globaltest

April 19, 2009

`GOstructure` *Class "GOstructure" to serve as input for the function gtGO*

Description

The class `GOstructure` stores the Gene Ontology information that can be used by the `gtGO` function to find a GO subgraph significantly associated with a response variable. Objects of type `GOstructure` are typically made with a call to `makeGOstructure`.

Slots

`ids`: Object of class "character". Vector of GO identifiers.

`offspring`: A named list of vectors of GO identifiers. Lists all offspring terms of each term.

`ancestors`: A named list of vectors of GO identifiers. Lists all ancestor terms of each term.

`genesets`: A named list of vectors of gene identifiers. Lists the genes annotated to each GO id.

Methods

show (GOstructure): Summarizes the object.

length (GOstructure): Gives the number of GO ids in the object.

genesets (GOstructure): Extracts the genesets slot.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

See Also

`gtGO`, `makeGOstructure`.

`annotation.vandeVijver` *Some selected annotations for the breast cancer data from Rosetta Inpharmatics LLC and Netherlands Cancer Institute, to serve as example data in the globaltest package.*

Description

A named list of 64 vectors of probe identifiers appearing in the `vandeVijver`, linking the probes to annotation terms from the Gene Ontology (GO) database. All 64 GO terms are descendants of the GO term Cell Cycle (GO:0007049).

Usage

```
data(annotation.vandeVijver)
```

Format

A named list of vectors.

See Also

`vandeVijver`.

`checkerboard` *Checkerboard plot for Global Test*

Description

Produces a plot to visualize the test result produced by `globaltest`, by showing the association between pairs of samples.

Usage

```
checkerboard(gt, geneset, sort = TRUE, drawlabels = TRUE,
labelsize = 0.6, ...)
```

Arguments

`gt`	The output of a call to `globaltest`.
`geneset`	The name or number of the geneset to be plotted (only necessary if multiple genesets were tested)
`sort`	A logical flag to indicate whether the samples should be sorted by the clinical outcome to give a clearer picture.
`drawlabels`	Logical value to control drawing of the samplenames on the x- and y-axis of the plot.
`labelsize`	Relative size of the labels on the x- and y-axis. If it is `NULL` , the current value for `HYPERLINK(par("cex.axis"))(par("cex.axis"))` is used
`...`	Any extra arguments will be forwarded to the plotting function.

Details

The checkerboard shows the pairs of samples which have high covariance in white and the pairs with low covariance in black. This can be used to visualize the data and to search for outlying arrays.

The left and bottom margins are adjusted to allow enough space for the longest samplename.

Value

A matrix giving the old and the new sample numbers.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

References

J. J. Goeman, S. A. van de Geer, F. de Kort and J. C. van Houwelingen, 2004, *A global test for groups of genes: testing association with a clinical outcome*, *Bioinformatics* 20 (1) 93–99. See also the How To Globaltest.pdf included with this package.

See Also

`globaltest`, `sampleplot`, `geneplot`.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with annotation:
data(vandeVijver)
data(annotation.vandeVijver)
aPathway <- annotation.vandeVijver[1]

gt <- globaltest(vandeVijver, "ESR1", aPathway)

if (interactive()){
  checkerboard(gt[1])
}
```

---

geneplot *Gene Plot for Global Test*

Description

Produces a plot to show the influence of individual genes on the test result produced by `globaltest`.

Usage

