Package 'GlobalAncova'

November 12, 2026

Title Global test for groups of variables via model comparisons

Version 4.30.0

Date 2021-01-27

Author U. Mansmann, R. Meister, M. Hummel, R. Scheufele, with contributions from S. Knueppel

Description The association between a variable of interest (e.g. two groups) and the global pattern of a group of variables (e.g. a gene set) is tested via a global F-test. We give the following arguments in support of the GlobalAncova approach: After appropriate normalisation, gene-expression-data appear rather symmetrical and outliers are no real problem, so least squares should be rather robust. ANCOVA with interaction yields saturated data modelling e.g. different means per group and gene. Covariate adjustment can help to correct for possible selection bias. Variance homogeneity and uncorrelated residuals cannot be expected. Application of ordinary least squares gives unbiased, but no longer optimal estimates (Gauss-Markov-Aitken). Therefore, using the classical F-test is inappropriate, due to correlation. The test statistic however mirrors deviations from the null hypothesis. In combination with a permutation approach, empirical significance levels can be approximated. Alternatively, an approximation yields asymptotic p-values. The framework is generalized to groups of categorical variables or even mixed data by a likelihood ratio approach. Closed and hierarchical testing procedures are supported. This work was supported by the NGFN grant 01 GR 0459, BMBF, Germany and BMBF grant 01ZX1309B, Germany.

Maintainer Manuela Hummel <manuela.hummel@web.de>

Depends methods, corpcor, globaltest

Imports annotate, AnnotationDbi, Biobase, dendextend, GSEABase, VGAM

Suggests GO.db, golubEsets, hu6800.db, vsn, Rgraphviz

License GPL (>= 2)

biocViews Microarray, OneChannel, DifferentialExpression, Pathways, Regression

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

git_branch RELEASE_3_23

git_last_commit 7942dc5

git_last_commit_date 2026-04-28

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Contents

bindata . . . 2
colon.normal . . . 3
colon.pheno . . . 4
colon.tumour . . . 5
GAhier class . . . 5
gGlobalAncova . . . 7
gGlobalAncova.hierarchical . . . 8
GlobalAncova . . . 10
GlobalAncova gene set testing methods . . . 13
GlobalAncova-methods . . . 15
GlobalAncova.closed . . . 16
GlobalAncova.closed-methods . . . 18
GlobalAncova.decomp . . . 19
pair.compare . . . 20
pathways . . . 22
phenodata . . . 22
Plot.all . . . 23
Plot.features . . . 24
Plot.genes . . . 25
Plot.genes-methods . . . 27
Plot.sequential . . . 28
Plot.subjects . . . 29
Plot.subjects-methods . . . 31
vantVeer . . . 31

Index **33**

`bindata` *Simulated binary data*

Description

Simulated data consisting of 24 binary variables and a binary outcome `Y` with 100 observations. Names of variables associated with the outcome start with `true`, names of other variables start with `zero`.

Usage

```
data(bindata)
```

See Also

`gGlobalAncova`

Examples

```
data(bindata)
#str(bindata)
```

`colon.normal` *Gene expression data*

Description

Normalized gene expression data of 12 patients with colorectal cancer. Samples are taken from inside the tumours. Additionally, from same patients samples are taken from normal tissue, see `colon.normal`. The expression matrix is only an exemplary subset of 1747 probe sets associated with cell proliferation.

Usage

```
data(colon.normal)
```

Format

The format is:

```
num [1:1747, 1:12] 8.74 10.53 8.48 12.69 8.55 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:1747] "200808_s_at" "215706_x_at" "217185_s_at" "202136_at" ...
..$ : chr [1:12] "Co10.N.E.84.F.CEL" "Co14.N.E.89.F.CEL" "Co17.N.E.1037.F.CEL" "Co1.N.E.31.F.CEL"
...
```

References

Groene, J. et al., 2006, Transcriptional census of 36 microdissected colorectal cancers yields a gene signature to distinguish UICC II and III, *Int J Cancer* 119(8):1829–36.

Examples

```
data(colon.normal)
#str(colon.normal)
```

`colon.pheno` *Covariate information for the colon data*

Description

Covariate data for the colon data example:

sex Sex of the patient.

age Age of the patient.

location Location of the tumour.

grade Histologic tumour grade.

UICC.stage UICC stage of colorectal carcinoma.

Usage

```
data(colon.pheno)
```

Format

The format is:

`'data.frame':` 12 obs. of 5 variables:

`$sex:` Factor w/ 2 levels "0","1": 2 2 1 2 2 1 2 1 2 1 ...

`$age:` int 71 76 63 73 58 66 60 66 86 76 ...

`$location:` Factor w/ 2 levels "distal","proximal": 1 1 1 1 1 1 1 1 2 1 ...

`$grade:` Factor w/ 2 levels "2","3": 1 1 2 2 1 2 1 2 2 2 ...

`$UICC.stage:` Factor w/ 2 levels "2","3": 2 1 2 1 2 1 1 1 2 1 ...

References

Groene, J. et al., 2006, Transcriptional census of 36 microdissected colorectal cancers yields a gene signature to distinguish UICC II and III, *Int J Cancer* 119(8):1829–36.

Examples

```
data(colon.pheno)
#str(colon.pheno)
```

colon.tumour *Gene expression data*

Description

Normalized gene expression data of 12 patients with colorectal cancer. Samples are taken from inside the tumours. Additionally, from same patients samples are taken from normal tissue, see colon.normal. The expression matrix is only an exemplary subset of 1747 probe sets associated with cell proliferation.

Usage

```
data(colon.tumour)
```

Format

The format is:

```
num [1:1747, 1:12] 8.77 10.40 8.52 12.86 8.28 ...
- attr(*, "dimnames")=List of 2
..$ : chr [1:1747] "200808_s_at" "215706_x_at" "217185_s_at" "202136_at" ...
..$ : chr [1:12] "Co10.T.IT.83.F.CEL" "Co14.T.IT.88.F.CEL" "Co17.T.IT.563.F.CEL" "Co1.T.IT.30.F.CEL"
...
```

References

Groene, J. et al., 2006, Transcriptional census of 36 microdissected colorectal cancers yields a gene signature to distinguish UICC II and III, *Int J Cancer* 119(8):1829–36.

