments

p	a numeric vector containing the p-values of the genes.
lambda	a numeric vector or value specifying the λ values used in the estimation of the prior probability.
ncs.value	a character string. Only used if lambda is a vector. Either "max" or "paper". For details, see Details.
ncs.weights	a numerical vector of the same length as lambda containing the weights used in the natural cubic spline fit. By default no weights are used.

Details

For each value of lambda, $\pi_0(\lambda)$ is computed by the number of p-values p larger than λ divided by $(1-\lambda)/m$, where m is the length of p.

If lambda is a value, $\pi_0(\lambda)$ is the estimate for the prior probabiltity π_0 that a gene is not differentially expressed.

If lambda is a vector, a natural cubic spline h with 3 degrees of freedom is fitted through the data points $(\lambda, \pi_0(\lambda))$, where each point is weighed by ncs.weights. π_0 is estimated by $h(v)$, where $v = \max\{\lambda\}$ if ncs.value="max", and $v = 1$ if ncs.value="paper".

Value

p0	the estimate of the prior probability that a gene is not differentially expressed.
spline.out	the output of smooth.spline used in this function.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Storey, J.D., and Tibshirani, R. (2003). Statistical Significance for Genome-wide Studies. *PNAS*, 100, 9440-9445.

See Also

SAM-class,sam,qvalue.cal

Examples

```
## Not run:
  # Load the package multtest and the data of Golub et al. (1999)
  # contained in multtest.
  library(multtest)
  data(golub)

  # Perform a SAM analysis.
  sam.out<-sam(golub, golub.cl, B=100, rand=123)

  # Estimate the prior probability that a gene is not significant
  pi0.est(sam.out@p.value)

## End(Not run)
```

---

plotArguments *Plot Arguments*

Description

Utility function for generating a plot of a SAM or an EBAM object in an html output.

Usage

```
plotArguments(pos.stats = NULL, sig.col = 3, xlim = NULL, ylim = NULL,
      main = NULL, xlab = NULL, ylab = NULL, pty = "s", lab = c(10, 10, 7),
      pch = NULL, sig.cex = 1, stats.cex = 0.8, y.intersp = 1.3)
```

Arguments

pos.stats	an integer between 0 and 2 for a SAM plot, and between 0 and 4 for an EBAM plot. See help.sam(plot) or help.ebam(plot), respectively, for how pos.stats can be specified, and for its default.
sig.col	a specification of the color of the significant genes. If sig.col has length 1, all the points corresponding to significant genes are marked in the color specified by sig.col. Only for a SAM plot: If length(sig.col) == 2, the down-regulated genes, i.e. the genes with negative expression score d, are marked in the color specified by sig.col[1], and the up-regulated genes, i.e. the genes with positive d, are marked in the color specified by sig.col[2]. For a description of how colors are specified, see par.

xlim	a numeric vector of length 2 specifying the x limits (minimum and maximum) of the plot.
ylim	a numeric vector of length 2 specifying the y limits of the plot.
main	a character string naming the main title of the plot.
xlab	a character string naming the label of the x axis.
ylab	a character string naming the label of the y axis.
pty	a character specifying the type of plot region to be used. "s" (default for a SAM plot) generates a square plotting region, and "m" (default for an EBAM plot) the maximal plotting region.
lab	a numeric vector of length 3 specifying the approximate number of tickmarks on the x axis and on the y axis and the label size.
pch	either an integer specifying a symbol or a single character to be used as the default in plotting points. For a description of how pch can be specified, see par.
sig.cex	a numerical value giving the amount by which the symbols of the significant genes should be scaled relative to the default.
stats.cex	the size of the statistics printed in the plot relative to the default size. Only available for an EBAM plot.
y.intersp	a numeric value specifying the space between the rows in which the statistics are plotted. Only available for an EBAM plot.

Value

A list required by sam2html or ebam2html if addPlot = TRUE.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

sam2html,ebam2html

---

plotFindArguments *Plot Arguments*

Description

Utility function for generating a plot of the posterior probabilities in an html file when searching for the optimal value of the fudge factor in an EBAM analysis.

Usage

```
plotFindArguments(onlyTab = FALSE, logit = TRUE, pos.legend = NULL,
      legend.cex = 0.8, col = NULL, main = NULL, xlab = NULL, ylab = NULL,
      only.a0 = FALSE, lty = 1, lwd = 1, y.intersp = 1.1)
```

Arguments

onlyTab	if TRUE, then this plot is not generated and only the table of the number of differentially expressed genes and the estimated FDR for the different values of the fudge factor is shown.
logit	should the posterior probabilities be logit-transformed before they are plotted?
pos.legend	an integer between 0 and 4. See help.finda0(plot) for how pos.legend can be specified, and for its default.
legend.cex	the size of the text in the legend relative to the default size
col	a vector specifying the colors of the lines for the different values of the fudge factor. For a description of how colors can be specified, see par.
main	a character string naming the main title of the plot.
xlab	a character string naming the label of the x axis.
ylab	a character string naming the label of the y axis.
only.a0	if TRUE, only the values of a_0 are shown in the legend. If FALSE, both the values of a_0 and the corresponding number of differentially expressed genes are shown.
lty	a value or vector specifying the line type of the curves. For details, see par.
lwd	a numeric value specifying the width of the plotted lines. For details, see par.
y.intersp	a numeric value specifying the space between the rows of the legend.

Value

A list required by ebam2html if findA0 is specified.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

ebam2html

---

qvalue.cal *Computation of the q-value*

Description

Computes the q-values of a given set of p-values.

Usage

```
qvalue.cal(p, p0, version = 1)
```

Arguments

p	a numeric vector containing the p-values.
p0	a numeric value specifying the prior probability that a gene is not differentially expressed.
version	If version=2, the original version of the q-value, i.e. min{pFDR}, will be computed. if version=1, min{FDR} will be used in the computation of the q-value.

Details

Using version = 1 in qvalue.cal corresponds to setting robust = FALSE in the function qvalue of John Storey's R package **qvalue**, while version = 2 corresponds to robust = TRUE.

Value

A vector of the same length as p containing the q-values corresponding to the p-values in p.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Storey, J.D. (2003). The positive False Discovery Rate: A Bayesian Interpretation and the q-value. *Annals of Statistics*, 31, 2013-2035.

Storey, J.D., and Tibshirani, R. (2003). Statistical Significance for Genome-wide Studies. *PNAS*, 100, 9440-9445.

See Also

pi0.est,SAM-class,sam

Examples

```
## Not run:
  # Load the package multtest and the data of Golub et al. (1999)
  # contained in multtest.
  library(multtest)
  data(golub)

  # Perform a SAM analysis.
  sam.out<-sam(golub, golub.cl, B=100, rand=123)

  # Estimate the prior probability that a gene is not significant.
  pi0 <- pi0.est(sam.out@p.value)$p0

  # Compute the q-values of the genes.
  q.value <- qvalue.cal(sam.out@p.value, pi0)

## End(Not run)
```

rowWilcoxon *Rowwise Wilcoxon Rank Sum Statistics*

Description

Computes either the Wilcoxon Rank Sum or Signed Rank Statistics for all rows of a matrix simultaneously.