```
geneplot(gt, geneset, genesubset, scale = FALSE, drawlabels = TRUE,
  labelsize = 0.6, plot = TRUE, addlegend = TRUE, ...)
```

Arguments

gt	The output of a call to globaltest.
geneset	The name or number of the geneset to be plotted (only necessary if multiple genesets were tested).
genesubset	A vector of names or numbers of genes to be plotted (default: plot all genes)
scale	Logical: should the bars be scaled to unit standard deviation?
drawlabels	Logical value to control drawing of gene ids (rownames) on the x-axis of the plot.
labelsize	Relative size of the labels on the x-axis. If it is NULL , the current value for HYPERLINK(par("cex.axis"))(par("cex.axis")) is used
plot	If FALSE: does not plot, but only returns a gt.barplot object.
addlegend	If FALSE: does not add a legend to the plot or to the gt.barplot object.
...	Any extra arguments will be forwarded to the plotting function.

Details

The geneplot shows a bar an a reference line for each gene. The bar shows the influence of each gene on the test result: the test statistic Q is the average of the bars of the genes. The reference line shows the expected influence if the gene was not associated with the outcome. The marks on the bars show the standard deviation of the bar under the null hypothesis. The color of the bar indicates positive or negative correlation of the gene with the clinical outcome, to distinguish between up and downregulation.

The bottom margin is adjusted to allow enough space for the longest gene id to draw under the axis.

Value

An object of type gt.barplot.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

References

For references, type: citation("globaltest"). See also the vignette GlobalTest.pdf included with this package.

See Also

globaltest, sampleplot.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with annotation:
data(vandeVijver)
data(annotation.vandeVijver)

gt <- globaltest(vandeVijver, "StGallen", annotation.vandeVijver)

if (interactive()){
```

```
  geneplot(gt["cell cycle checkpoint"])
}

gp <- geneplot(gt["cell cycle checkpoint"], plot = FALSE)
if (interactive()){
  plot(sort(gp))
}
```

`globaltest` *Global Test*

Description

In microarray data, tests a (list of) group(s) of genes for significant association with a given clinical variable.

Usage

```
globaltest(X, Y, genesets, model,
    levels, d, event = 1, adjust,
    method = c("auto", "asymptotic", "permutations", "gamma"),
    nperm = 10^4, scaleX = TRUE, accuracy = 50, ...)
```

Arguments

`X`	Either a matrix of gene expression data, where columns correspond to samples and rows to genes or a Bioconductor `ExpressionSet`. The data should be properly normalized beforehand (and log- or otherwise transformed), but missing values are allowed (coded as `NA`). Gene and sample names can be included as the row and column names of `X`.
`Y`	A vector with the clinical outcome of interest, having one value for each sample. If `X` is an `ExpressionSet` it can also be the name of a covariate in the `phenoData` from the `ExpressionSet`, or a `formula` object using these names. If the clinical outcome is survival, `Y` should contain the survival times.
`genesets`	Either a vector or a list of vectors. Indicates the group(s) of genes to be tested. Each vector in `genesets` can be given in three formats. Either it can be a vector with 1 (`TRUE`) or 0 (`FALSE`) for each gene in `X`, with 1 indicating that the gene belongs to the group. Or it can be a vector containing the column numbers (in `X`) of the genes belonging to the group. Or it can be a subset of the rownames or `featureNames` for `X`.
`model`	Globaltest will try to determine the correct model from the input of `Y` and `d`. To override the automatic choice, use `model = "logistic"` for a two-valued outcome `Y` , `model = "linear"` for a continuous outcome and `model = "survival"` for a survival outcome.
`levels`	If `Y` is a factor (or a category in the PhenoData slot of `X`) and contains more than 2 levels: `levels` is a vector of levels of `Y` to test. If `levels` is length 2: test these 2 groups against each other. If levels is length 1: test that level against the others.