Examples

```
data(colon.tumour)
#str(colon.tumour)
```

GAhier class *Class* "GAhier"

Description

Class for storing results of hierarchical testing procedure performed by gGlobalAncova.hierarchical

Usage

```
## S4 method for signature 'GAhier'
show(object)
## S4 method for signature 'GAhier'
results(object)
## S4 method for signature 'GAhier'
sigEndnodes(object, onlySingleton=FALSE)
## S4 method for signature 'GAhier'
Plot.hierarchy(object, dend, col=1:2, lwd=1:2, collab, returndend=FALSE, cex.labels=1.5, ...)
```

Arguments

object	object of class GAhier
onlySingleton	if TRUE, only names of singleton variables within the significant nodes are returned
dend	dendrogram object specifying the hierarchy of the variables
col	colors for significant and non-significant nodes and branches, respectively
lwd	line width for branches to non-significant and significant nodes, respectively
collab	vector of colors for coloring dendrogram leave labels (can be independent of significant/non-significant nodes); has to be named according to variable names; if missing, significant and non-significant variables are colored using colors defined in col
returndend	if TRUE, updated dendrogram object is returned (e.g. for use in further plots)
cex.labels	size of leave labels
...	further graphical parameters, passed to plot.dendrogram

Slots

clustervariables: Object of class "list" containing names of variables in each tested cluster

p.values: Object of class "list" containing p-values for each tested cluster

alpha: Object of class "numeric"; chosen global significance level; if K had been specified, this additionally contains the adjusted significance levels for the K sub-hierarchies

n.variables: Object of class "numeric"; number of variables in total

permstats: Object of class "matrixOrNULL"; if returnPermstats had been set to TRUE, this is a matrix containing individual statistics for all variables for all permutations, otherwise NULL

Methods

show signature(object = "GAhier"): Show general information and significant end nodes

results signature(object = "GAhier"): Get a data.frame with significant end nodes, number and names of variables included in each node and corresponding p-value

sigEndnodes signature(object = "GAhier"): Get names of signficant end nodes

Plot.hierarchy signature(object = "GAhier"): Plot hierarchy dendrogram, where significant nodes (and branches to those nodes) are highlighted

Note

Coloring the dendrogram in `Plot.hierarchy` is based on functionality from the **globaltest** package

Author(s)

Manuela Hummel <m.hummel@dkfz.de>

See Also

gGlobalAncova.hierarchical

Examples

```
showClass("GAhier")

# see examples in documentation of gGlobalAncova.hierarchical
```

---

gGlobalAncova *Generalized GlobalAncova group test*

Description

Computation of a permutation test for the association between sets of variables (e.g. genes, SNPs, ...) and clinical entities. The variables can be continuous, binary, categorical, ordinal, or of mixed types. The test is carried out by comparing the deviances of the full generalized linear model and the reduced model lacking the design parameters of interest. The variable-wise models are summarized to a global test statistic for the complete set.

Usage

```
gGlobalAncova(data, formula.full, formula.red=~1, model.dat, Sets, sumstat=sum, perm=10000)
```

Arguments

`data`	`data.frame` of variables to be tested in sets (columns=variables); (multi-) categorical variables should be factors, ordinal variables should be ordered factors
`formula.full`	model formula for the full model
`formula.red`	model formula for the reduced model (that does not contain the terms of interest)
`model.dat`	`data.frame` of regressors, containing variables specified in `formula.full` and `formula.red`
`Sets`	vector of names or indices of variables or list of those, defining sets of variables
`sumstat`	function for summarizing univariate test statistics; default is `sum`
`perm`	number of permutations

Value

A data.frame with test statistic and p-value for each tested set.

Note

The test is fast for categorical data and categorical design variable. For other types of variables and more complex designs it is rather slow.

This work was supported by BMBF grant 01ZX1309B, Germany.

Author(s)

Reinhard Meister <meister@beuth-hochschule.de>
Manuela Hummel <m.hummel@dkfz.de>

See Also

GlobalAncova

Examples

```
data(bindata)
gGlobalAncova(bindata[,-1], formula.full = ~group, model.dat = bindata, perm = 1000)
```

---

gGlobalAncova.hierarchical

Hierarchical testing procedure using generalized GlobalAncova

---

Description

Hierarchical testing procedure according to Meinshausen (2008) screening for groups of related variables within a hierarchy instead of screening individual variables independently. Groups are tested by the generalized GlobalAncova approach. The family-wise error rate is simultaneously controlled over all levels of the hierarchy. In order to reduce computational complexity for large hierarchies, a "short cut" is implemented, where the testing procedure is applied separately to K sub-hierarchies. The p-values are adjusted such that they are identical to the ones obtained when testing the complete hierarchy.

Usage

```
gGlobalAncova.hierarchical(data, H, formula.full, formula.red=~1, model.dat, sumstat=sum,
                           alpha=0.05, K, perm=10000, returnPermstats=FALSE, permstats)
```

Arguments

data	data.frame of variables (columns=variables) to be tested hierarchically; (multi-) categorical variables should be factors, ordinal variables should be ordered factors
H	dendrogram object specifying the hierarchy of the variables; labels(H) has to coincide with colnames(data)
formula.full	model formula for the full model

formula.red	model formula for the reduced model (that does not contain the terms of interest)
model.dat	data.frame of regressors, containing variables specified in formula.full and formula.red
sumstat	function for summarizing univariate test statistics; default is sum
alpha	global significance level
K	optional integer; if this is specified, "short cut" on hierarchical testing will be applied separately to K sub-hierarchies
perm	number of permutations
returnPermstats	if TRUE, the variable-wise statistics for all permutations are returned
permstats	if variable-wise permutation statistics were calculated previously, they can be provided in order not to repeat permutation testing (but only the hierarchical prodcedure); useful e.g. if procedure is run again with different alpha and/or hierarchy H; NOTE: data, formula.full and formula.red must be identical to the previous call

Details

The hierarchical procedure starts with testing the global null hypothesis that all variables are not associated with the design of interest, and then moves down the given hierarchy testing subclusters of variables. A subcluster is only tested if the null hypothesis corresponding to its ancestor cluster could be rejected. The p-values are adjusted for multiple testing according to cluster size $p_{C,adj} = p_C m/|C|$, where m is the total number of variables and $|C|$ is the number of variables in cluster C.

If K is specified and the procedure is split to K sub-hierarchies containing $m_1, \ldots, m_K$ variables, p-values are additionally adjusted by $\tau = m/m_k, k = 1, \ldots, K$, such that resulting p-values are identical to the ones obtained when testing the complete hierarchy

$$p_{C,adj,k} \cdot \tau = p_C m_k/|C| \cdot m/m_k = p_{C,adj}$$

Value

an object of class GAhier

Author(s)

Manuela Hummel <m.hummel@dkfz.de>

References

Meinshausen N, 2008. Hierarchical testing of variable importance. Biometrika, 95(2):265

See Also

gGlobalAncova, GAhier, Plot.hierarchy

Examples

```
data(bindata)
X <- as.matrix(bindata[,-1])

# get a hierarchy for variables
dend <- as.dendrogram(hclust(dist(t(X))))

# hierarchical test
set.seed(555)
res <- gGlobalAncova.hierarchical(X, H = dend, formula.full = ~group, model.dat = bindata, alpha = 0.05, perm = 1000
res
results(res)

# get names of significant clusters
sigEndnodes(res)

# visualize results
Plot.hierarchy(res, dend)

# starting with 3 sub-hierarchies
set.seed(555)
res2 <- gGlobalAncova.hierarchical(X, H = dend, K = 3, formula.full = ~group, model.dat = bindata, alpha = 0.05, per

results(res2)
```

GlobalAncova *Global test for differential gene expression*

Description

Computation of a F-test for the association between expression values and clinical entities. In many cases a two way layout with gene and a dichotomous group as factors will be considered. However, adjustment for other covariates and the analysis of arbitrary clinical variables, interactions, gene co-expression, time series data and so on is also possible. The test is carried out by comparison of corresponding linear models via the extra sum of squares principle. Corresponding p-values, permutation p-values and/or asymptotic p-values are given.