Usage

```
rowWilcoxon(X, cl, rand = NA)
```

Arguments

X	a matrix in which each row corresponds to a variable, and each column to an observation/sample.
cl	a numeric vector consisting of ones and zeros. The length of cl must be equal to the number of observations. If cl consists of zeros and ones, Wilcoxon Rank Sums are computed. If cl contains only ones, Wilcoxon Signed Rank Statistics are calculated.
rand	Sets the random number generator into a reproducible state. Ignored if Wilcoxon rank sums are computed, or X contains no zeros.

Details

If there are ties, then the ranks of the observations belonging to the same group of tied observations will be set to the maximum rank available for the corresponding group.

Value

A numeric vector containing Wilcoxon rank statistics for each row of X.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

wilc.stat,wilc.ebam

sam *Significance Analysis of Microarray*

Description

Performs a Significance Analysis of Microarrays (SAM). It is possible to perform one and two class analyses using either a modified t-statistic or a (standardized) Wilcoxon rank statistic, and a multiclass analysis using a modified F-statistic. Moreover, this function provides a SAM procedure for categorical data such as SNP data and the possibility to employ an user-written score function.

Usage

```
sam(data, cl, method = d.stat, control=samControl(),
    gene.names = dimnames(data)[[1]], ...)
```

Arguments

data	a matrix, a data frame, or an ExpressionSet object. Each row of data (or exprs(data), respectively) must correspond to a variable (e.g., a gene), and each column to a sample (i.e.\ an observation). Can also be a list (if method = chisq.stat or method = trend.stat). For details on how to specify data in this case, see chisq.stat.
cl	a vector of length ncol(data) containing the class labels of the samples. In the two class paired case, cl can also be a matrix with ncol(data) rows and 2 columns. If data is an ExpressionSet object, cl can also be a character string naming the column of pData(data) that contains the class labels of the samples. If data is a list, cl needs not to be specified. In the one-class case, cl should be a vector of 1's. In the two class unpaired case, cl should be a vector containing 0's (specifying the samples of, e.g., the control group) and 1's (specifying, e.g., the case group). In the two class paired case, cl can be either a numeric vector or a numeric matrix. If it is a vector, then cl has to consist of the integers between -1 and $-n/2$ (e.g., before treatment group) and between 1 and $n/2$ (e.g., after treatment group), where n is the length of cl and k is paired with $-k$, $k = 1, \ldots, n/2$. If cl is a matrix, one column should contain -1's and 1's specifying, e.g., the before and the after treatment samples, respectively, and the other column should contain integer between 1 and $n/2$ specifying the $n/2$ pairs of observations. In the multiclass case and if method = chisq.stat, cl should be a vector containing integers between 1 and g, where g is the number of groups. (In the case of chisq.stat, cl needs not to be specified if data is a list of groupwise matrices.) For examples of how cl can be specified, see the manual of **siggenes**.
method	a character string or a name specifying the method/function that should be used in the computation of the expression scores d. If method = d.stat, a modified t-statistic or F-statistic, respectively, will be computed as proposed by Tusher et al. (2001).

If method = wilc.stat, a Wilcoxon rank sum statistic or Wilcoxon signed rank statistic will be used as expression score.

For an analysis of categorical data such as SNP data, method can be set to chisq.stat. In this case Pearson's ChiSquare statistic is computed for each row.

If the variables are ordinal and a trend test should be applied (e.g., in the two-class case, the Cochran-Armitage trend test), method = trend.stat can be employed.

It is also possible to use an user-written function to compute the expression scores. For details, see Details.

control — further optional arguments for controlling the SAM analysis. For these arguments, see samControl.

gene.names — a character vector of length nrow(data) containing the names of the genes. By default the row names of data are used.

... — further arguments of the specific SAM methods. If method = d.stat, see the help of d.stat. If method = wilc.stat, see the help of wilc.stat. If method = chisq.stat, see the help of chisq.stat.

Details

sam provides SAM procedures for several types of analysis (one and two class analyses with either a modified t-statistic or a Wilcoxon rank statistic, a multiclass analysis with a modified F statistic, and an analysis of categorical data). It is, however, also possible to write your own function for another type of analysis. The required arguments of this function must be data and cl. This function can also have other arguments. The output of this function must be a list containing the following objects:

d: a numeric vector consisting of the expression scores of the genes.

d.bar: a numeric vector of the same length as na.exclude(d) specifying the expected expression scores under the null hypothesis.

p.value: a numeric vector of the same length as d containing the raw, unadjusted p-values of the genes.

vec.false: a numeric vector of the same length as d consisting of the one-sided numbers of falsely called genes, i.e. if $d > 0$ the numbers of genes expected to be larger than d under the null hypothesis, and if $d < 0$, the number of genes expected to be smaller than d under the null hypothesis.

s: a numeric vector of the same length as d containing the standard deviations of the genes. If no standard deviation can be calculated, set s = numeric(0).

s0: a numeric value specifying the fudge factor. If no fudge factor is calculated, set s0 = numeric(0).

mat.samp: a matrix with B rows and ncol(data) columns, where B is the number of permutations, containing the permutations used in the computation of the permuted d-values. If such a matrix is not computed, set mat.samp = matrix(numeric(0)).

msg: a character string or vector containing information about, e.g., which type of analysis has been performed. msg is printed when the function print or summary, respectively, is called. If no such message should be printed, set msg = "".

`fold`: a numeric vector of the same length as `d` consisting of the fold changes of the genes. If no fold change has been computed, set `fold = numeric(0)`.

If this function is, e.g., called `foo`, it can be used by setting `method = foo` in `sam`. More detailed information and an example will be contained in the siggenes manual.

Value

An object of class SAM.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Schwender, H., Krause, A., and Ickstadt, K. (2006). Identifying Interesting Genes with siggenes. *RNews*, 6(5), 45-50.

Schwender, H. (2004). Modifying Microarray Analysis Methods for Categorical Data – SAM and PAM for SNPs. To appear in: *Proceedings of the the 28th Annual Conference of the GfKl*.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116-5121.