d	A vector or the name of a covariate in the phenoData from the ExpressionSet X, to indicate which samples experienced an event. Providing a value for d automatically sets model = "survival"
event	The value or values of d that indicates that there was an event.
adjust	Confounders or risk factors for which the test must be adjusted. Must be either a data frame or (if X is an ExpressionSet) the names of covariates in the phenoData from X or a formula object using these names. Default: no adjustment.
method	The method for calculation the p-value. Use codemethod = "asymptotic" for the full asymptotic distribution of the test statistic; method = "gamma" for the gamma (= scaled chi-squared) approximation to that distribution and method = "permutations" for a permutation p-value. The default: method = "auto" chooses the permutations method if the number of possible permutations does not exceed 10,000 and the asymptotic otherwise. Note that method = "gamma" was the default option prior to version 4.0.0.
nperm	A number of permutations. This gives the (maximum) number of permutations to be used if method = "permutations" or method = "auto".
scaleX	If true, rescales the expression matrix to get pleasant value for all test statistics. The expression matrix X is multiplied by a constant in such a way that the expected value EQ of the test statistic for the global test becomes exactly 10. This rescaling has no effect on the p-values.
accuracy	Numerical tuning parameter useable only with the asymptotic method and a non-survival response. Determines how much small eigenvalues of the R matrix are smoothed away to increase computation speed. Choose smaller values for quicker computations but conservative p-values; choose larger values for slower calculations but more accuracy.
...	Captures deprecated input for compatibility with older versions of globaltest.

Details

The Global Test tests whether a group of genes (of any size from one single gene to all genes on the array) is significantly associated with a clinical variable. The group could be for example a known pathway, an area on the genome or the set of all genes. The test investigates whether samples with similar clinical outcomes tend to have similar gene expression patterns. For a significant result it is not necessary that the genes in the group have similar expression patterns, only that many of them are correlated with the outcome.

Value

The function returns an object of class gt.result.

Note

The options globaltest options sampling and permutation have been replaced by separate functions from version 3.0. See sampling and permutations.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

References

For references, type: `citation("globaltest")`. See also the vignette GlobalTest.pdf included with this package.

See Also

Many more examples in the vignette! `geneplot`, `sampleplot`, `sampling`, `gt.multtest`, `permutations`, `checkerboard`, `regressionplot`.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with annotation:
data(vandeVijver)
data(annotation.vandeVijver)

# Many possible calls. See the vignette for more examples and explanation.
globaltest(vandeVijver, "StGallen")
globaltest(vandeVijver, "StGallen", annotation.vandeVijver)
globaltest(vandeVijver, "Surv(TIMEsurvival, EVENTdeath)", annotation.vandeVijver)
globaltest(vandeVijver, StGallen ~ Posnodes + StGallen, annotation.vandeVijver)
globaltest(vandeVijver, "StGallen", method = "p")

# Store the test result
# See help(gt.result) for more options
gt <- globaltest(vandeVijver, "StGallen", annotation.vandeVijver)
gt[1:2]
sort(gt)
p.value(gt)

# Also with simple vector/matrix input
X <- matrix(rnorm(3000), 100, 30)  # random expression data
Y <- 1:30                          # a response variable
pathway <- 1:40                    # a pathway

globaltest(X, Y)
globaltest(X, Y, pathway)
```

`gostructure.vandeVijver`

A GOstructure object for the breast cancer data from Rosetta Inpharmatics LLC and Netherlands Cancer Institute, to serve as example data in the globaltest package.

Description

A object with information on 64 annotation terms from the Gene Ontology (GO) database, all descendants of the GO term Cell Cycle (GO:0007049). The object contains information on ancestors, descendants and associated genes.

Usage