There are three possible ways of using `GlobalAncova`. The general way is to define formulas for the full and reduced model, respectively, where the formula terms correspond to variables in `model.dat`. An alternative is to specify the full model and the name of the model terms that shall be tested regarding differential expression. In order to make this layout compatible with the function call in the first version of the package there is also a method where simply a group variable (and possibly covariate information) has to be given. This is maybe the easiest usage in cases where no 'special' effects like e.g. interactions are of interest.

Usage

```
## S4 method for signature 'matrix,formula,formula,ANY,missing,missing,missing'
GlobalAncova(xx, formula.full, formula.red, model.dat,
```

```
       test.genes, method = c("permutation","approx","both","Fstat"), perm = 10000, max.group.size = 25

## S4 method for signature
## 'matrix,formula,missing,ANY,missing,missing,character'
GlobalAncova(xx, formula.full, model.dat,test.terms,
       test.genes, method = c("permutation","approx","both","Fstat"), perm = 10000, max.group.size = 25

## S4 method for signature 'matrix,missing,missing,missing,ANY,ANY,missing'
GlobalAncova(xx, group, covars = NULL,
       test.genes, method = c("permutation","approx","both","Fstat"), perm = 10000, max.group.size = 25
```

Arguments

xx	Matrix of gene expression data, where columns correspond to samples and rows to genes. The data should be properly normalized beforehand (and log- or otherwise transformed). Missing values are not allowed. Gene and sample names can be included as the row and column names of xx.
formula.full	Model formula for the full model.
formula.red	Model formula for the reduced model (that does not contain the terms of interest.)
model.dat	Data frame that contains all the variable information for each sample.
group	Vector with the group membership information.
covars	Vector or matrix which contains the covariate information for each sample.
test.terms	Character vector that contains names of the terms of interest.
test.genes	Vector of gene names or a list where each element is a vector of gene names.
method	p-values can be calculated permutation-based ("permutation") or by means of an approximation for a mixture of chi-square distributions ("approx"). Both p-values are provided when specifying method = "both". With option "Fstat" only the global F-statistics are returned without p-values or further information.
perm	Number of permutations to be used for the permutation approach. The default is 10,000.
max.group.size	Maximum size of a gene set for which the asymptotic p-value is calculated. For bigger gene sets the permutation approach is used.
eps	Resolution of the asymptotic p-value.
acc	Accuracy parameter needed for the approximation. Higher values indicate higher accuracy.

Value

If test.genes = NULL a list with components

effect	Name(s) of the tested effect(s)
ANOVA	ANOVA table
test.result	F-value, theoretical p-value, permutation-based and/or asymptotic p-value
terms	Names of all model terms

If a collection of gene sets is provided in `test.genes` a matrix is returned whose columns show the number of genes, value of the F-statistic, theoretical p-value, permutation-based and/or asymptotic p-value for each of the gene sets.

Methods

xx = "matrix", formula.full = "formula", formula.red = "formula", model.dat = "ANY", group = "missing", covars
In this method, besides the expression matrix `xx`, model formulas for the full and reduced model and a data frame `model.dat` specifying corresponding model terms have to be given. Terms that are included in the full but not in the reduced model are those whose association with differential expression will be tested. The arguments `group`, `covars` and `test.terms` are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "formula", formula.red = "missing", model.dat = "ANY", group = "missing", covars =
In this method, besides the expression matrix `xx`, a model formula for the full model and a data frame `model.dat` specifying corresponding model terms are required. The character argument `test.terms` names the terms of interest whose association with differential expression will be tested. The basic idea behind this method is that one can select single terms, possibly from the list of terms provided by previous `GlobalAncova` output, and test them without having to specify each time a model formula for the reduced model. The arguments `formula.red`, `group` and `covars` are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "missing", formula.red = "missing", model.dat = "missing", group = "ANY", covars =
Besides the expression matrix `xx` a clinical variable `group` is required. Covariate adjustment is possible via the argument `covars` but more complex models have to be specified with the methods described above. This method emulates the function call in the first version of the package. The arguments `formula.full`, `formula.red`, `model.dat` and `test.terms` are '"missing"' since they are not needed for this method.

Note

This work was supported by the NGFN project 01 GR 0459, BMBF, Germany.

Author(s)

Reinhard Meister <meister@beuth-hochschule.de>
Ulrich Mansmann <mansmann@ibe.med.uni-muenchen.de>
Manuela Hummel <m.hummel@dkfz-heidelberg.de>
with contributions from Sven Knueppel

References

Mansmann, U. and Meister, R., 2005, Testing differential gene expression in functional groups, *Methods Inf Med* 44 (3).

See Also

`Plot.genes`, `Plot.subjects`, `GlobalAncova.closed`, `GAGO`, `GlobalAncova.decomp`

Examples

```
data(vantVeer)
data(phenodata)
data(pathways)

GlobalAncova(xx = vantVeer, formula.full = ~metastases + ERstatus, formula.red = ~ERstatus, model.dat = phenodata,
GlobalAncova(xx = vantVeer, formula.full = ~metastases + ERstatus, test.terms = "metastases", model.dat = phenodata
GlobalAncova(xx = vantVeer, group = phenodata$metastases, covars = phenodata$ERstatus, test.genes=pathways[1], met
```

GlobalAncova gene set testing methods

Gene set testing of gene set databases using GlobalAncova

Description

Three functions adapted from package **globaltest** to test gene sets from databases for association of the gene expression profile with a response variable. Three function are provided for Gene Ontology and for the Broad Institute's gene sets.

Usage

```
GAGO (xx, ..., id, annotation, probe2entrez,
           ontology = c("BP", "CC", "MF"), minsize=1, maxsize=Inf,
           multtest = c("holm", "focuslevel", "BH", "BY"),
           focuslevel = 10, sort = TRUE)

GABroad (xx, ..., id, annotation, probe2entrez, collection,
           category = c("c1", "c2", "c3", "c4", "c5"),
           multtest = c("holm", "BH", "BY"), sort = TRUE)
```

Arguments

`xx`	Matrix of gene expression data, where columns correspond to samples and rows to genes. Gene names have to be included as the row names of `xx`
`...`	Arguments describing the tests to be performed are passed on to `GlobalAncova`. Note that only the approximative version of `GlobalAncova` is used here and hence the parameter `method` is not available. Even though the number of permutations (`perm`) may be specified since very large gene sets (with more genes than `max.group.size`) are treated with the permutation test.
`id`	The identifier(s) of gene sets to be tested (character vector). If omitted, tests all gene sets in the database.
`annotation`	The name of the probe annotation package for the microarray that was used, or the name of the genome wide annotation package for the species (e.g. org.Hs.eg.db for human). If an organism package is given, the argument `probe2entrez` must be supplied.

probe2entrez	Use only if no probe annotation package is available. A mapping from probe identifiers to entrez gene ids. May be an environment, named list or named vector.
multtest	The method of multiple testing correction. Choose from: Benjamini and Hochberg FDR control (BH); Benjamini and Yekutieli FDR control (BY) or Holm familywise error control (holm). For GAGO also the focus level method is available. See focusLevel.
sort	If TRUE, sorts the results to increasing p-values.
ontology	The ontology or ontologies to be used. Default is to use all three ontologies.
minsize	The minimum number of probes that may be annotated to a gene set. Gene sets with fewer annotated probes are discarded.
maxsize	The maximum number of probes that may be annotated to a gene set. Gene sets with more annotated probes are discarded.
focuslevel	The focus level to be used for the focus level method. Either a vector of gene set ids, or a numerical level. In the latter case, findFocus is called with maxsize at the specified level to find a focus level.
collection	The Broad gene set collection, created by a call to getBroadSets.
category	The subcategory of the Broad collection to be tested. The default is to test all sets.