See Also

SAM-class,d.stat,wilc.stat, chisq.stat, samControl

Examples

```
## Not run:
  # Load the package multtest and the data of Golub et al. (1999)
  # contained in multtest.
  library(multtest)
  data(golub)

  # golub.cl contains the class labels.
  golub.cl

  # Perform a SAM analysis for the two class unpaired case assuming
  # unequal variances.
  sam.out <- sam(golub, golub.cl, B=100, rand=123)
  sam.out

  # Obtain the Delta plots for the default set of Deltas
  plot(sam.out)

  # Generate the Delta plots for Delta = 0.2, 0.4, 0.6, ..., 2
  plot(sam.out, seq(0.2, 0.4, 2))

  # Obtain the SAM plot for Delta = 2
  plot(sam.out, 2)
```

```
  # Get information about the genes called significant using
  # Delta = 3.
  sam.sum3 <- summary(sam.out, 3, entrez=FALSE)

  # Obtain the rows of golub containing the genes called
  # differentially expressed
  sam.sum3@row.sig.genes

  # and their names
  golub.gnames[sam.sum3@row.sig.genes, 3]

  # The matrix containing the d-values, q-values etc. of the
  # differentially expressed genes can be obtained by
  sam.sum3@mat.sig

  # Perform a SAM analysis using Wilcoxon rank sums
  sam(golub, golub.cl, method="wilc.stat", rand=123)


  # Now consider only the first ten columns of the Golub et al. (1999)
  # data set. For now, let's assume the first five columns were
  # before treatment measurements and the next five columns were
  # after treatment measurements, where column 1 and 6, column 2
  # and 7, ..., build a pair. In this case, the class labels
  # would be
  new.cl <- c(-(1:5), 1:5)
  new.cl

  # and the corresponding SAM analysis for the two-class paired
  # case would be
  sam(golub[,1:10], new.cl, B=100, rand=123)

  # Another way of specifying the class labels for the above paired
  # analysis is
  mat.cl <- matrix(c(rep(c(-1, 1), e=5), rep(1:5, 2)), 10)
  mat.cl

  # and the above SAM analysis can also be done by
  sam(golub[,1:10], mat.cl, B=100, rand=123)

## End(Not run)
```

SAM-class *Class SAM*

Description

This is a class representation for several versions of the SAM (Significance Analysis of Microarrays) procedure proposed by Tusher et al. (2001).

Objects from the Class

Objects can be created using the functions sam, sam.dstat, sam.wilc and sam.snp.

Slots

d: Object of class "numeric" representing the expression scores of the genes.

d.bar: Object of class "numeric" representing the expected expression scores under the null hypothesis.

vec.false: Object of class "numeric" containing the one-sided expected number of falsely called genes.

p.value: Object of class "numeric" consisting of the p-values of the genes.

s: Object of class "numeric" representing the standard deviations of the genes. If the standard deviations are not computed, s will be set to numeric(0).

s0: Object of class "numeric" representing the value of the fudge factor. If not computed, s0 will be set to numeric(0).

mat.samp: Object of class "matrix" containing the permuted group labels used in the estimation of the null distribution. Each row represents one permutation, each column one observation (pair). If no permutation procedure has been used, mat.samp will be set to matrix(numeric(0)).

p0: Object of class "numeric" representing the prior probability that a gene is not differentially expressed.

mat.fdr: Object of class "matrix" containing general information as the number of significant genes and the estimated FDR for several values of Δ. Each row represents one value of Δ, each of the 9 columns one statistic.

q.value: Object of class "numeric" consisting of the q-values of the genes. If not computed, q.value will be set to numeric(0).

fold: Object of class "numeric" representing the fold changes of the genes. If not computed, fold will be set to numeric(0).

msg: Object of class "character" containing information about, e.g., the type of analysis. msg is printed when the functions print and summary, respectively, are called.

chip: Object of class "character" naming the microarray used in the analysis. If no information about the chip is available, chip will be set to "".

Methods

identify signature(x = "SAM"): After generating a SAM plot, identify can be used to obtain information about the genes by clicking on the symbols in the SAM plot. For details, see help.sam(identify). Arguments are listed by args.sam(identify).

plot signature(x = "SAM"): Generates a SAM plot or the Delta plots. If the specified delta in plot(object,delta) is a numeric value, a SAM plot will be generated. If delta is either not specified or a numeric vector, the Delta plots will be generated. For details, see ?sam.plot2, ?delta.plot or help.sam(plot),respectively. Arguments are listed by args.sam(plot).

print signature(x = "SAM"): Prints general information such as the number of significant genes and the estimated FDR for a set of Δ. For details, see help.sam(print). Arguments are listed by args.sam(print).

show `signature(object = "SAM")`: Shows the output of the SAM analysis.

summary `signature(object = "SAM")`: Summarizes the results of a SAM analysis. If `delta` in `summary(object,delta)` is not specified or a numeric vector, the information shown by print and some additional information will be shown. If `delta` is a numeric vector, the general information for the specific Δ is shown and additionally gene-specific information about the genes called significant using this value of Δ. The output of summary is an object of class sumSAM which has the slots `row.sig.genes`, `mat.fdr`, `mat.sig` and `list.args`. For details, see `help.sam(summary)`. All arguments are listed by `args.sam(summary)`.

Note

SAM was developed by Tusher et al. (2001).

!!! There is a patent pending for the SAM technology at Stanford University. !!!

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Schwender, H., Krause, A. and Ickstadt, K. (2003). Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany. http://www.sfb475.uni-dortmund.de/berichte/tr44-03.pdf.

Schwender, H. (2004). Modifying Microarray Analysis Methods for Categorical Data – SAM and PAM for SNPs. To appear in: *Proceedings of the the 28th Annual Conference of the GfKl*.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116-5121.

See Also

`sam`,`args.sam`,`sam.plot2`, `delta.plot`

Examples

```
## Not run:
  # Load the package multtest and the data of Golub et al. (1999)
  # contained in multtest.
  library(multtest)
  data(golub)

  # Perform a SAM analysis for the two class unpaired case assuming
  # unequal variances.
  sam.out <- sam(golub, golub.cl, B=100, rand=123)
  sam.out

  # Alternative ways to show the output of sam.
  show(sam.out)
  print(sam.out)

  # Obtain a little bit more information.
```

```
summary(sam.out)

# Print the results of the SAM analysis for other values of Delta.
print(sam.out, seq(.2, 2, .2))

# Again, the same with additional information.
summary(sam.out, seq(.2, 2, .2))

# Obtain the Delta plots for the default set of Deltas.
plot(sam.out)

# Generate the Delta plots for Delta = 0.2, 0.4, 0.6, ..., 2.
plot(sam.out, seq(0.2, 0.4, 2))

# Obtain the SAM plot for Delta = 2.
plot(sam.out, 2)

# Get information about the genes called significant using
# Delta = 3.
sam.sum3 <- summary(sam.out, 3)
sam.sum3

# Obtain the rows of the Golub et al. (1999) data set containing
# the genes called differentially expressed
sam.sum3@row.sig.genes

# and their names
golub.gnames[sam.sum3@row.sig.genes, 3]

# The matrix containing the d-values, q-values etc. of the
# differentially expressed genes can be obtained by
sam.sum3@mat.sig

## End(Not run)
```

`sam.plot2` *SAM Plot*

Description

Generates a SAM plot for a specified value of Delta.