```
data(gostructure.vandeVijver)
```

Format

A GOstructure object.

See Also

GOstructure, vandeVijver.

---

gt.barplot-class *Class "gt.barplot" for results of the functions geneplot and sampleplot*

Description

The class gt.barplot stores the output of a call to geneplot or sampleplot.

Slots

res: Object of class "matrix". The plotted values, with a row for each gene or sample, and the following columns: influence: the height of the bar, Einf: expected influence under the null hypothesis, sd.inf: standard deviation of the influence under the null hypothesis, up: the colour of the plotted bar (1=green; 0=red)

drawlabels: Object of class "logical". Whether gene or sample labels should be printed at the x-axis.

labelsize: Object of class "numeric". The size of the gene/sample labels.

legend: The meaning of the colours of the bars, to be used in the legend.

colour: The numbers of the colours used in the plot.

colourCode: The meaning of the colours used (shorter version of legend.

Methods

show (gt.barplot): Summarizes plot in numbers.

plot (gt.barplot): Redraws the plot. Possible extra arguments: labelsize and drawlabels.

"[" (gt.barplot): Extracts a subset of the genes or samples.

result (gt.barplot): Extracts the results matrix as a data.frame.

length (gt.barplot): The number of bars (genes or samples) in the object.

scale (gt.barplot): Scales the bars to unit standard deviation.

z.score (gt.barplot): Calculates the z.scores for each bar: (influence - expected inf) / sd(inf).

sort (gt.barplot): Sorts the bars to a decreasing z.score.

names (gt.barplot): The names of the genes or samples.

names<- (gt.result): Changes the names of the genes or samples.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

See Also

sampleplot, geneplot, globaltest.

`gt.result-class` *Class "gt.result" for results of the function globaltest*

Description

The class gt.result is the output of a call to `globaltest` and the input of various plotting functions to visualize the test result.

Slots

`res`: Object of class "matrix". Test results summary.

`eX`: Object of class "matrix". The transformed data matrix.

`genesets`: A list of vectors indicating the tested genes.

`IminH`: Object of class "matrix". Needed when drawing the various diagnostic plots.

`fit`: Object of class "list". This list contains one element which is of class "lm", "glm", "coxph" or "coxph.null". Thhis latter object contains the fitted model of the null hypothesis.

`PermQs`: Object of class "matrix". Stores the permuted test statistics generated by a call to `permutations`.

`SamplingZs`: Object of class "list". Stores the standardized test statistics of random gene sets generated by a call to `sampling`.

Methods

show (gt.result): Summarizes the test result.

"[" (gt.result): Extracts results of one or more pathways if multiple pathways were tested.

length (gt.result): The number of pathways tested.

size (gt.result): Extracts a vector with the number of genes tested for each pathway.

p.value (gt.result): Extracts the p-values.

z.score (gt.result): Extracts z-score (Q - EQ) / sd(Q).

sort (gt.result): Sorts the pathways to increasing p-values.

result (gt.result): Extracts the results matrix.

names (gt.result): Extracts the pathway names.

names<- (gt.result): Changes the pathway names.

combine (gt.result): Combines two gt.result objects to one, provided the data and model match.

fit (gt.result): Extracts the fitted (adjust)model.

hist (gt.result): Produces a histogram to visualize the permutations generated by `permutations`.

geneplot (gt.result): The `geneplot` produces a plot to show the influence of individual genes on the test result produced by `globaltest`.

sampleplot (gt.result): The `sampleplot` produces a plot to show the influence of individual samples on the test result produced by `globaltest`.

permutations (gt.result): The function `permutations` recalculates the p-values using permutations of the outcome Y.

sampling (gt.result): The function `sampling` compares the p-values with p-values of randomly generated pathways.

checkerboard (gt.result): Produces a plot to visualize the test result produced by `globaltest` by showing the association between pairs of samples.

regressionplot (gt.result): Produces a plot which can be used to visualize the effect of specific samples on the test result produced by `globaltest`.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

See Also

`globaltest`, `sampleplot`, `geneplot`, `permutations`.

`Multiple testing on the GO graph`

Multiple testing on the GO graph

Description

Three function to test (part of) the GO graph for association of the gene expression profile of GO terms with a response variable. Used together, these functions return multiplicity-adjusted p-values calculated using the Focus Level procedure that preserves the structure of the GO graph.

Usage