Details

These are utility functions to make it easier to do gene set testing of gene sets available in gene set databases. The functions automatically retrieve the gene sets, preprocess and select them, perform global test, do multiple testing correction, and sort the results on the basis of their p-values. All functions require that annotate and the appropriate annotation packages are installed. GAGO requires the GO.db package; GABroad requires the user to download the XML file "msigdb_v2.5.xml" from \ http://www.broad.mit.edu/gsea/downloads.jsp, and to preprocess that file using the getBroadSets function.

Value

The function returns a data frame with raw and multiplicity-adjusted p-values for each gene set.

Note

Functions GAGOand GABroad correspond to functions gtGO, and gtBroad in package **globaltest**. The difference is in the use of the GlobalAncova test instead of gt within the procedures.

Author(s)

Jelle Goeman: <j.j.goeman@lumc.nl>; Jan Oosting; Manuela Hummel

References

Goeman, J.J. and Mansmann, U., Multiple testing on the directed acyclic graph of Gene Ontology. Bioinformatics 2008; 24(4): 537-44.

See Also

gtGO, gtKEGG, gtBroad, GlobalAncova, gt,

Examples

```
# see vignettes of packages GlobalAncova and globaltest and help of gtGO
```

GlobalAncova-methods *Methods for Function GlobalAncova*

Description

There are three possible ways of using GlobalAncova. The general way is to define formulas for the full and reduced model, respectively, where the formula terms correspond to variables in model.dat. An alternative is to specify the full model and the name of the model terms that shall be tested regarding differential expression. In order to make this layout compatible with the function call in the first version of the package there is also a method where simply a group variable (and possibly covariate information) has to be given. This is maybe the easiest usage in cases where no 'special' effects like e.g. interactions are of interest.

Methods

xx = "matrix", formula.full = "formula", formula.red = "formula", model.dat = "ANY", group = "missing", covars
In this method, besides the expression matrix xx, model formulas for the full and reduced model and a data frame model.dat specifying corresponding model terms have to be given. Terms that are included in the full but not in the reduced model are those whose association with differential expression will be tested. The arguments group, covars and test.terms are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "formula", formula.red = "missing", model.dat = "ANY", group = "missing", covars =
In this method, besides the expression matrix xx, a model formula for the full model and a data frame model.dat specifying corresponding model terms are required. The character argument test.terms names the terms of interest whose association with differential expression will be tested. The basic idea behind this method is that one can select single terms, possibly from the list of terms provided by previous GlobalAncova output, and test them without having to specify each time a model formula for the reduced model. The arguments formula.red, group and covars are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "missing", formula.red = "missing", model.dat = "missing", group = "ANY", covars =
Besides the expression matrix xx a clinical variable group is required. Covariate adjustment is possible via the argument covars but more complex models have to be specified with the methods described above. This method emulates the function call in the first version of the package. The arguments formula.full, formula.red, model.dat and test.terms are '"missing"' since they are not needed for this method.

`GlobalAncova.closed` *Closed testing procedure for testing several groups of genes using GlobalAncova*

Description

Computation of a closed testing procedure for several groups of genes, e.g. pathways, as an alternative of correcting for multiple testing. Starting from the pathways of interest a family of null hypotheses is created that is closed under intersection. Each null hypothesis can be rejected at a given level if it is rejected along with all hypotheses included in it.

There are three possible ways of using `GlobalAncova`. Also `GlobalAncova.closed` can be invoked with these three alternatives.

Usage

```
## S4 method for signature
## 'matrix,list,formula,formula,ANY,missing,missing,missing'
GlobalAncova.closed(xx, test.genes,
       formula.full, formula.red, model.dat, previous.test, level, method = c("permutation","approx"),
          max.group.size = 2500, eps = 1e-16, acc = 50)

## S4 method for signature
## 'matrix,list,formula,missing,ANY,missing,missing,character'
GlobalAncova.closed(xx, test.genes,
       formula.full, model.dat, test.terms, previous.test, level, method = c("permutation","approx"),
          max.group.size = 2500, eps = 1e-16, acc = 50)

## S4 method for signature
## 'matrix,list,missing,missing,missing,ANY,ANY,missing'
GlobalAncova.closed(xx, test.genes,
       group, covars = NULL, previous.test, level, method = c("permutation","approx"), perm = 10000,
          max.group.size = 2500, eps = 1e-16, acc = 50)
```

Arguments

`xx`	Matrix of gene expression data, where columns correspond to samples and rows to genes. The data should be properly normalized beforehand (and log- or otherwise transformed). Missing values are not allowed. Gene and sample names can be included as the row and column names of `xx`.
`test.genes`	A list of named pathways that shall be tested, each containing vectors of gene names.
`previous.test`	The output of a call to `GlobalAncova` with specified option `test.genes` according to the pathways of interest (optional).
`level`	The global level of significance of the testing procedure.
`formula.full`	Model formula for the full model.

formula.red	Model formula for the reduced model (that does not contain the terms of interest).
model.dat	Data frame that contains all the variable information for each sample.
group	Vector with the group membership information.
covars	Vector or matrix which contains the covariate information for each sample.
test.terms	Character vector that contains names of the terms of interest.
method	Raw p-values can be calculated permutation-based ("permutation") or by means of an approximation ("approx").
perm	Number of permutations to be used for the permutation approach. The default is 10,000.
max.group.size	Maximum size of a gene set for which the asymptotic p-value is calculated. For bigger gene sets the permutation approach is used.
eps	Resolution of the asymptotic p-value.
acc	Accuracy parameter needed for the approximation. Higher values indicate higher accuracy.

Value

A list with components

new.data	Family of null hypotheses (vectors of genes to be tested simultaneously with GlobalAncova).
test.results	Test results for each pathway of interest and all hypotheses included in it.
significant	Names of the significant pathways.
not.significant	Names of the non significant pathways.

Methods

xx = "matrix", test.genes="list", formula.full = "formula", formula.red = "formula", model.dat = "ANY", group = '

In this method, besides the expression matrix xx and the list of gene groups test.genes, model formulas for the full and reduced model and a data frame model.dat specifying corresponding model terms have to be given. Terms that are included in the full but not in the reduced model are those whose association with differential expression will be tested. The arguments group, covars and test.terms are '"missing"' since they are not needed for this method.

xx = "matrix", test.genes="list", formula.full = "formula", formula.red = "missing", model.dat = "ANY", group = "

In this method, besides the expression matrix xx and the list of gene groups test.genes, a model formula for the full model and a data frame model.dat specifying corresponding model terms are required. The character argument test.terms names the terms of interest whose association with differential expression will be tested. The arguments formula.red, group and covars are '"missing"' since they are not needed for this method.

xx = "matrix", test.genes="list", formula.full = "missing", formula.red = "missing", model.dat = "missing", group =

Besides the expression matrix xx and the list of gene groups test.genes a clinical variable

group is required. Covariate adjustment is possible via the argument covars but more complex models have to be specified with the methods described above. This method emulates the function call in the first version of the package. The arguments formula.full, formula.red, model.dat and test.terms are '"missing"' since they are not needed for this method.