Usage

```
sam.plot2(object, delta, pos.stats = NULL, sig.col = 3, xlim = NULL,
      ylim = NULL, main = NULL, xlab = NULL, ylab = NULL, pty = "s",
      lab = c(10, 10, 7), pch = NULL, sig.cex = 1, ...)
```

Arguments

object	an object of class SAM.
delta	a numeric value specifying the value of Δ for which the SAM plot should be generated.
pos.stats	an integer between 0 and 2. If pos.stats = 1, general information as the number of significant genes and the estimated FDR for the specified value of delta will be plotted in the upper left corner of the plot. If pos.stats = 2, these information will be plotted in the lower right corner. If pos.stats = 0, no information will be plotted. By default, pos.stats = 1 if the expression score d can be both positive and negative, and pos.stats = 2 if d can only take positive values.
sig.col	a specification of the color of the significant genes. If sig.col has length 1, all the points corresponding to significant genes are marked in the color specified by sig.col. If length(sig.col) == 2, the down-regulated genes, i.e. the genes with negative expression score d, are marked in the color specified by sig.col[1], and the up-regulated genes, i.e. the genes with positive d, are marked in the color specified by sig.col[2]. For a description of how colors are specified, see par.
xlim	a numeric vector of length 2 specifying the x limits (minimum and maximum) of the plot.
ylim	a numeric vector of length 2 specifying the y limits of the plot.
main	a character string naming the main title of the plot.
xlab	a character string naming the label of the x axis.
ylab	a character string naming the label of the y axis.
pty	a character specifying the type of plot region to be used. "s" (default) generates a square plotting region, and "m" the maximal plotting region.
lab	a numeric vector of length 3 specifying the approximate number of tickmarks on the x axis and on the y axis and the label size.
pch	either an integer specifying a symbol or a single character to be used as the default in plotting points. For a description of how pch can be specified, see par.
sig.cex	a numerical value giving the amount by which the symbols of the significant genes should be scaled relative to the default.
...	further graphical parameters. See plot.default and par.

Value

A SAM plot.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116-5121.

See Also

SAM-class,sam,md.plot

Examples

```
## Not run:
  # Load the package multtest and the data of Golub et al. (1999)
  # contained in multtest.
  library(multtest)
  data(golub)

  # Perform a SAM analysis for the two class unpaired case assuming
  # unequal variances.
  sam.out <- sam(golub, golub.cl, B=100, rand=123)

  # Generate a SAM plot for Delta = 2
  sam.plot2(sam.out, 2)

  # Alternatively way of generating the same SAM plot
  plot(sam.out, 2)

  # As an alternative, the MD plot can be generated.
  md.plot(sam.out, 2)


## End(Not run)
```

---

samControl *Further SAM Arguments*

Description

Specifies most of the optional arguments of sam.

Usage

```
samControl(delta = NULL, n.delta = 10, p0 = NA, lambda = seq(0, 0.95, 0.05),
   ncs.value = "max", ncs.weights = NULL, q.version = 1)
```

Arguments

delta	a numeric vector specifying a set of values for the threshold Δ that should be used. If NULL, n.delta Δ values will be computed automatically.
n.delta	a numeric value specifying the number of Δ values that will be computed over the range of all possible values for Δ if delta is not specified.
p0	a numeric value specifying the prior probability π_0 that a gene is not differentially expressed. If NA, p0 will be computed by the function pi0.est.
lambda	a numeric vector or value specifying the λ values used in the estimation of the prior probability. For details, see pi0.est.
ncs.value	a character string. Only used if lambda is a vector. Either "max" or "paper". For details, see pi0.est.
ncs.weights	a numerical vector of the same length as lambda containing the weights used in the estimation of π_0. By default no weights are used. For details, see ?pi0.est.
q.version	a numeric value indicating which version of the q-value should be computed. If q.version = 2, the original version of the q-value, i.e. min{pFDR}, will be computed. If q.version = 1, min{FDR} will be used in the calculation of the q-value. Otherwise, the q-value is not computed. For details, see qvalue.cal.

Details

These parameters should only be changed if they are fully understood.

Value

A list containing the values of the parameters that are used in sam.

Author(s)

Holger Schwender, <holger.schwender@udo.edu>

References

Schwender, H., Krause, A., and Ickstadt, K. (2006). Identifying Interesting Genes with siggenes. *RNews*, 6(5), 45-50.

Storey, J.D. and Tibshirani, R. (2003). Statistical Significance for Genome-Wide Studies. *Proceedings of the National Academy of Sciences*, 100, 9440-9445.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116-5121.

See Also

limma2sam, sam

siggenes-internal *Internal siggenes functions*

Description

Internal siggenes functions.

Details

These functions are not meant to be directly called by the user.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

siggenes2excel *CSV file of a SAM or an EBAM object*

Description

Generates a csv file for either a SAM or an EBAM object for the use in Excel. This csv file can contain general information as the number of differentially expressed genes and the estimated FDR, and gene-specific information on the differentially expressed genes.

Usage

```
sam2excel(object, delta, file, excel.version=1, n.digits = 3, what = "both",
       entrez = FALSE, chip = "", quote = FALSE)

ebam2excel(object, delta, file, excel.version=1, n.digits = 4, what = "both",
       entrez = FALSE, chip = "", quote = FALSE)
```

Arguments

object	either a SAM or an EBAM object.
delta	a numerical value specifying the Delta value.
file	character string naming the file in which the output should be stored. Must have the suffix ".csv".
excel.version	either 1 or 2. If excel.version=1 (default) a csv file for the use in an Excel version with American standard settings (sep="," and dec=".") will be generated. If excel.version=2 a csv file for the European standard setting (sep=";" and dec=",") will be generated.
n.digits	integer specifying the number of decimal places used in the output.

what	either "both", "stats" or "genes". If "stats" general information will be shown. If "genes" gene-specific information will be given. If "both" both general and gene-specific information will be shown.
entrez	logical indicating if both the Entrez links and the symbols of the genes will be added to the output.
chip	character string naming the chip type used in this analysis. Must be specified as in the meta-data section of Bioconductor (e.g., "hgu133a" for the Affymetrix HG-U133A chip). Only needed if ll = TRUE. If the argument data in sam(data, cl, ...) has been specified by an ExpressionSet object chip need not to be specified.
quote	logical indicating if character strings and factors should be surrounded by double quotes. For details see write.table.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

sam, sam2html, ebam, ebam2html

---

siggenes2html *HTML page for a SAM or an EBAM object*

Description

Generates a html page for a SAM or an EBAM object. This html page can contain general information as the number of differentially expressed genes and the estimated FDR, the SAM or EBAM plot, and gene-specific information on the differentially expressed genes.

