maigesPack

April 19, 2010

`activeMod` *Functional classification of gene groups*

Description

This function calculate profiles of expression for groups of genes in each sample of the dataset and for each biological condition (group of samples).

Usage

```
activeMod(data=NULL, gNameID="GeneName", samples=NULL, usePaths=FALSE,
          sLabelID="Classification", adjP="none", cutExp=1,
          cutPhiper=0.05)
```

Arguments

`data`	object of class `maiges` to be used to functionally classify gene groups stored in `GeneGrps` slot.
`gNameID`	character string specifying identification of gene label to be used.
`sLabelID`	idem to the previous argument for identification of sample label.
`samples`	a list with character vectors specifying the groups that must be compared.
`usePaths`	logical specifying if the pathways given in `Paths` slot must also be used, defaults to FALSE.
`cutExp`	real number specifying the cutoff for expression levels (to discretise the expression)
`adjP`	character string giving the type of p-value adjustment. May be 'Bonferroni', 'Holm', 'Hochberg', 'SidakSS', 'SidakSD', 'BH', 'BY' or 'none'. Defaults to 'none'. See function `mt.rawp2adjp` in package multtest for more details.
`cutPhiper`	p-value cutoff to select significant gene groups.

Details

If the argument `samples` is NULL, all types defined by the sample label given by `sLabelID` are used. It is possible to use the `plot.maigesActMod` and `image.maigesActMod` to display the results of this analysis. This function is based in the method proposed by Segal et al. (2004).

Value

The result of this function is an object of class maigesActMod.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Segal, E.; Friedman, N.; Koller, D. and Regev, A. A module map showing conditional activity of expression modules in cancer. **Nature Genetics**, 36, 1090-1098, 2004. (http://www.nature.com/ng/journal/v36/n10/abs/ng1434.html)

See Also

activeModScoreHTML, maigesActMod, plot.maigesActMod, image.maigesActMod, mt.rawp2adjp

Examples

```
## Loading a little dataset
data(gastro)

## Doing functional classification of gene groups for 'Tissue' sample label
gastro.mod = activeMod(gastro.summ, sLabelID="Tissue", cutExp=1,
  cutPhiper=0.05)

## Doing functional classification of gene groups together with the
## networks given by Paths slot for 'Tissue' sample label. Also we are
## using a cuttoff for p-value of hipergeometric test as 0.1
gastro.mod = activeMod(gastro.summ, sLabelID="Tissue", cutExp=1,
  cutPhiper=0.1, usePaths=TRUE)
```

---

activeModScoreHTML *Save HTML file with global gene scores from functional gene groups classification*

Description

This function takes an object of class maigesActMod, that is generated using the function activeMod to do functional classification of gene groups, and save an HTML file with global score for genes separated by gene groups (modules).

Usage

```
activeModScoreHTML(mod=NULL, dir="./", fileSave="scores")
```

Arguments

mod	object of class maigesActMod, resulted from functional classification of gene groups.
dir	character string giving the folder to save the file.
fileSave	string giving the file name. You don't need to put the extension 'html', it will be put automatically.

Value

This function generates an HTML file and don't return any R value or object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

plot.maigesActMod, activeMod

Examples

```
## Loading a little dataset
data(gastro)

## Doing functional classification of gene groups for 'Tissue' sample label
gastro.mod = activeMod(gastro.summ, sLabelID="Tissue", cutExp=1,
  cutPhiper=0.05)

activeModScoreHTML(gastro.mod)
```

---

activeNet *Functional classification of gene networks*

Description

This function calculate a statistic for each gene network in each biological condition that measure the profile of activation of the network in that condition. Also the function measures the significance of the results.