Note

This work was supported by the NGFN project 01 GR 0459, BMBF, Germany.

Author(s)

Reinhard Meister <meister@beuth-hochschule.de>
Ulrich Mansmann <mansmann@ibe.med.uni-muenchen.de>
Manuela Hummel <m.hummel@dkfz.de>

References

Marcus, R., Peritz, E. and Gabriel, K.R., 1976, On closed testing procedures with special reference to ordered analysis of variance, *Biometrika* 63 (3): 655–660.

See Also

GlobalAncova, Plot.genes, Plot.subjects

GlobalAncova.closed-methods

Methods for Function GlobalAncova.closed

Description

There are three possible ways of using GlobalAncova, use methods ? GlobalAncova for getting more information. Also GlobalAncova.closed can be invoked with these three alternatives.

Methods

xx = "matrix", test.genes="list", formula.full = "formula", formula.red = "formula", model.dat = "ANY", group = '

In this method, besides the expression matrix xx and the list of gene groups test.genes, model formulas for the full and reduced model and a data frame model.dat specifying corresponding model terms have to be given. Terms that are included in the full but not in the reduced model are those whose association with differential expression will be tested. The arguments group, covars and test.terms are '"missing"' since they are not needed for this method.

xx = "matrix", test.genes="list", formula.full = "formula", formula.red = "missing", model.dat = "ANY", group = "

In this method, besides the expression matrix xx and the list of gene groups test.genes, a model formula for the full model and a data frame model.dat specifying corresponding model terms are required. The character argument test.terms names the terms of interest whose association with differential expression will be tested. The arguments formula.red, group and covars are '"missing"' since they are not needed for this method.

xx = "matrix", test.genes="list", formula.full = "missing", formula.red = "missing", model.dat = "missing", group =
Besides the expression matrix `xx` and the list of gene groups `test.genes` a clinical variable `group` is required. Covariate adjustment is possible via the argument `covars` but more complex models have to be specified with the methods described above. This method emulates the function call in the first version of the package. The arguments `formula.full`, `formula.red`, `model.dat` and `test.terms` are '"missing"' since they are not needed for this method.

`GlobalAncova.decomp` *GlobalAncova with sequential and type III sum of squares decomposition and adjustment for global covariates*

Description

Computation of a F-test for the association between expression values and clinical entities. The test is carried out by comparison of corresponding linear models via the extra sum of squares principle. In models with various influencing factors extra sums of squares can be treated with sequential and type III decomposition. Adjustment for global covariates, e.g. gene expression values in normal tissue as compared to tumour tissue, can be applied. Given theoretical p-values may not be appropriate due to correlations and non-normality. The functions are hence seen more as a descriptive tool.

Usage

```
GlobalAncova.decomp(xx, formula, model.dat = NULL, method = c("sequential", "type3", "all"), test.gene
```

Arguments

`xx`	Matrix of gene expression data, where columns correspond to samples and rows to genes. The data should be properly normalized beforehand (and log- or otherwise transformed). Missing values are not allowed. Gene and sample names can be included as the row and column names of `xx`.
`formula`	Model formula for the linear model.
`model.dat`	Data frame that contains all the variable information for each sample.
`method`	Whether sequential or type III decomposition or both should be calculated.
`test.genes`	Vector of gene names or a list where each element is a vector of gene names.
`genewise`	Shall the sequential decomposition be displayed for each single gene in a (small) gene set?
`zz`	Global covariate, i.e. matrix of same dimensions as `xx`.
`zz.per.gene`	If set to `TRUE` the adjustment for the global covariate is applied on a gene-wise basis.

Value

Depending on parameters `test.genes`, `method` and `genewise` ANOVA tables, or lists of ANOVA tables for each decomposition and/or gene set, or lists with components of ANOVA tables for each gene are returned.

Note

This work was supported by the NGFN project 01 GR 0459, BMBF, Germany.

Author(s)

Ramona Scheufele <ramona.scheufele@charite.de>
Reinhard Meister <meister@tfh-berlin.de>
Manuela Hummel <hummel@ibe.med.uni-muenchen.de>
Urlich Mansmann <mansmann@ibe.med.uni-muenchen.de>

See Also

Plot.sequential, pair.compare, GlobalAncova

Examples

```
data(vantVeer)
data(phenodata)
data(pathways)

# sequential or type III decomposition
GlobalAncova.decomp(xx = vantVeer, formula = ~ grade + metastases + ERstatus, model.dat = phenodata, method = "seque
GlobalAncova.decomp(xx = vantVeer, formula = ~ grade + metastases + ERstatus, model.dat = phenodata, method = "type3

# adjustment for global covariate
data(colon.tumour)
data(colon.normal)
data(colon.pheno)
GlobalAncova.decomp(xx = colon.tumour, formula = ~ UICC.stage + sex + location, model.dat = colon.pheno, method = "a
```

---

pair.compare *Pairwise comparisons of factor levels within GlobalAncova*

Description

Pairwise comparisons of gene expression in different levels of a factor by GlobalAncova tests. The method uses the reduction in residual sum of squares obtained when two respective factor levels are set to the same level. Holm-adjusted permutation-based p-values are given.

Usage

```
pair.compare(xx, formula, group, model.dat = NULL, test.genes = NULL, perm = 10000)
```

Arguments

xx	Matrix of gene expression data, where columns correspond to samples and rows to genes. The data should be properly normalized beforehand (and log- or otherwise transformed). Missing values are not allowed. Gene and sample names can be included as the row and column names of xx.
formula	Model formula for the linear model.
group	Factor for which pairwise comparisons shall be calculated.
model.dat	Data frame that contains all the variable information for each sample.
test.genes	Vector of gene names or a list where each element is a vector of gene names.
perm	Number of permutations to be used for the permutation approach. The default is 10,000.

Value

An ANOVA table, or list of ANOVA tables for each gene set, for the pairwise comparisons.

Note

This work was supported by the NGFN project 01 GR 0459, BMBF, Germany.

Author(s)

Ramona Scheufele <ramona.scheufele@charite.de>
Reinhard Meister <meister@tfh-berlin.de>
Manuela Hummel <m.hummel@dkfz.de>
Urlich Mansmann <mansmann@ibe.med.uni-muenchen.de>

See Also

GlobalAncova, GlobalAncova.decomp

Examples

```
data(vantVeer)
data(phenodata)
data(pathways)

pair.compare(xx = vantVeer, formula = ~ grade, group = "grade", model.dat = phenodata, test.genes = pathways[1:3], p
```

pathways *Cancer related pathways*

Description

A list of nine cancer related pathways corresponding to the van t'Veer data. Each element contains a vector gene names corresponding to those in the data set.