Usage

```
ebam2html(object, delta, filename, addStats = TRUE, addPlot = TRUE,
       addGenes = TRUE, findA0 = NULL, varName = NULL, entrez = TRUE,
       refseq = TRUE, symbol = TRUE, omim = FALSE, ug = FALSE,
       fullname = FALSE, chipname = "", cdfname = NULL,
       which.refseq = "NM", refsnp = NULL, max.associated = 2,
       n.digits = 3, bg.col = "white", text.col = "black", link.col = "blue",
       plotArgs = plotArguments(), plotFindArgs = plotFindArguments(),
       bg.plot.adjust = FALSE, plotname = NULL, plotborder = 0,
       tableborder = 1, new.window = TRUE, load = TRUE, ...)

sam2html(object, delta, filename, addStats = TRUE, addPlot = TRUE,
       addGenes = TRUE, varName = NULL, entrez = TRUE, refseq = TRUE,
       symbol = TRUE, omim = FALSE, ug = FALSE, fullname = FALSE,
       bonf = FALSE, chipname = "", cdfname = NULL, which.refseq = "NM",
```

```
refsnp = NULL, max.associated = 2, n.digits = 3, bg.col = "white",
text.col = "black", link.col = "blue", plotArgs = plotArguments(),
bg.plot.adjust = FALSE, plotname = NULL, plotborder = 0,
tableborder = 1, new.window = TRUE, load = TRUE, ...)
```

Arguments

object	a SAM or an EBAM object.
delta	a numerical value specifying the Delta value.
filename	character string naming the file in which the output should be stored. Must have the suffix ".html".
addStats	logical indicating if general information as the number of differentially expressed genes and the estimated FDR should be added to the html page.
addPlot	logical indicating if the SAM/EBAM plot should be added to the html page
addGenes	logical indicating if gene-specific information on the differentially expressed genes should be added to the html page.
findA0	an object of class FindA0. If specified, the numbers of differentially expressed genes and the estimated FDRs for the different possible values of the fudge factor and the corresponding plot of the logit-transformed posterior probabilities are included in the html file.
varName	character string indicating how the variables should be named. If NULL, the variables will be referred to as SNPs in the output if method = cat.stat, and as Genes otherwise.
entrez	logical indicating if Entrez links should be added to the output. Ignored if addGenes = FALSE.
refseq	logical indicating if RefSeq links should be added to the output. Ignored if addGenes = FALSE.
symbol	logical indicating if the gene symbols should be added to the output. Ignored if addGenes = FALSE.
omim	logical indicating if OMIM links should be added to the output. Ignored if addGenes = FALSE.
ug	logical indicating if UniGene links should be added to the output. Ignored if addGenes = FALSE.
fullname	logical indicating whether the full gene names should be added to the output. Ignored if addGenes = FALSE.
bonf	logical indicating whether Bonferroni adjusted p-values should be added to the output. Ignored if addGenes = FALSE.
chipname	character string specifying the chip type used in the analysis. Must be specified as in the meta-data section of Bioconductor (e.g., "hgu133a" for the Affymetrix HG-U133A chip). Needs not to be specified if cdfname is specified. For Affymetrix SNP chips (starting with the 500k array set), chipname can be specified by the metadata package name, i.e.\ either by "pd.genomewidesnp.5", by "pd.genomewidesnp.6", by "pd.mapping250k.nsp", or by "pd.mapping250k.sty", to add links to the Affymetrix webpage of the SNPs to the html output. Ignored if addGenes = FALSE.

`cdfname` character string specifying the cdf name of the used chip. Must exactly follow the nomenclatur of the Affymetrix chips (e.g., `"HG-U133A"` for the Affymetrix HG-U133A chip). If specified, links to the Affymetrix webpage for the interesting genes will be added to the output. If SNP chips are considered, `chipname` instead of `cdfname` must be specified for obtaining these links. Ignored if `addGenes = FALSE`.

`which.refseq` character string or vector naming the first two letters of the RefSeq links that should be displayed in the html file.

`refsnp` either a character vector or a data frame. If the former, `refsnp` containis the RefSNP IDs of the SNPs used in the SAM/EBAM analysis, where `names(refsnp)` specifies the names of these SNPs, i.e.\ their probe set IDs. If a data frame, then one column of `refsnp` must contain the RefSNP IDs of the SNPs, and the name of this column must be `RefSNP`. The other columns can contain additional annotations such as the chromosome or the physical position of each SNPs. The row names of `refsnp` must specify the SNPs, i.e.\ must be the probe set IDs of the SNPs. Using `buildSNPannotation` from the package **scrime** such a data frame can be generated automatically from the metadata package corresponding to the considered SNP chip.

`max.associated` integer specifying the maximum number of genes associated with the respective SNP displayed in the html output. If all entries should be shown, set `max.associated = 0`. This however might result in a very large html output. For details, see `shortenGeneDescription` in the package `scrime`.

`n.digits` integer specifying the number of decimal places used in the output.

`bg.col` specification of the background color of the html page. See `par` for how colors can be specified.

`text.col` specification of the color of the text used in the html page. See `par` for how colors can be specified.

`link.col` specification of the color of the links used in the html file. See `par` for how colors can be specified.

`plotArgs` further arguments for generating the SAM/EBAM plot. These are the arguments used by the SAM/EBAM specific `plot` method. See the help of `plotArguments` for these arguments. Ignored if `addPlot = FALSE`.

`plotFindArgs` further arguments for generating the (logit-transformed) posterior probabilities for the different values of the fudge factor. Ignored if `findA0 = NULL`. See the help of `plotFindArguments` for these arguments.

`bg.plot.adjust` logical indicating if the background color of the SAM plot should be the same as the background color of the html page. If `FALSE` (default) the background of the plot is white. Ignored if `addPlot = FALSE`.

`plotname` character string naming the file in which the SAM/EBAM plot is stored. This file is needed when the SAM/EBAM plot should be added to the html page. If not specified the SAM/EBAM plot will be stored as png file in the same folder as the html page. Ignored if `addPlot = FALSE`.

`plotborder` integer specifying the thickness of the border around the plot. By default, `plotborder = 0`, i.e.\ no border is drawn around the plot. Ignored if `addPlot = FALSE`.

tableborder	integer specifying the thickness of the border of the table. Ignored if addGenes = FALSE.
new.window	logical indicating if the links should be opened in a new window.
load	logical value indicating whether to attempt to load the required annotation data package if it is not already loaded. For details, see the man page of lookUp in the package **annotate**.
...	further graphical arguments for the SAM/EBAM plot. See plot.default and par. Ignored if addPlot = FALSE.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

SAM-class, sam, EBAM-class, ebam, link.genes, link.siggenes, plotArguments, plotFindArguments

---

sumSAM-class *Classes sumSAM and sumEBAM*

Description

These classes are just used for a nicer output of the summary of an object of class SAM or EBAM, respectively.

Objects from the Class

Objects can be created by calls of the form new("sumSAM", ...), or by using the function summary(object) when object is a SAM-class object.

Objects can be created by calls of the form new("sumEBAM", ...), or by using the function summary(object) when object is an EBAM-class object.

Slots

row.sig.genes: Object of class "numeric" consisting of the row numbers of the significant genes in the data matrix.

mat.fdr: Object of class "matrix" containing general information as the number of differentially expressed genes and the estimated FDR for either one or several values of Delta.

mat.sig: Object of class "data.frame" containing gene-specific statistics as the d-values (or z-values) and the q-values or (the local FDR) of the differentially expressed genes.

list.args: Object of class "list" consisting of some of the specified arguments of summary needed for internal use.