Usage

```
activeNet(data=NULL, samples=NULL, sLabelID="Classification",
          type="Rpearson", bRep=1000, alternative = "greater",
          adjP="none")
```

Arguments

data	object of class maiges to be used to functionally classify gene networks stored in Paths slot.
sLabelID	character string specifying identification of sample label to be used.
samples	a list with character vectors specifying the groups that must be compared.
type	character string giving the type of correlation to be calculated. May be 'Rpearson' (default), 'pearson', 'kendall', 'spearman' or 'MI'.
bRep	integer number specifying the bootstraps to be done in the correlation test.
alternative	character string specifying the alternative hypotheses. May be 'greater' (default) to test the activity of the networks in accordance to the to the graph or 'less' to test the activity of the network antagonic to the graph.
adjP	character string giving the type of p-value adjustment. May be 'Bonferroni', 'Holm', 'Hochberg', 'SidakSS', 'SidakSD', 'BH', 'BY' or 'none'. Defaults to 'none'. See function mt.rawp2adjp in package multtest for more details.

Details

If the argument `samples` is NULL, all types defined by the sample label given by `sLabelID` are used. It is possible to use the `plot.maigesActNet` and `image.maigesActNet` methods to display the results of this analysis.

Value

The result of this function is an object of class `maigesActNet`.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`activeNetScoreHTML`, `maigesActNet`, `plot.maigesActNet`, `image.maigesActNet`, `mt.rawp2adjp`

Examples

```
## Loading the dataset
data(gastro)

## Doing functional classification of gene networks for sample Label
## given by 'Tissue'
gastro.net = activeNet(gastro.summ, sLabelID="Tissue")
```

---

activeNetScoreHTML *Save HTML file with scores and p-values from functional gene networks classification*

Description

This function takes an object of class `maigesActNet`, that is generated using the function `activeNet` to do functional classification of gene groups, and save an HTML file with global score for genes separated by gene groups (modules).

Usage

```
activeNetScoreHTML(mod=NULL, dir="./", fileSave="scores")
```

Arguments

`mod`	object of class `maigesActNet`, resulted from functional classification of gene networks.
`dir`	character string giving the folder to save the file.
`fileSave`	string giving the file name. You don't need to put the extension 'html', it will be put automatically.

Value

This function generates an HTML file and don't return any R value or object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

plot.maigesActNet, activeNet

Examples

```
## Loading the dataset
data(gastro)

## Doing functional classification of gene networks for sample Label
## given by 'Tissue'
gastro.net = activeNet(gastro.summ, sLabelID="Tissue")

activeNetScoreHTML(gastro.net)
```

---

addGeneGrps *Function to load gene groups into maigesPreRaw object*

Description

This function read a directory and read files containing the genes for specific gene groups. This files must have one gene per line. This function stores the gene groups read in the slot GeneGrps into objects of class maigesPreRaw.

Usage

```
addGeneGrps(data, folder="./", ext=".txt")
```

Arguments

data	object of maigesPreRaw class.
folder	char string specifying the directory of gene groups. The function tests the presence or not of the final bar.
ext	string giving the extension of the files, defaults to '.txt'. The function also tests the presence of the initial dot.

Details

If the data object already has gene groups with names equal to some someones that are been read, the groups with repeated names are not added. Warning messages are printed for every repeated group name.

The folder directory must contain only one file for each gene group of interest. These files must discriminate one gene per line. The identification of the genes must be done by one of the gene labels given by genemap (see loadData).

Value

This function returns another object of class maigesPreRaw, with the slot GeneGroups actualised.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

maigesPreRaw, addPaths

Examples

```
## Don't run because you don't have the gene sets in a readable folder.
## Not run:
gastro = addGeneGrps(gastro, folder="geneGrps", ext="txt")

## End(Not run)
```

---

addPaths *Function to load gene pathways into maigesPreRaw object*

Description

This function read a directory and read files containing the gene pathways in TGF format. This format must have the genes of the pathway sequentially in lines numbered from 1, followed by a '\#' character that separate the nodes (given by genes) from edges, that must be specified as number of the origin gene followed by a space, the number of the final gene, another space and the weight of the iteration. This function stores the gene networks read in the slot `Paths` into objects of class maigesPreRaw.

Usage

```
addPaths(data, folder="./", ext=".tgf")
```

Arguments

`data`	object of maigesPreRaw class.
`folder`	char string specifying the directory of gene groups. The function tests the presence or not of the final bar.
`ext`	string giving the extension of the files, defaults to '.tgf'. The function also tests the presence of the initial dot.