Usage

```
data(pathways)
```

Format

The format is:

```
List of 9
$ androgen_receptor_signaling: chr [1:72] "AW025529" "NM_001648" "NM_001753" "NM_003298"
...
$ apoptosis : chr [1:187] "AB033060" "NM_002341" "NM_002342" "AI769763" ...
$ cell_cycle_control : chr [1:31] "NM_001759" "NM_001760" "NM_001786" "NM_001789" ...
$ notch_delta_signalling : chr [1:34] "NM_002405" "AL133036" "NM_003260" "NM_004316"
...
$ p53_signalling : chr [1:33] "NM_002307" "NM_002392" "NM_003352" "NM_002745" ...
$ ras_signalling : chr [1:266] "D25274" "AI033397" "NM_003029" "NM_001626" ...
$ tgf_beta_signaling : chr [1:82] "NM_003036" "AI090812" "AI697699" "AI760298" ...
$ tight_junction_signaling : chr [1:326] "D25274" "AA604213" "AF018081" "NM_003005" ...
$ wnt_signaling : chr [1:176] "AB033058" "AB033087" "NM_003012" "NM_003014" ...
```

Examples

```
data(pathways)
#str(pathways)
```

phenodata *Covariate information for the van t'Veer data*

Description

Covariate data for the van t'Veer example:

Sample Sample number.

metastases Development of distant metastases within five years (0-no/1-yes).

grade Tumor grade (three ordere levels).

ERstatus Estrogen receptor status (pos-positive/neg-negative).

Usage

```
data(phenodata)
```

Format

The format is:

'data.frame': 96 obs. of 4 variables:

$Sample: int 1 2 3 4 5 6 7 8 9 10 ...

$metastases: int 0 0 0 0 0 0 0 0 0 0 ...

$grade: int 2 1 3 3 3 2 1 3 3 2 ...

$ERstatus: Factor w/ 2 levels "neg","pos": 2 2 1 2 2 2 2 1 2 2 ...

Examples

```
data(phenodata)
#str(phenodata)
```

Plot.all *Combined visualization of sequential decomposition and influence of single genes on the GlobalAncova statistic*

Description

Plot that combines Plot.genes and Plot.sequential into one graphic.

Usage

```
Plot.all(xx, formula, model.dat = NULL, test.genes = NULL, name.geneset = "")
```

Arguments

xx	Matrix of gene expression data, where columns correspond to samples and rows to genes. The data should be properly normalized beforehand (and log- or otherwise transformed). Missing values are not allowed. Gene and sample names can be included as the row and column names of xx.
formula	Model formula for the linear model.
model.dat	Data frame that contains all the variable information for each sample.
test.genes	Vector of gene names or gene indices specifying a gene set.
name.geneset	Name of the plotted geneset.

Note

This work was supported by the NGFN project 01 GR 0459, BMBF, Germany.

Author(s)

Ramona Scheufele <ramona.scheufele@charite.de>
Reinhard Meister <meister@tfh-berlin.de>
Manuela Hummel <m.hummel@dkfz.de>
Urlich Mansmann <mansmann@ibe.med.uni-muenchen.de>

See Also

Plot.genes, Plot.sequential, GlobalAncova.decomp, GlobalAncova

Examples

```
data(vantVeer)
data(phenodata)
data(pathways)

Plot.all(vantVeer, formula = ~ ERstatus + metastases + grade, model.dat = phenodata, test.genes = pathways[[3]], na
```

Plot.features *Features Plot for generalized Global Ancova*

Description

Produces a plot to show the influence of individual variables on the test result produced by gGlobalAncova. The variables can be continuous, binary, categorical, ordinal, or of mixed types.

Usage

```
Plot.features(data, formula.full, formula.red = ~1, model.dat, Set, returnValues = FALSE, ...)
```

Arguments

data	data.frame of variables to be tested in sets (columns=variables); (multi-) categorical variables should be factors, ordinal variables should be ordered factors
formula.full	model formula for the full model
formula.red	model formula for the reduced model (that does not contain the terms of interest)
model.dat	data.frame of regressors, containing variables specified in formula.full and formula.red
Set	optional vector of names or indices of variables, defining the set of variables to plot; if missing, all variables in data are shown
returnValues	shall variable-wise statistics = bar heights be returned?
...	graphical parameters passed to barplot

Value

If returnValues = TRUE, a vector with the bar heights is returned.

Author(s)

Manuela Hummel <m.hummel@dkfz.de>

See Also

gGlobalAncova, Plot.genes

Examples

```
data(bindata)
Plot.features(bindata[,-1], formula.full = ~group, model.dat = bindata)
```

Plot.genes *Genes Plot for Global Ancova*

Description

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

There are three possible ways of using GlobalAncova. Also Plot.genes can be invoked with these three alternatives.

Usage

```
## S4 method for signature 'matrix,formula,formula,ANY,missing,missing,missing'
Plot.genes(xx, formula.full, formula.red, model.dat, group, covars = NULL,test.terms,test.genes, Color

## S4 method for signature
## 'matrix,formula,missing,ANY,missing,missing,character'
Plot.genes(xx, formula.full, formula.red, model.dat, group, covars = NULL,test.terms,test.genes, Color

## S4 method for signature 'matrix,missing,missing,missing,ANY,ANY,missing'
Plot.genes(xx,formula.full, formula.red, model.dat, group, covars = NULL,test.terms,test.genes, Colorg
```

Arguments

xx	Matrix of gene expression data, where columns correspond to samples and rows to genes. The data should be properly normalized beforehand (and log- or otherwise transformed). Missing values are not allowed. Gene and sample names can be included as the row and column names of xx.
formula.full	Model formula for the full model.
formula.red	Model formula for the reduced model (that does not contain the terms of interest.)
model.dat	Data frame that contains all the variable information for each sample.
group	Vector with the group membership information.
covars	Vector or matrix which contains the covariate information for each sample.

test.terms	Character vector that contains names of the terms of interest.
test.genes	Vector of gene names or gene indices specifying the gene set. If missing, the plot refers to all genes in xx.
Colorgroup	Character variable giving the group that specifies coloring. If the function is called using the argument group then this variable is assumed to be relevant for coloring.
legendpos	Position of the legend (a single keyword from the list '"bottomright"', '"bottom"', '"bottomleft"', '"left"', '"topleft"', '"top"', '"topright"', '"right"' and '"center"').
returnValues	Shall bar heights (gene-wise reduction in sum of squares) be returned?
bar.names	Vector of bar labels. If missing, gene names from test.genes or row names of xx are taken.
...	Graphical parameters for specifying colors, titles etc.