```
gtGO(..., GO, focus, maxalpha = 0.05, stopafter = 100, verbose = FALSE)
makeGOstructure(data, annotation, top, only.ids, ontology = c("BP", "CC", "MF"
getFocus(GOstructure, maxatoms = 10)
```

Arguments

`...`	Arguments describing the tests to be preformed are passed on to `globaltest`.
`GO`	An object of class `GOstructure` describing the structure of the GO graph. This object should be created using `makeGOstructure`
`focus`	A vector of GO ids to describe te focus level. Typically made using `getFocus`.
`maxalpha`	The maximum multiplicity-adjusted p-value. The algorithm will stop when this value is exceeded.
`stopafter`	The maximum number of significant GO terms to be found. The algorithm will stop when this value is exceeded.
`verbose`	If set to `TRUE`, prints much more extensive progress information.
`data`	The data set for which the `GOstructure` object is to be made. Can be either an `ExpressionSet` or a vector of probe indicators.
`annotation`	The name of the metaData annotation package to be used. If a metadata package is not available, the name of the organism package (e.g. `org.HS.eg.db` can be given instead, and the mapping from probe identifiers to entrez can be supplied with the `entrez` argument.
`top`	The node to be used as top node of the `GOstructure`. Defaults to one of "GO:0008150", "GO:0005575", "GO:0003674", depending on he ontology chosen. The `GOstructure` object will only contain offspring of the chosen top node.

`only.ids`	A vector of GO ids. If this is supplied, GO ids not appearing in this list will not appear in the `GOstructure` object.
`ontology`	The ontology to be used. One of Biological Process (BP), Cellular Component (CC) or Molecular Function (MF).
`entrez`	A vector of EntrezGene ids with the same length as the number of genes in `data`, (and in the same order) giving the mapping from probe identifiers to EntrezGene ids. If this argument is used, the `annotation` argument should give the name of the organism annotation package.
`unreliable`	Can be used to designate one or more of the GO evidence codes as unreliable. Value must be a vector containing one or more from "IC", "IDA", "IEA", "IEP", "IGI", "IMP", "IPI", "ISS", "NAS", "ND", "RCA", "TAS", "NR".
`GOstructure`	An object of class `GOstructure`.
`maxatoms`	A tuning parameter that governs the choice of the focus level. `getFocus` will choose those GO terms in the focus level in such a way that all offspring gene sets of each focus level term can be constructed as unions of no more than `maxatoms` atom gene sets. Default value of `maxatoms` is 10. Higher values quickly lead to slower computation. Lower values typically lead to reduced power.

Details

These functions should be used in the following order. First use `makeGOstructure` to make a GO graph tailored to a specific data set. Then `getFocus` can be used to choose a focus level. Finally `gtGO` performs the focus level procedure.

Value

The function returns a named vector of multiplicity-adjusted p-values. Adjusted p-values of GO terms not appearing in this vector are larger than the chosen value of `maxalpha`.

Note

`gtGO` cannot be used in combination with the permutation version of `globaltest`.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

References

For references, type: `citation("globaltest")`. See also the vignette GlobalTest.pdf included with this package.

See Also

Examples in the vignette! `globaltest`, `GOstructure`, `gt.multtest`.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with small GO structure (all descendants of Cell Cycle)
# See the vignette for details on making a GO structure object.
data(vandeVijver)
data(gostructure.vandeVijver)

focus <- getFocus(gostructure.vandeVijver)
adjustedP <- gtGO(vandeVijver, "Surv(TIMEsurvival, EVENTdeath)",
  GO = gostructure.vandeVijver, focus = focus)

# The significant GO graph can be visualized using
# the GOstats and Rgraphviz packages.
# See the vignette for details. Type: vignette("GlobalTest")
```

`mlogit` *Multinomial Logistic Regression*

Description

Fits a multinomial logistic regression model to a nominal scale outcome.

Usage

```
mlogit(formula, data, control = glm.control())
```

Arguments

`formula`	An object of class `formula` containing a symbolic description of the model to be fit. See the documentation of `formula` for details.
`data`	An optional data frame containing the variables in the model. If not found in 'data', the variables are taken from the environment from which 'mlogit' is called.
`control`	A list of parameters for controlling the fitting process. See the documentation of `glm.control` for details.

