GlobalAncova

April 19, 2009

`Multiple testing on the GO graph`

Multiple testing on the GO graph

Description

Three functions adapted from package **globaltest** 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

```
GAGO(..., GO, focus, maxalpha = 0.05, stopafter = 100, verbose = FALSE)
```

Arguments

`...`	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.
`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.

Details

Previous to a call to `GAGO`, 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 `GAGO` 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

Function `GAGO` corresponds to function `gtGO` in package **globaltest**. The difference is in the use of the `GlobalAncova` test instead of `globaltest` within the focus level procedure.

Author(s)

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

References

Goeman, J.J. and Mansmann, U., Family-wise error rate on the directed acyclic graph of Gene Ontology, submitted.

See Also

`gtGO`, `GlobalAncova`, `globaltest`, `GOstructure`, `makeGOstructure`, `getFocus`

Examples

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

`GlobalAncova` *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", cov

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", cova
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", cova
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` *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 =

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

## 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 =
```

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", cov

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", cova
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", cova
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 ⟨hummel@ibe.med.uni-muenchen.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 = ~ERstatu
GlobalAncova(xx = vantVeer, formula.full = ~metastases + ERstatus, test.terms = "metastas
GlobalAncova(xx = vantVeer, group = phenodata$metastases, covars = phenodata$ERstatus, te
```

GlobalAncova.closed

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", grou
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.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,
          previous.test, level, formula.full, formula.red, model.dat, method = c
          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,
          previous.test, level, formula.full, test.terms, model.dat, method = c(
          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,
          previous.test, level, group, covars = NULL, method = c("permutation","
          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", grou
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@tfh-berlin.de⟩
Ulrich Mansmann ⟨mansmann@ibe.med.uni-muenchen.de⟩
Manuela Hummel ⟨hummel@ibe.med.uni-muenchen.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.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", "typ
```

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 =
GlobalAncova.decomp(xx = vantVeer, formula = ~ grade + metastases + ERstatus, model.dat =

# 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
```

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 ⟨hummel@ibe.med.uni-muenchen.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
```

Plot.genes *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", cov
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", cova
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", cova
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.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, test.genes, Colorgroup = NU

## S4 method for signature 'matrix, formula, missing, ANY,
##   missing, missing, character':
Plot.genes(xx, formula.full, test.terms, model.dat, test.genes, Colorgroup = NUL

## S4 method for signature 'matrix, missing, missing,
##   missing, ANY, ANY, missing':
Plot.genes(xx, group, covars = NULL, test.genes, Colorgroup = NULL, legendpos =
```

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", cov

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", cova
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", cova
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 ⟨hummel@ibe.med.uni-muenchen.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,
Plot.genes(xx = vantVeer, formula.full = ~metastases + ERstatus, test.terms = "metastases
Plot.genes(xx = vantVeer, group = phenodata$metastases, covars = phenodata$ERstatus, test
```

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 ⟨hummel@ibe.med.uni-muenchen.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 = phenodat
```

Plot.subjects *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", cov

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", cova
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", cova
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.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, test.genes, Colorgroup =

## S4 method for signature 'matrix, formula, missing, ANY,
##   missing, missing, character':
Plot.subjects(xx, formula.full, test.terms, model.dat, test.genes, Colorgroup =

## S4 method for signature 'matrix, missing, missing,
##   missing, ANY, ANY, missing':
Plot.subjects(xx, group, covars = NULL, test.genes, Colorgroup = NULL, sort = FA
```

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", cov
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", cova
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", cova
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 ⟨hummel@ibe.med.uni-muenchen.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 = ~ERstat
Plot.subjects(xx = vantVeer, formula.full = ~metastases + ERstatus, test.terms = "metasta
Plot.subjects(xx = vantVeer, group = phenodata$metastases, covars = phenodata$ERstatus, t
```

`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)
```

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 = 100
```

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 ⟨hummel@ibe.med.uni-muenchen.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, te
```

---

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)
```

---

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