Methods

xx = "matrix", formula.full = "formula", formula.red = "formula", model.dat = "ANY", group = "missing", covars
In this method, besides the expression matrix xx, model formulas for the full and reduced model and a data frame model.dat specifying corresponding model terms have to be given. Terms that are included in the full but not in the reduced model are those whose association with differential expression will be tested. The arguments group, covars and test.terms are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "formula", formula.red = "missing", model.dat = "ANY", group = "missing", covars =
In this method, besides the expression matrix xx, a model formula for the full model and a data frame model.dat specifying corresponding model terms are required. The character argument test.terms names the terms of interest whose association with differential expression will be tested. The arguments formula.red, group and covars are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "missing", formula.red = "missing", model.dat = "missing", group = "ANY", covars =
Besides the expression matrix xx a clinical variable group is required. Covariate adjustment is possible via the argument covars but more complex models have to be specified with the methods described above. This method emulates the function call in the first version of the package. The arguments formula.full, formula.red, model.dat and test.terms are '"missing"' since they are not needed for this method.

Note

This work was supported by the NGFN project 01 GR 0459, BMBF, Germany.

Author(s)

Reinhard Meister <meister@tfh-berlin.de>
Ulrich Mansmann <mansmann@ibe.med.uni-muenchen.de>
Manuela Hummel <m.hummel@dkfz.de>

See Also

GlobalAncova, Plot.subjects, Plot.sequential

Examples

```
data(vantVeer)
data(phenodata)
data(pathways)

Plot.genes(xx = vantVeer, formula.full = ~metastases + ERstatus, formula.red = ~ERstatus, model.dat = phenodata, te
Plot.genes(xx = vantVeer, formula.full = ~metastases + ERstatus, test.terms = "metastases", model.dat = phenodata,
Plot.genes(xx = vantVeer, group = phenodata$metastases, covars = phenodata$ERstatus, test.genes = pathways[[3]])
```

Plot.genes-methods *Methods for Function Plot.genes*

Description

There are three possible ways of using `GlobalAncova`, use `methods ? GlobalAncova` for getting more information. Also `Plot.genes` can be invoked with these three alternatives.

Methods

xx = "matrix", formula.full = "formula", formula.red = "formula", model.dat = "ANY", group = "missing", covars
In this method, besides the expression matrix `xx`, model formulas for the full and reduced model and a data frame `model.dat` specifying corresponding model terms have to be given. Terms that are included in the full but not in the reduced model are those whose association with differential expression will be tested. The arguments `group`, `covars` and `test.terms` are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "formula", formula.red = "missing", model.dat = "ANY", group = "missing", covars =
In this method, besides the expression matrix `xx`, a model formula for the full model and a data frame `model.dat` specifying corresponding model terms are required. The character argument `test.terms` names the terms of interest whose association with differential expression will be tested. The arguments `formula.red`, `group` and `covars` are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "missing", formula.red = "missing", model.dat = "missing", group = "ANY", covars =
Besides the expression matrix `xx` a clinical variable `group` is required. Covariate adjustment is possible via the argument `covars` but more complex models have to be specified with the methods described above. This method emulates the function call in the first version of the package. The arguments `formula.full`, `formula.red`, `model.dat` and `test.terms` are '"missing"' since they are not needed for this method.

`Plot.sequential` *Visualization of sequential decomposition*

Description

Plot to show the sum of squares decomposition for each gene into parts according to all variables.

Usage

```
Plot.sequential(xx, formula, model.dat = NULL, test.genes = NULL, name.geneset = "")
```

Arguments

`xx`	Matrix of gene expression data, where columns correspond to samples and rows to genes. The data should be properly normalized beforehand (and log- or otherwise transformed). Missing values are not allowed. Gene and sample names can be included as the row and column names of `xx`.
`formula`	Model formula for the linear model.
`model.dat`	Data frame that contains all the variable information for each sample.
`test.genes`	Vector of gene names or gene indices specifying a gene set.
`name.geneset`	Name of the plotted geneset.

Note

This work was supported by the NGFN project 01 GR 0459, BMBF, Germany.

Author(s)

Ramona Scheufele <ramona.scheufele@charite.de>
Reinhard Meister <meister@tfh-berlin.de>
Manuela Hummel <m.hummel@dkfz.de>
Urlich Mansmann <mansmann@ibe.med.uni-muenchen.de>

See Also

`GlobalAncova.decomp`, `Plot.genes`, `GlobalAncova`

Examples

```
data(vantVeer)
data(phenodata)
data(pathways)

Plot.sequential(vantVeer, formula = ~ ERstatus + metastases + grade, model.dat = phenodata, test.genes = pathways[[
```

Plot.subjects *Subjects Plot for GlobalAncova*

Description

Produces a plot to show the influence of the samples on the test result produced by GlobalAncova.

There are three possible ways of using GlobalAncova. Also Plot.subjects can be invoked with these three alternatives.

Usage

```
## S4 method for signature 'matrix,formula,formula,ANY,missing,missing,missing'
Plot.subjects(xx, formula.full, formula.red, model.dat, group,covars = NULL, test.terms,test.genes, Co

## S4 method for signature
## 'matrix,formula,missing,ANY,missing,missing,character'
Plot.subjects(xx, formula.full,formula.red, model.dat, group,covars = NULL, test.terms,test.genes, Col

## S4 method for signature 'matrix,missing,missing,missing,ANY,ANY,missing'
Plot.subjects(xx, formula.full, formula.red, model.dat, group,covars = NULL, test.terms,test.genes, Co
```

Arguments

xx	Matrix of gene expression data, where columns correspond to samples and rows to genes. The data should be properly normalized beforehand (and log- or otherwise transformed). Missing values are not allowed. Gene and sample names can be included as the row and column names of xx.
formula.full	Model formula for the full model.
formula.red	Model formula for the reduced model (that does not contain the terms of interest.)
model.dat	Data frame that contains all the variable information for each sample.
group	Vector with the group membership information.
covars	Vector or matrix which contains the covariate information for each sample.
test.terms	Character vector that contains names of the terms of interest.
test.genes	Vector of gene names or gene indices specifying the gene set. If missing, the plot refers to all genes in xx.
Colorgroup	Character variable giving the group that specifies coloring. If the function is called using the argument group then this variable is assumed to be relevant for coloring.
sort	Should the samples be ordered by colorgroup?
legendpos	Position of the legend (a single keyword from the list '"bottomright"', '"bottom"', '"bottomleft"', '"left"', '"topleft"', '"top"', '"topright"', '"right"' and '"center"').

returnValues	Shall bar heights (subject-wise reduction in sum of squares) be returned?
bar.names	Vector of bar labels. If missing, column names of xx are taken.
...	Graphical parameters for specifying colors, titles etc.

Methods

xx = "matrix", formula.full = "formula", formula.red = "formula", model.dat = "ANY", group = "missing", covars
In this method, besides the expression matrix xx, model formulas for the full and reduced model and a data frame model.dat specifying corresponding model terms have to be given. Terms that are included in the full but not in the reduced model are those whose association with differential expression will be tested. The arguments group, covars and test.terms are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "formula", formula.red = "missing", model.dat = "ANY", group = "missing", covars =
In this method, besides the expression matrix xx, a model formula for the full model and a data frame model.dat specifying corresponding model terms are required. The character argument test.terms names the terms of interest whose association with differential expression will be tested. The arguments formula.red, group and covars are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "missing", formula.red = "missing", model.dat = "missing", group = "ANY", covars =
Besides the expression matrix xx a clinical variable group is required. Covariate adjustment is possible via the argument covars but more complex models have to be specified with the methods described above. This method emulates the function call in the first version of the package. The arguments formula.full, formula.red, model.dat and test.terms are '"missing"' since they are not needed for this method.