Details

The function mlogit fits a multinomial logistic regression model for a multi-valued outcome with nominal scale. The implementation and behaviour are designed to mimic those of `glm`, but the options are (as yet) more limited. Missing values are not allowed in the data.

The model is fitted without using a reference outcome category; the parameters are made identifiable by the requirement that the sum of corresponding regression coefficients over the outcome categories is zero.

Value

An object of (S4) class `mlogit`. The class has slots: coefficients (matrix), standard.err (matrix), fitted.values (matrix), x (matrix), y (matrix), formula (formula), call (call), df.null (numeric), df.residual (numeric), null.deviance (numeric), deviance (numeric), iter (numeric), converged (logical).

Methods implemented for the `mlogit` class are `coefficients`, `fitted.values`, `residuals` and which extract the relevant quantities, and `summary`, which gives the same output as with a `glm` object.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

See Also

`glm`, `multinom`.

Examples

```
y <- factor(rep(1:4, 5))
x <- 1:20
fit <- mlogit(y ~ x)
summary(fit)
residuals(fit)
```

---

gt.multtest *Correct globaltest results for multiple testing*

Description

Corrects the raw p-values resulting from a call to `globaltest` for multiple testing, using either Benjamini and Hochberg's False Discovery Rate or Holm's procedure for controlling the Family-Wise Error Rate.

Usage

```
gt.multtest(gt, proc = c("FDR", "FWER"))
```

Arguments

`gt`	The output of a call to `globaltest`.
`proc`	The procedure to be used. Either "FDR" for Benjamini and Hochberg's (1995) False Discovery Rate-controlling procedure or "FWER" for Holm's (1979) Family-Wise Error Rate controlling procedure.

Details

This function is completely based on the `mt.rawp2adjp` function from the `multtest` package.

Value

An object of class `gt.result`.

Note

This function must be called *prior* to any selection of significant genes.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

References

For references, type: citation("globaltest"). See also the vignette GlobalTest.pdf included with this package.

See Also

globaltest, gtGO.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with annotation:
data(vandeVijver)
data(annotation.vandeVijver)

gt <- globaltest(vandeVijver, "StGallen", annotation.vandeVijver)
sort(gt.multtest(gt))
```

---

permutations *Permutations version of the Global Test*

Description

Produces permutations of the globaltest test statistic and uses these to recalculate the p-value.

Usage

```
permutations(gt, geneset, nperm = 10^4)
```

Arguments

gt	The output of a call to globaltest.
geneset	The name or number of the geneset(s) to be used (only necessary if multiple genesets were tested).
nperm	The number of permutations to be used.

Details

A permutation p-value is calculated by comparing the value of the test statistic using permutations of the clinical outcome to the value of the test statistic for the true clinical outcome. If the number of possible permutations is smaller than the value of nperm, all possible permutations are used, otherwise nperm random permutations. The permuted test statistics are stored for later visualisation using hist.

Value

An object of class gt.result.

Note

Permutations does not work if the adjusted version of globaltest was used.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

References

For references, type: citation("globaltest"). See also the vignette GlobalTest.pdf included with this package.

See Also

globaltest, sampleplot, geneplot.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with annotation:
data(vandeVijver)
data(annotation.vandeVijver)
aPathway <- annotation.vandeVijver[1]

gt <- globaltest(vandeVijver, "StGallen", aPathway)
perm.gt <- permutations(gt)

if (interactive()) {
  hist(perm.gt)
}
```

---

regressionplot *Regression Plot for Global Test*

Description

Produces a plot which can be used to visualize the effect of specific samples on the test result produced by globaltest.

Usage

```
regressionplot(gt, geneset, sampleid, ...)
```

Arguments

gt	The output of a call to globaltest.
geneset	The name or number of the geneset to be plotted (only necessary if multiple genesets were tested).
sampleid	An optional vector of names or numbers of the samples of interest.
...	Any extra arguments will be forwarded to the plotting function.