Details

If the `data` object already has gene networks with names equal to some someones that are been read, the nets with repeated names are not added. Warning messages are printed for every repeated group name.

The `folder` directory must contain only one file for each pathway of interest. These files must be done in TGF format, as described into description above. The gene identification are matched with some column from `genemap` (see loadData).

Value

This function return another object of class maigesPreRaw, with the slot `Paths` actualised.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

maigesPreRaw, addGeneGrps

Examples

```
## Don't run because you don't have the pathways in a readable folder.
## Not run:
gastro = addPaths(gastro, folder="geneNets", ext="tgf")

## End(Not run)
```

---

bootstrapCor *Calculate bootstrap p-values for correlation measures*

Description

This function takes a numerical matrix (or two vectors) and calculates bootstrapped (by permutation) p-values to test if the correlation value is equal to zero. If the first argument is a matrix, the p-values are calculated between all pairs of rows of the matrix.

Usage

```
bootstrapCor(x, y=NULL, bRep, type="Rpearson", ret="p-value",
             alternative="two.sided")
```

Arguments

x	numerical matrix or vector to be analysed. If a vector, the argument y must be informed.
y	numerical vector. Must be informed if x is a vector. If x is a matrix, this argument is ignored. Defaults to NULL.
bRep	number of permutation to be done in the test.
type	character string specifying the type of correlation statistic to be used. Possible values are 'Rpearson', 'pearson', 'spearman' or 'kendall'.
ret	character string with the value to return. Must be 'p-value' (default) for the usual p-value or 'max', to return the maximum absolute correlation value obtained by the permutation.
alternative	character specifying the type of test to do, must be 'two.sided' (default), 'less' or 'greater'.

Details

Pearson, spearman and kendall types of correlation values are calculated by cor function from package stats. The method Rpearson was developed in this package and is a generalisation of the *jackniffe* correlation proposed by Heyer et al. (1999), it is calculated using the function robustCorr.

Value

The result of this function is a square matrix (length equal to the number of rows of x) if x is a matrix or a numerical value if x and y are vectors. The result is the p-values or maximum correlation values calculated by permutation tests.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Heyer, L.J.; Kruglyak, S. and Yooseph, S. Exploring expression data: identification and analysis of coexpressed genes, **Genome Research**, 9, 1106-1115, 1999 (http://www.genome.org/cgi/content/full/9/11/1106)

See Also

cor, robustCorr

Examples

```
x <- runif(50, 0, 1)
y <- rbeta(50, 1, 2)
bootstrapCor(x, y, bRep=100)

z <- matrix(rnorm(100, 0, 1), 4, 25)
bootstrapCor(z, bRep=100)
```

---

bootstrapMI *Calculate bootstrap p-values for mutual information (MI) measures*

Description

This function takes a numerical matrix (or two vectors) and calculates bootstrapped (by permutation) p-values to test if the mutual information value is equal to zero. If the first argument is a matrix, the p-values are calculated between all pairs of rows of the matrix.

Usage

```
bootstrapMI(x, y=NULL, bRep, ret="p-value")
```

Arguments

x	numerical matrix or vector to be analysed. If a vector, the argument y must be informed.
y	numerical vector. Must be informed if x is a vector. If x is a matrix, this argument is ignored. Defaults to NULL.
bRep	number of permutation to be done in the test.
ret	character string with the value to return. Must be 'p-value' (default) for the usual p-value or 'max', to return the maximum absolute correlation value obtained by the permutation.

Details

The method implemented in this function is proposed by Butte and Kohane (2000). The MI value is calculated using the function `MI`.

Value

The result of this function is a square matrix (length equal to the number of rows of `x`) if `x` is a matrix or a numerical value if `x` and `y` are vectors. The result is the p-values or maximum MI values calculated by permutation tests.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Butte, A.J. and Kohane, I.S. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In Pacific Symposium on Biocomputing, 5, 415-426, 2000 (http://psb.stanford.edu/psb-online/proceedings/psb00/)

See Also

`MI`

Examples

```
x <- runif(50, 0, 1)
y <- rbeta(50, 1, 2)
bootstrapMI(x, y, bRep=100)

z <- matrix(rnorm(100, 0, 1), 4, 25)
bootstrapMI(z, bRep=100)
```

---

bootstrapT *Calculate bootstrap p-values for t statistics*

Description

This function takes a numerical matrix and column indexes for two groups to calculate bootstrapped (by re-sampling) p-values comparing the equality of means from the two groups.