Methods

print `signature(x = "sumSAM")`: Prints the output of the SAM-specific method summary.

show `signature(object = "sumSAM")`: Shows the output of the summary of a SAM analysis.

print `signature(x = "sumEBAM")`: Prints the output of the EBAM-specific method summary.

show `signature(object = "sumEBAM")`: Shows the output of the summary of a EBAM analysis.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

See Also

SAM-class, EBAM-class

---

trend.ebam *EBAM Analysis of Linear Trend*

Description

Generates the required statistics for an Empirical Bayes Analysis of Microarrays for a linear trend in (ordinal) data.

In the two-class case, the Cochran-Armitage trend statistic is computed. Otherwise, the statistic for the general test of trend described on page 87 of Agresti (2002) is determined.

Should not be called directly, but via ebam(..., method = trend.ebam).

Usage

```
## Default S3 method:
trend.ebam(data, cl, catt = TRUE, approx = TRUE, n.interval = NULL,
    df.dens = NULL, knots.mode = NULL, type.nclass = "wand",
    B = 100, B.more = 0.1, B.max = 50000, n.subset = 10,
    fast = FALSE, df.ratio = 3, rand = NA, ...)

## S3 method for class 'list'
trend.ebam(data, cl, catt = TRUE, approx = TRUE, n.interval = NULL,
    df.dens = NULL, knots.mode = NULL, type.nclass = "wand", ...)
```

Arguments

data	either a numeric matrix or data frame, or a list. If a matrix or data frame, then each row must correspond to a variable (e.g., a SNP), and each column to a sample (i.e.\ an observation). The values in the matrix or data frame are interpreted as the scores for the different levels of the variables. If the number of observations is huge it is better to specify data as a list consisting of matrices, where each matrix represents one group and summarizes

how many observations in this group show which level at which variable. The row and column names of all matrices must be identical and in the same order. The column names must be interpretable as numeric scores for the different levels of the variables. These matrices can, e.g., be generated using the function `rowTables` from the package **scrime**. (It is recommended to use this function, as `trend.stat` has been made for using the output of `rowTables`.) For details on how to specify this list, see the examples section on this man page, and the help for `rowChisqMultiClass` in the package **scrime**.

`cl` a numeric vector of length `ncol(data)` indicating to which classes the samples in the matrix or data frame `data` belongs. The values in `cl` must be interpretable as scores for the different classes. Must be specified if `data` is a matrix or a data frame, whereas `cl` can but must not be specified if `data` is a list. If specified in the latter case, `cl` must have length `data`, i.e.\ one score for each of the matrices, and thus for each of the groups. If not specified, `cl` will be set to the integers between 1 and c, where c is the number of classes/matrices.

`catt` should the Cochran-Armitage trend statistic be computed in the two-class case? If `FALSE`, the trend statistic described on page 87 of Agresti (2002) is determined which differs by the factor $(n-1)/n$ from the Cochran-Armitage trend statistic.

`approx` should the null distribution be approximated by the χ^2-distribution with one degree of freedom? If `FALSE`, a permutation method is used to estimate the null distribution. If `data` is a list, `approx` must currently be `TRUE`.

`n.interval` the number of intervals used in the logistic regression with repeated observations for estimating the ratio f_0/f (if `approx = FALSE`), or in the Poisson regression used to estimate the density of the observed z-values (if `approx = TRUE`). If `NULL`, `n.interval` is set to 139 if `approx = FALSE`, and estimated by the method specified by `type.nclass` if `approx = TRUE`.

`df.dens` integer specifying the degrees of freedom of the natural cubic spline used in the Poisson regression to estimate the density of the observed z-values. Ignored if `approx = FALSE`. If `NULL`, `df.dens` is set to 3 if the degrees of freedom of the appromimated null distribution, i.e.\ the χ^2-distribution, are less than or equal to 2, and otherwise `df.dens` is set to 5.

`knots.mode` if `TRUE` the `df.dens` - 1 knots are centered around the mode and not the median of the density when fitting the Poisson regression model. Ignored if `approx = FALSE`. If not specified, `knots.mode` is set to `TRUE` if the degrees of freedom of the approximated null distribution, i.e.\ tht χ^2-distribution, are larger than or equal to 3, and otherwise `knots.mode` is set to `FALSE`. For details on this density estimation, see `denspr`.

`type.nclass` character string specifying the procedure used to compute the number of cells of the histogram. Ignored if `approx = FALSE` or `n.interval` is specified. Can be either `"wand"` (default), `"scott"`, or `"FD"`. For details, see `denspr`.

`B` the number of permutations used in the estimation of the null distribution, and hence, in the computation of the expected z-values.

`B.more` a numeric value. If the number of all possible permutations is smaller than or equal to (1+`B.more`)*`B`, full permutation will be done. Otherwise, `B` permutations are used.

B.max	a numeric value. If the number of all possible permutations is smaller than or equal to B.max, B randomly selected permutations will be used in the computation of the null distribution. Otherwise, B random draws of the group labels are used.
n.subset	a numeric value indicating in how many subsets the B permutations are divided when computing the permuted z-values. Please note that the meaning of n.subset differs between the SAM and the EBAM functions.
fast	if FALSE the exact number of permuted test scores that are more extreme than a particular observed test score is computed for each of the variables/SNPs. If TRUE, a crude estimate of this number is used.
df.ratio	integer specifying the degrees of freedom of the natural cubic spline used in the logistic regression with repeated observations. Ignored if approx = TRUE.
rand	numeric value. If specified, i.e. not NA, the random number generator will be set into a reproducible state.
...	ignored.

Value

A list containing statistics required by ebam.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Agresti, A.\ (2002). *Categorical Data Analysis*. Wiley, Hoboken, NJ. 2nd Edition.

Efron, B., Tibshirani, R., Storey, J.D., and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment, *JASA*, 96, 1151-1160.

See Also

EBAM-class,ebam, trend.stat, chisq.ebam

Examples

```
## Not run:
  # Generate a random 1000 x 40 matrix consisting of the values
  # 1, 2, and 3, and representing 1000 variables and 40 observations.

  mat <- matrix(sample(3, 40000, TRUE), 1000)

  # Assume that the first 20 observations are cases, and the
  # remaining 20 are controls, and that the values 1, 2, 3 in
  # mat can be interpreted as scores for the different levels
  # of the variables.

  cl <- rep(1:2, e=20)
```

```
  # Then an EBAM analysis of linear trend can be done by

  out <- ebam(mat, cl, method=trend.ebam)
  out

  # The same results can also be obtained by employing
  # contingency tables, i.e. by specifying data as a list.
  # For this, we need to generate the tables summarizing
  # groupwise how many observations show which level at
  # which variable. These tables can be obtained by

  library(scrime)
  cases <- rowTables(mat[, cl==1])
  controls <- rowTables(mat[, cl==2])
  ltabs <- list(cases, controls)

  # And the same EBAM analysis as above can then be
  # performed by

  out2 <- ebam(ltabs, method=trend.ebam)
  out2

## End(Not run)
```

`trend.stat` *SAM Analysis of Linear Trend*

Description

Generates the required statistics for a Significance Analysis of Microarrays for a linear trend in (ordinal) data.

In the two-class case, the Cochran-Armitage trend statistic is computed. Otherwise, the statistic for the general test of trend described on page 87 of Agresti (2002) is determined.