Details

The regressionplot plots, for all pairs of samples, the covariance between the expression patterns against the covariance between their clinical outcomes. Each point in the plot therefore represents a pair of samples. A regression line is fitted through the samples, which visualizes the test result of the function `globaltest`. A steeply increasing slope indicates a high (possibly significant) value of the test statistic.

An optional argument `sampleid` can be supplied, giving sample numbers of possibly outlying arrays. In this case, all pairs of arrays involving one of the arrays in `sampleid` is marked as a red cross, while the other pairs are marked as a blue dot. The blue line which is fitted through all points can now be compared to a red dotted line which is fitted though only the red crosses.

Value

`NULL` (no output).

Note

Regressionplot does not work if the adjusted version of globaltest was used.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

References

J. J. Goeman, S. A. van de Geer, F. de Kort and J. C. van Houwelingen, 2004, *A global test for groups of genes: testing association with a clinical outcome*, *Bioinformatics* 20 (1) 93–99. See also the How To Globaltest.pdf included with this package.

See Also

`globaltest`, `sampleplot`, `geneplot`.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with annotation:
data(vandeVijver)
data(annotation.vandeVijver)

gt <- globaltest(vandeVijver, "ESR1", annotation.vandeVijver)

if (interactive()) {
  regressionplot(gt[1])
  regressionplot(gt[1], sampleid = "122")
}
```

`sampleplot` *Sample Plot for Global Test*

Description

Produces a plot to show the influence of individual samples on the test result produced by `globaltest`.

Usage

```
sampleplot(gt, geneset, samplesubset, scale = TRUE, drawlabels = TRUE,
  labelsize = 0.6, plot = TRUE, addlegend = TRUE, ...)
```

Arguments

`gt`	The output of a call to `globaltest`.
`geneset`	The name or number of the geneset to be plotted (only necessary if multiple genesets were tested).
`samplesubset`	A vector of names or numbers of samples to be plotted. Default: plot all samples
`scale`	Logical: should the bars be scaled to unit standard deviation?
`drawlabels`	Logical value to control drawing of the samplenames on the x-axis of the plot.
`labelsize`	Relative size of the labels on the x-axis. If it is `NULL` , the current value for `HYPERLINK(par("cex.axis"))(par("cex.axis"))` is used
`plot`	If `FALSE`: does not plot, but only returns a `gt.barplot` object.
`addlegend`	If `FALSE`: does not add a legend to the plot or to the `gt.barplot` object.
`...`	Any extra arguments will be forwarded to the plotting function.

Details

The sampleplot shows a bar and a reference line for each sample. The bar shows the influence of each gene on the test statistic. Samples with a positive influence carry evidence against the null hypothesis (in favour of a significant p-value), because they are are similar in expression profile to samples with a similar clinical outcome. Samples with a negative influence bar supply evidence in favour of the null hypothesis and of a non-significant p-value: they are relatively similar in expression profile to samples with a different clinical outcome.

The influence varies around zero if the tested geneset is not associated with the outcome. Marks on the bars show the standarddeviation of the influence under the null hypothesis for those samples which are more than one standard deviation away from zero.

The color of the bar indicates the sign of the residual of Y. In a logistic model the coloring this distinguishes the original groups.

The bottom margin is adjusted to allow enough space for the longest samplename to draw under the axis.

Value

An object of type `gt.barplot`.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

See Also

globaltest, geneplot, regressionplot, checkerboard.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with annotation:
data(vandeVijver)
data(annotation.vandeVijver)

gt <- globaltest(vandeVijver, "StGallen", annotation.vandeVijver)

if (interactive()){
  sampleplot(gt[1])
}

sp <- sampleplot(gt[1], plot = FALSE)
if (interactive()){
  plot(sort(sp))
}
```

---

sampling *Sampling random 'pathways' for the Global Test*

Description

For every pathway in a result of `globaltest`, calculates how many randomly drawn groups of genes of the same size have a smaller or equal p-value.