Usage

```
bootstrapT(x, k=20000, obs1, obs2, ...)
```

Arguments

`x`	numerical matrix to be bootstrapped. The t statistics is calculated by row using the column indexes given by `obs1` and `obs2` for the two groups tested.
`k`	number of bootstrap re-samplings to be done. Defaults to 20000.
`obs1`	logical or numerical column indexes of the first group.
`obs2`	logical or numerical column indexes of the second group.
`...`	additional parameters for `t.test` function from package *stats*.

Value

The result of this function is a numerical matrix with number of rows given by the rows of the argument `x` and 3 columns. The first column contain the difference of means between the two groups, the second one contain the original t statistic and the last one gives the bootstrapped p-values, for all rows of the matrix `x`.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`t.test` from package *stats*.

Examples

```
z <- matrix(rnorm(100, 0, 1), 4, 25)
bootstrapT(z, k=100, obs1=1:14, obs2=15:25)
```

---

boxplot *Method boxplot for objects defined in this package*

Description

Generic function `boxplot` to display boxplots of the data.

Usage

```
## S3 method for class 'maigesRaw':
boxplot(x, ...)

## S3 method for class 'maiges':
boxplot(x, name=NULL, gLabelID=NULL, sLabelID=NULL, gSamples=NULL, ...)

## S3 method for class 'maigesANOVA':
boxplot(x, name=NULL, gLabelID=NULL, sLabelID=NULL, gSamples=NULL, ...)

## S3 method for class 'maigesDEcluster':
boxplot(x, name=NULL, gLabelID=NULL, sLabelID=NULL, gSamples=NULL, ...)
```

Arguments

`x`	an object of any class defined in this package
`name`	character string specifying a gene to do boxplot of their expression values along the types specified by `sLabelID`.
`gLabelID`	character value giving the name of gene label to be used to search for parameter `name`.
`sLabelID`	idem to `gLabelID`, specifying the name of sample label to be used to separate the gene observations.
`gSamples`	a named list containing character vectors defining groups of samples from `sLabelID`.

`...` additional parameters to `boxplot` method defined in *graphics* package or to `maBoxplot` method defined in *marray* package (for `maigesRaw`, `maiges` or `maigesANOVA` classes), in this case the additional parameters must not be named, because these names conflict with the `boxplot` generic function definition.

Details

This method uses the function `maBoxplot` from marray package to show boxplots of the W values along all the slides of a dataset or along specific accessor methods to stratify the data in objects of class `maigesRaw`. For objects of classes `maiges` or `maigesANOVA` this is also done if the argument `name` is NULL, else the method shows boxplots for the expression values of the gene specified by `name` stratified by sample types from `sLabelID`. For objects of class `maigesDEcluster` only boxplots of genes are produced and the argument `name` may not be NULL.

If you especify the `y` parameter (but not named), defined for the method `maBoxplot` in package *marray*, it will be displayed the M values instead of W.

Author(s)

Gustavo H. Esteves <gesteves@vision.ime.usp.br>

See Also

`maBoxplot` in the *marray* package and `boxplot` in the *graphics* package.

Examples

```
## Loading the dataset
data(gastro)

## To see the boxplots for W values in all chips
boxplot(gastro.raw)  ## maigesRaw class
boxplot(gastro.norm) ## maigesNorm class
boxplot(gastro.summ) ## summarized data (also maigesNorm class)

## To see the boxplots for W values in individual chips
## separating into print tips.
boxplot(gastro.raw[,1])  ## maigesRaw class, first chip
boxplot(gastro.norm[,8]) ## maigesNorm class, 8th chip
boxplot(gastro.summ[,19]) ## summarized data (also maigesNorm class), 19th chip


## Boxplot for individual genes into ANOVA model fitting
gastro.ANOVA = designANOVA(gastro.summ, factors="Tissue")
gastro.ANOVAfit = deGenesANOVA(gastro.ANOVA, retF=TRUE)

boxplot(gastro.ANOVAfit, name="KLK13", gLabelID="GeneName",
sLabelID="Tissue")
```

[-method *Sub-setting methods for maiges objects*

Description

Sub-setting methods were defined for the classes presented in this package, maiges, maigesANOVA, maigesRaw and maigesPreRaw. These methods create instances of the given class, for a subset of spots and/or arrays in a batch.