Should not be called directly, but via sam(..., method = trend.stat).

Usage

```
## Default S3 method:
trend.stat(data, cl, catt = TRUE, approx = TRUE, B = 100,
   B.more = 0.1, B.max = 50000, n.subset = 10, rand = NA, ...)

## S3 method for class 'list'
trend.stat(data, cl, catt = TRUE, approx = TRUE, B = 100,
   B.more = 0.1, B.max = 50000, n.subset = 10, rand = NA, ...)
```

Arguments

data	either a numeric matrix or data frame, or a list. If a matrix or data frame, then each row must correspond to a variable (e.g., a SNP), and each column to a sample (i.e.\ an observation). The values in the matrix or data frame are interpreted as the scores for the different levels of the variables. If the number of observations is huge it is better to specify data as a list consisting of matrices, where each matrix represents one group and summarizes how many observations in this group show which level at which variable. The row and column names of all matrices must be identical and in the same order. The column names must be interpretable as numeric scores for the different levels of the variables. These matrices can, e.g., be generated using the function rowTables from the package **scrime**. (It is recommended to use this function, as trend.stat has been made for using the output of rowTables.) For details on how to specify this list, see the examples section on this man page, and the help for rowChisqMultiClass in the package **scrime**.
cl	a numeric vector of length ncol(data) indicating to which classes the samples in the matrix or data frame data belongs. The values in cl must be interpretable as scores for the different classes. Must be specified if data is a matrix or a data frame, whereas cl can but must not be specified if data is a list. If specified in the latter case, cl must have length data, i.e.\ one score for each of the matrices, and thus for each of the groups. If not specified, cl will be set to the integers between 1 and c, where c is the number of classes/matrices.
catt	should the Cochran-Armitage trend statistic be computed in the two-class case? If FALSE, the trend statistic described on page 87 of Agresti (2002) is determined which differs by the factor $(n-1)/n$ from the Cochran-Armitage trend statistic.
approx	should the null distribution be approximated by the χ^2-distribution with one degree of freedom? If FALSE, a permutation method is used to estimate the null distribution. If data is a list, approx must currently be TRUE.
B	the number of permutations used in the estimation of the null distribution, and hence, in the computation of the expected d-values.
B.more	a numeric value. If the number of all possible permutations is smaller than or equal to (1+B.more)*B, full permutation will be done. Otherwise, B permutations are used.
B.max	a numeric value. If the number of all possible permutations is smaller than or equal to B.max, B randomly selected permutations will be used in the computation of the null distribution. Otherwise, B random draws of the group labels are used.
n.subset	a numeric value indicating how many permutations are considered simultaneously when computing the expected d-values.
rand	numeric value. If specified, i.e. not NA, the random number generator will be set into a reproducible state.
...	ignored.

Value

A list containing statistics required by sam.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Agresti, A.\ (2002). *Categorical Data Analysis*. Wiley, Hoboken, NJ. 2nd Edition.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116-5121.

See Also

SAM-class,sam, chisq.stat, trend.ebam

Examples

```
## Not run:
  # Generate a random 1000 x 40 matrix consisting of the values
  # 1, 2, and 3, and representing 1000 variables and 40 observations.

  mat <- matrix(sample(3, 40000, TRUE), 1000)

  # Assume that the first 20 observations are cases, and the
  # remaining 20 are controls, and that the values 1, 2, 3 in mat
  # can be interpreted as scores for the different levels
  # of the variables represented by the rows of mat.

  cl <- rep(1:2, e=20)

  # Then an SAM analysis of linear trend can be done by

  out <- sam(mat, cl, method=trend.stat)
  out

  # The same results can also be obtained by employing
  # contingency tables, i.e. by specifying data as a list.
  # For this, we need to generate the tables summarizing
  # groupwise how many observations show which level at
  # which variable. These tables can be obtained by

  library(scrime)
  cases <- rowTables(mat[, cl==1])
  controls <- rowTables(mat[, cl==2])
  ltabs <- list(cases, controls)

  # And the same SAM analysis as above can then be
  # performed by

  out2 <- sam(ltabs, method=trend.stat, approx=TRUE)
  out2

## End(Not run)
```

wilc.ebam *EBAM Analysis Using Wilcoxon Rank Statistics*

Description

Generates the required statistics for an Empirical Bayes Analysis of Microarrays analysis using standardized Wilcoxon rank statistics.

Should not be called directly, but via ebam(..., method = wilc.ebam).

Usage

```
wilc.ebam(data, cl, approx50 = TRUE, ties.method = c("min", "random",
    "max"), use.offset = TRUE, df.glm = 5, use.row = FALSE, rand = NA)
```

Arguments

data	a matrix or a data frame. Each row of data must correspond to a variable (e.g., a gene), and each column to a sample (i.e.\ an observation).
cl	a numeric vector of length ncol(data) containing the class labels of the samples. In the two class paired case, cl can also be a matrix with ncol(data) rows and 2 columns. For details on how cl should be specified, see ebam.
approx50	if TRUE, the null distribution will be approximated by the standard normal distribution. Otherwise, the exact null distribution is computed. This argument will automatically be set to FALSE if there are less than 50 samples in each of the groups.
ties.method	either "min" (default), "random", or "max". If "random", the ranks of ties are randomly assigned. If "min" or "max", the ranks of ties are set to the minimum or maximum rank, respectively. For details, see the help of rank. If use.row = TRUE, then ties.method = "max" is used. For the handling of Zeros, see Details.
use.offset	should an offset be used in the Poisson regression employed to estimate the density of the observed Wilcoxon rank sums? If TRUE, the log-transformed values of the null density is used as offset.
df.glm	integer specifying the degrees of freedom of the natural cubic spline employed in the Poisson regression.
use.row	if TRUE, rowWilcoxon is used to compute the Wilcoxon rank statistics.
rand	numeric value. If specified, i.e. not NA, the random number generator will be set into a reproducible state.

Details

Standardized versions of the Wilcoxon rank statistics are computed. This means that $W* = (W - W_{mean})/W_{sd}$ is used as expression score z, where W is the usual Wilcoxon rank sum statistic or Wilcoxon signed rank statistic, respectively.

In the computation of these statistics, the ranks of ties are by default set to the minimum rank. In the computation of the Wilcoxon signed rank statistic, zeros are randomly set either to a very small positive or negative value.

If there are less than 50 observations in each of the groups, the exact null distribution will be used. If there are more than 50 observations in at least one group, the null distribution will by default be approximated by the standard normal distribution. It is, however, still possible to compute the exact null distribution by setting `approx50` to `FALSE`.

Value

A list of statistics required by `ebam`.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Efron, B., Storey, J.D., Tibshirani, R.\ (2001). Microarrays, empirical Bayes methods, and the false discovery rate, *Technical Report*, Department of Statistics, Stanford University.

Schwender, H., Krause, A. and Ickstadt, K. (2003). Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany.