Usage

```
sampling(gt, geneset, ndraws = 10^3)
```

Arguments

`gt`	The output of a call to `globaltest`.
`geneset`	The name or number of the geneset(s) to be used (only necessary if multiple genesets were tested).
`ndraws`	The number of random pathways to be used.

Details

For every pathway in `gt[geneset]`, a number `ndraws` random 'pathways' is selected by randomly sampling sets of genes of the same size as the tested pathway. A 'comparative p-value' is calculated by counting what proportion of the random pathways has a larger standardized test statistic (Q - EQ) / sd(Q) than the tested pathway.

Value

An object of class `gt.result`.

Note

The function `sampling` cannot be applied to a `gt.result` object resulting from a call to `permutations`.

Author(s)

Jelle Goeman: ⟨j.j.goeman@lumc.nl⟩; Jan Oosting

References

Jelle J. Goeman, Jan Oosting, Anne-Marie Cleton-Jansen, Jakob, K. Anninga, Hans C. van Houwelingen (2005) Testing association of a pathway with survival. Bioinformatics 21, 1950-1957. See also the vignette Globaltest.pdf that comes with this package.

See Also

`globaltest`, `permutations`, `sampleplot`, `geneplot`.

Examples

```
# Breast cancer data (ExpressionSet) from the Netherlands Cancer
# Institute with annotation:
data(vandeVijver)
data(annotation.vandeVijver)

gt <- globaltest(vandeVijver, "StGallen", annotation.vandeVijver)

sampling(gt, ndraws=100)
```

---

vandeVijver — *A subset of the Breast cancer data from Rosetta Inpharmatics LLC and Netherlands Cancer Institute.*

Description

An `ExpressionSet` containing expression data of 100 breast cancer patients on 407 selected genes plus the following phenotype data.

SampleID Sample ID given by NKI for 295 samples in cohort study (NEJM 347, 01999, 2002)

FirstSeriesID Sample ID in first series given by NKI for 78 samples in our first study (Nature 415, p530, 2002)

Posnodes Lymph node status from pathology report

EVENTmeta Distant metastasis at any time during follow-up, 0 = no, 1 = yes

EVENTdeath Death, 0 = no, 1 = yes

TIMEsurvival Total survival interval in years between first date of treatment and last date of follow-up (if EVENTdeath = 0) or date of death (if EVENTdeath = 1)

TIMErecurrence Disease free interval in years between first date of treatment and last date of follow-up (if all 'EVENTs' are 0) or date of first EVENT (if one ore more 'EVENTs' are 1, i.e., local recurrence, loco-regional recurrence, second primary tumor or distant metastasis)

TIMEmeta Metastasis free interval in years between first date of treatment and date of diagnosis of distant metastasis (only if EVENTmeta = 1) (note that interval relates to metastasis at any time during follow-up and not just as first event)

ESR1 Estrogen receptor alpha expression measurement from microarray (0 = no, 1 = yes)

NIH Patient characteristics according to NIH criteria, 1 = low risk, 0 = high risk

St Gallen Patient characteristics according to St. Gallen criteria, 1 = low risk, 0 = intermediate/high risk

Usage

```
data(vandeVijver)
```

Format

An `ExpressionSet`. See the Biobase package

References

M.J. van de Vijver, Y.D. He, L.J. van 't Veer, H. Dai, A.A.M. Hart, D.W. Voskuil, G.J. Schreiber, J.L. Peterse, C. Roberts, M.J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E.T. Rutgers, S.H. Friend, and R. Bernards (2002). A gene-expression signature as a predictor of survival in breast cancer. New England Journal of Medicine 347 (25), 1999–2009.

See Also

`annotation.vandeVijver`.

Index