Methods

x = ANY generic method.

x = maiges `x[i, j]` extract object of class maiges for spots with indexes i and samples with indexes j.

x = maigesANOVA `x[i, j]` extract object of class maigesANOVA for spots with indexes i and samples with indexes j.

x = maigesRaw `x[i, j]` extract object of class maigesRaw for spots with indexes i and arrays with indexes j.

x = maigesPreRaw `x[i, j]` extract object of class maigesPreRaw for spots with indexes i and arrays with indexes j.

See Also

maiges, maigesANOVA, maigesRaw and maigesPreRaw.

Examples

```
## Loading the dataset
data(gastro)

gastro[1:10,]
gastro[1,1]

gastro.raw[rep(TRUE, 15),]

gastro.norm[c(1,4,6), c(10, 18)]
```

calcA *Method calcA to calculate A values*

Description

Generic function calcA to calculate A values from classes of microarray data objects defined in this package.

Usage

```
calcA(object, ...)

## Default S3 method:
calcA(object, ...)

## S3 method for class 'maigesRaw':
calcA(object, bkgSub="subtract", ...)
```

Arguments

object	object of any class. But only some methods are defined in this moment.
bkgSub	character string indicating the type of background subtraction. May be 'none', 'subtract', 'half', 'minimum', 'movingmin', 'edwards', 'normexp' or 'rma'. Uses limma and defaults to 'subtract'.
...	additional parameters for calcA method.

Details

This method receive an object (at moment of class maiges, maigesRaw or maigesANOVA) and returns the matrix of A values. For objects of class maigesRaw it uses the function backgroundcorrect from *limma* package to do background correction before the calculation of A values.

Author(s)

Gustavo H. Esteves <gesteves@vision.ime.usp.br>

See Also

backgroundcorrect in the limma package and calcW.

Examples

```
## Loading the dataset
data(gastro)

calcA(gastro.raw)
calcA(gastro.norm)
```

---

calcW *Method calcW to calculate W values*

Description

Generic function calcW to calculate W values from classes of microarray data objects defined in this package.

Usage

```
calcW(object, ...)

## Default S3 method:
calcW(object, ...)

## S3 method for class 'maigesRaw':
calcW(object, bkgSub="subtract", ...)
```

Arguments

object	object of any class. But only some methods are defined in this moment.
bkgSub	character string indicating the type of background subtraction. May be 'none', 'subtract', 'half', 'minimum', 'movingmin', 'edwards', 'normexp' or 'rma'. Uses limma and defaults to 'subtract'.
...	additional parameters for calcW method.

Details

This method receive an object (at moment of class maiges, maigesRaw or maigesANOVA) and returns the matrix of W values. For objects of class maigesRaw it uses the function backgroundcorrect from *limma* package to do background correction before the calculation of W values.

Author(s)

Gustavo H. Esteves <gesteves@vision.ime.usp.br>

See Also

backgroundcorrect in the limma package and calcA.

Examples

```
## Loading the dataset
data(gastro)

calcW(gastro.raw)
calcW(gastro.norm)
```

---

classifyKNN *Function to do discrimination analysis*

Description

Function to search by groups of few genes, also called cliques, that can discriminate (or classify) between two distinct biological sample types, using the k nearest neighbourhood method. This function uses exhaustive search.