Note

This work was supported by the NGFN project 01 GR 0459, BMBF, Germany.

Author(s)

Reinhard Meister <meister@tfh-berlin.de>
Ulrich Mansmann <mansmann@ibe.med.uni-muenchen.de>
Manuela Hummel <m.hummel@dkfz.de>

See Also

GlobalAncova, Plot.genes, Plot.sequential

Examples

```
data(vantVeer)
data(phenodata)
data(pathways)

Plot.subjects(xx = vantVeer, formula.full = ~metastases + ERstatus, formula.red = ~ERstatus, model.dat = phenodata,
Plot.subjects(xx = vantVeer, formula.full = ~metastases + ERstatus, test.terms = "metastases", model.dat = phenodat
Plot.subjects(xx = vantVeer, group = phenodata$metastases, covars = phenodata$ERstatus, test.genes = pathways[[3]]
```

Plot.subjects-methods *Methods for Function Plot.subjects*

Description

There are three possible ways of using GlobalAncova, use methods ? GlobalAncova for getting more information. Also Plot.subjects can be invoked with these three alternatives.

Methods

xx = "matrix", formula.full = "formula", formula.red = "formula", model.dat = "ANY", group = "missing", covars
In this method, besides the expression matrix xx, model formulas for the full and reduced model and a data frame model.dat specifying corresponding model terms have to be given. Terms that are included in the full but not in the reduced model are those whose association with differential expression will be tested. The arguments group, covars and test.terms are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "formula", formula.red = "missing", model.dat = "ANY", group = "missing", covars =
In this method, besides the expression matrix xx, a model formula for the full model and a data frame model.dat specifying corresponding model terms are required. The character argument test.terms names the terms of interest whose association with differential expression will be tested. The arguments formula.red, group and covars are '"missing"' since they are not needed for this method.

xx = "matrix", formula.full = "missing", formula.red = "missing", model.dat = "missing", group = "ANY", covars =
Besides the expression matrix xx a clinical variable group is required. Covariate adjustment is possible via the argument covars but more complex models have to be specified with the methods described above. This method emulates the function call in the first version of the package. The arguments formula.full, formula.red, model.dat and test.terms are '"missing"' since they are not needed for this method.

vantVeer *Gene expression data*

Description

Normalized gene expression data for the van t'Veer example: A subset of 96 samples without BRCA1 or BRCA2 mutations and 1113 genes associated with nine cancer related pathways (see also ?pathways) was chosen.

Usage

```
data(vantVeer)
```

Format

The format is:
num [1:1113, 1:96] 0.13 0.936 -0.087 0.118 0.168 -0.081 0.023 -0.086 -0.154 0.025 ...
- attr(*, "dimnames")=List of 2
..$: chr [1:1113] "AW025529" "NM_001648" "NM_001753" "NM_003298" ...
..$: chr [1:96] "1" "2" "3" "4" ...

Examples

```
data(vantVeer)
#str(vantVeer)
```

Index

∗ **classes**
GAhier class, 5
∗ **datasets**
bindata, 2
colon.normal, 3
colon.pheno, 4
colon.tumour, 5
pathways, 22
phenodata, 22
vantVeer, 31
∗ **hplot**
Plot.all, 23
Plot.features, 24
Plot.genes, 25
Plot.genes-methods, 27
Plot.sequential, 28
Plot.subjects, 29
Plot.subjects-methods, 31
∗ **htest**
GlobalAncova gene set testing methods, 13
∗ **methods**
GlobalAncova-methods, 15
∗ **models**
gGlobalAncova, 7
gGlobalAncova.hierarchical, 8
GlobalAncova, 10
GlobalAncova.closed, 16
GlobalAncova.closed-methods, 18
GlobalAncova.decomp, 19
pair.compare, 20

bindata, 2

colon.normal, *3*, 3, *5*
colon.pheno, 4
colon.tumour, 5

dendrogram, *6*

findFocus, *14*
focusLevel, *14*

GABroad *(*GlobalAncova gene set testing methods*)*, 13
GAGO, *12*
GAGO *(*GlobalAncova gene set testing methods*)*, 13
GAhier, *9*
GAhier *(*GAhier class*)*, 5
GAhier class, 5
GAhier-class *(*GAhier class*)*, 5
getBroadSets, *14*
gGlobalAncova, *2*, 7, *9*, *24, 25*
gGlobalAncova.hierarchical, *5*, *7*, 8
GlobalAncova, *8*, 10, *13*, *15*, *18*, *20, 21*, *24–26*, *28–30*
GlobalAncova gene set testing methods, 13
GlobalAncova,matrix,formula,formula,ANY,missing,missing,mi *(*GlobalAncova*)*, 10
GlobalAncova,matrix,formula,missing,ANY,missing,missing,ch *(*GlobalAncova*)*, 10
GlobalAncova,matrix,missing,missing,missing,ANY,ANY,missin *(*GlobalAncova*)*, 10
GlobalAncova-methods, 15
GlobalAncova.closed, *12*, 16
GlobalAncova.closed,matrix,list,formula,formula,ANY,missin *(*GlobalAncova.closed*)*, 16
GlobalAncova.closed,matrix,list,formula,missing,ANY,missin *(*GlobalAncova.closed*)*, 16
GlobalAncova.closed,matrix,list,missing,missing,missing,AN *(*GlobalAncova.closed*)*, 16
GlobalAncova.closed-methods, 18
GlobalAncova.decomp, *12*, 19, *21*, *24*, *28*
gt, *15*
gtBroad, *15*
gtGO, *15*
gtKEGG, *15*

pair.compare, *20*, 20

pathways, 22
phenodata, 22
Plot.all, 23
plot.dendrogram, *6*
Plot.features, 24
Plot.genes, *12*, *18*, *23–25*, 25, *28*, *30*
Plot.genes,matrix,formula,formula,ANY,missing,missing,missing-method
(Plot.genes), 25
Plot.genes,matrix,formula,missing,ANY,missing,missing,character-method
(Plot.genes), 25
Plot.genes,matrix,missing,missing,missing,ANY,ANY,missing-method
(Plot.genes), 25
Plot.genes-methods, 27
Plot.hierarchy, *9*
Plot.hierarchy (GAhier class), 5
Plot.hierarchy,GAhier-method (GAhier
class), 5
Plot.sequential, *20*, *23*, *24*, *26*, 28, *30*
Plot.subjects, *12*, *18*, *26*, 29
Plot.subjects,matrix,formula,formula,ANY,missing,missing,missing-method
(Plot.subjects), 29
Plot.subjects,matrix,formula,missing,ANY,missing,missing,character-method
(Plot.subjects), 29
Plot.subjects,matrix,missing,missing,missing,ANY,ANY,missing-method
(Plot.subjects), 29
Plot.subjects-methods, 31

results (GAhier class), 5
results,GAhier-method (GAhier class), 5

show,GAhier-method (GAhier class), 5
sigEndnodes (GAhier class), 5
sigEndnodes,GAhier-method (GAhier
class), 5

vantVeer, 31