See Also

`ebam`, `wilc.stat`

---

wilc.stat *SAM Analysis Using Wilcoxon Rank Statistics*

Description

Generates the required statistics for a Significance Analysis of Microarrays analysis using standardized Wilcoxon rank statistics.

Should not be called directly, but via sam(..., method = wilc.stat).

Usage

```
wilc.stat(data, cl, gene.names = NULL, R.fold = 1, use.dm = FALSE,
    R.unlog = TRUE, na.replace = TRUE, na.method = "mean",
    approx50 = TRUE, ties.method=c("min","random","max"),
    use.row = FALSE, rand = NA)
```

Arguments

data	a matrix or a data frame. Each row of data must correspond to a variable (e.g., a gene), and each column to a sample (i.e.\ an observation).
cl	a numeric vector of length ncol(data) containing the class labels of the samples. In the two class paired case, cl can also be a matrix with ncol(data) rows and 2 columns. For details on how cl should be specified, see ?sam.
gene.names	a character vector of length nrow(data) containing the names of the genes.
R.fold	a numeric value. If the fold change of a gene is smaller than or equal to R.fold, or larger than or equal to 1/R.fold,respectively, then this gene will be excluded from the SAM analysis. The expression score d of excluded genes is set to NA. By default, R.fold is set to 1 such that all genes are included in the SAM analysis. Setting R.fold to 0 or a negative value will avoid the computation of the fold change. The fold change is only computed in the two-class unpaired case.
use.dm	if TRUE, the fold change is computed by 2 to the power of the difference between the mean log2 intensities of the two groups, i.e.\ 2 to the power of the numerator of the test statistic. If FALSE, the fold change is determined by computing 2 to the power of data (if R.unlog = TRUE) and then calculating the ratio of the mean intensity in the group coded by 1 to the mean intensity in the group coded by 0. The latter is the default, as this definition of the fold change is used in Tusher et al.\ (2001).
R.unlog	if TRUE, the anti-log of data will be used in the computation of the fold change. Otherwise, data is used. This transformation should be done if data is log2-tranformed. (In a SAM analysis, it is highly recommended to use log2-transformed expression data.) Ignored if use.dm = TRUE.
na.replace	if TRUE, missing values will be removed by the genewise/rowwise statistic specified by na.method. If a gene has less than 2 non-missing values, this gene will be excluded from further analysis. If na.replace = FALSE, all genes with one or more missing values will be excluded from further analysis. The expression score d of excluded genes is set to NA.
na.method	a character string naming the statistic with which missing values will be replaced if na.replace=TRUE. Must be either "mean" (default) or median.
approx50	if TRUE, the null distribution will be approximated by the standard normal distribution. Otherwise, the exact null distribution is computed. This argument will automatically be set to FALSE if there are less than 50 samples in each of the groups.
ties.method	either "min" (default), "random", or "max". If "random", the ranks of ties are randomly assigned. If "min" or "max", the ranks of ties are set to the minimum or maximum rank, respectively. For details, see the help of rank. If use.row = TRUE, ties.method = "max" will be used. For the handling of Zeros, see Details.
use.row	if TRUE, rowWilcoxon is used to compute the Wilcoxon rank statistics.
rand	numeric value. If specified, i.e. not NA, the random number generator will be set into a reproducible state.

Details

Standardized versions of the Wilcoxon rank statistics are computed. This means that $W* = (W - W_{mean})/W_{sd}$ is used as expression score d, where W is the usual Wilcoxon rank sum statistic or Wilcoxon signed rank statistic, respectively.

In the computation of these statistics, the ranks of ties are by default set to the minimum rank. In the computation of the Wilcoxon signed rank statistic, zeros are randomly set either to a very small positive or negative value.

If there are less than 50 observations in each of the groups, the exact null distribution will be used. If there are more than 50 observations in at least one group, the null distribution will by default be approximated by the standard normal distribution. It is, however, still possible to compute the exact null distribution by setting `approx50` to `FALSE`.

Value

A list containing statistics required by `sam`.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Schwender, H., Krause, A. and Ickstadt, K. (2003). Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany.

Tusher, V.G., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116-5121.

See Also

`SAM-class`,`sam`, `wilc.ebam`

---

z.ebam *EBAM analysis Using t- or F-test*

Description

Computes the required statistics for an Empirical Bayes Analysis with a modified t- or F-test.

Should not be called directly, but via `ebam(..., method = z.ebam)` or `find.a0(..., method = z.find)`, respectively.

Usage

```
z.ebam(data, cl, a0 = NULL, quan.a0 = NULL, B = 100, var.equal = FALSE,
    B.more = 0.1, B.max = 30000, n.subset = 10, fast = FALSE,
    n.interval = 139, df.ratio = NULL, rand = NA)

z.find(data, cl, B = 100, var.equal = FALSE, B.more = 0.1,
    B.max = 30000)
```

Arguments

data	a matrix, data frame or ExpressionSet object. Each row of data (or exprs(data)) must correspond to a variable (e.g., a gene), and each column to a sample (i.e.\ observation).
cl	a numeric vector of length ncol(data) containing the class labels of the samples. For details on how cl should be specified, see ebam.
a0	a numeric value specifying the fudge factor.
quan.a0	a numeric value between 0 and 1 specifying the quantile of the standard deviations of the genes that is used as fudge factor.
B	an integer indicating how many permutations should be used in the estimation of the null distribution.
var.equal	should the ordinary t-statistic assuming equal group variances be computed? If FALSE (default), Welch's t-statistic will be computed.
B.more	a numeric value. If the number of all possible permutations is smaller than or equal to (1+B.more)*B, full permutation will be done. Otherwise, B permutations are used. This avoids that B permutations will be used – and not all permutations – if the number of all possible permutations is just a little larger than B.
B.max	a numeric value. If the number of all possible permutations is smaller than or equal to B.max, B randomly selected permutations will be used in the computation of the null distribution. Otherwise, B random draws of the group labels are used. In the latter way of permuting, it is possible that some of the permutations are used more than once.
n.subset	an integer specifying in how many subsets the B permutations should be split when computing the permuted test scores. Note that the meaning of n.subset differs between the SAM and the EBAM functions.
fast	if FALSE the exact number of permuted test scores that are more extreme than a particular observed test score is computed for each of the genes. If TRUE, a crude estimate of this number is used.
n.interval	the number of intervals used in the logistic regression with repeated observations for estimating the ratio f_0/f.
df.ratio	integer specifying the degrees of freedom of the natural cubic spline used in the logistic regression with repeated observations.
rand	integer. If specified, i.e. not NA, the random number generator will be set into a reproducible state.

Value

A list of object required by `find.a0` or `ebam`, respectively.

Author(s)

Holger Schwender, <holger.schw@gmx.de>

References

Efron, B., Tibshirani, R., Storey, J.D. and Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment, *JASA*, 96, 1151-1160.

Schwender, H., Krause, A. and Ickstadt, K. (2003). Comparison of the Empirical Bayes and the Significance Analysis of Microarrays. *Technical Report*, SFB 475, University of Dortmund, Germany.

See Also

`ebam`, `find.a0`, `d.stat`

Index