Usage

```
classifyKNN(obj=NULL, sLabelID="Classification", facToClass=NULL,
            gNameID="GeneName", geneGrp=1, path=NULL, nGenes=3, kn=5)
```

Arguments

obj	object of class maiges to search the classifiers.
sLabelID	character string with the identification of the sample label to be used.
facToClass	named list with 2 character vectors specifying the samples to be compared. If NULL (default) the first 2 types of sLabelID are used.
gNameID	character string with the identification of gene label ID.
geneGrp	character or integer specifying the gene group to be tested (colnames of GeneGrps slot). If both geneGrp and path are NULL all genes are used. Defaults to 1 (first group).
path	character or integer specifying the gene network to be tested (names of Paths slot). If both geneGrp and path are NULL all genes are used. Defaults to NULL.
nGenes	integer specifying the number of genes in the clique, or classifier.
kn	number of neighbours for the *knn* method.

Details

Pay attention with the arguments geneGrp and path, if both of them is NULL an exhaustive search for all dataset will be done, and this search may be extremely computational intensive, which may result in a process during some weeks or months depending on the number of genes in your dataset.

If you want to construct classifiers from a group of several genes, the *search and choose* (SC) method may be an interesting option. It is implemented in the function classifyKNNsc. This function uses the function knn.cv from package *class* to construct k-nearest neighbour classifiers. It possible to use functions classifyLDA or classifySVM to construct classifiers using Fisher's linear discriminant analysis or support vector machines methods, respectively.

Value

The result of this function is an object of class maigesClass.

Author(s)

Elier B. Cristo, adapted by Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

knn.cv, classifyKNNsc, classifyLDA, classifySVM.

Examples

```
## Loading the dataset
data(gastro)

## Doing KNN classifier with 2 genes for the 6th gene group comparing
## the 2 categories from 'Type' sample label.
```

```
gastro.class = classifyKNN(gastro.summ, sLabelID="Type",
  gNameID="GeneName", nGenes=2, geneGrp=6)
gastro.class

## To do classifier with 3 genes for the 6th gene group comparing
## normal vs adenocarcinomas from 'Tissue' sample label
gastro.class = classifyKNN(gastro.summ, sLabelID="Tissue",
  gNameID="GeneName", nGenes=3, geneGrp=6,
  facToClass=list(Norm=c("Neso","Nest"), Ade=c("Aeso","Aest")))
```

`classifyKNNsc` *Function to do discrimination analysis, by the search and choose method*

Description

Function to search by groups of few genes, also called cliques, that can discriminate (or classify) between two distinct biological sample types, using the k nearest neighbours method. This function uses the search and choose method.

Usage

```
classifyKNNsc(obj=NULL, sLabelID="Classification", func="wilcox.test",
              facToClass=NULL, gNameID="GeneName", geneGrp=1, path=NULL,
              nGenes=3, cliques=100, kn=5)
```

Arguments

`obj`	object of class `maiges` to search the classifiers.
`sLabelID`	character string with the identification of the sample label to be used.
`func`	string specifying the function to be used to search by the initial one-dimensional classifiers, like 'wilcox.test' or 't.test'.
`facToClass`	named list with 2 character vectors specifying the samples to be compared. If NULL (default) the first 2 types of sLabelID are used.
`gNameID`	character string with the identification of gene label ID.
`geneGrp`	character or integer specifying the gene group to be tested (`colnames` of `GeneGrps` slot). If both `geneGrp` and `path` are NULL all genes are used. Defaults to 1 (first group).
`path`	character or integer specifying the gene network to be tested (`names` of `Paths` slot). If both `geneGrp` and `path` are NULL all genes are used. Defaults to NULL.
`nGenes`	integer specifying the number of genes in the clique, or classifier.
`cliques`	integer specifying the number of cliques or classifiers to be generated.
`kn`	number of neighbours for the *knn* method.

Details

This function implements the method known as Search and choose proposed by Cristo (2003). If you want to use an exhaustive search use the function `classifyKNN`.

This function uses the function `knn.cv` from package *class* to construct k-nearest neighbour classifiers. It is possible to use the functions `classifyLDAsc` and `classifySVMsc` to search by classifiers using Fisher's linear discriminant analysis and support vector machinnes, respectively.

Value

The result of this function is an object of class maigesClass.

Author(s)

Elier B. Cristo, adapted by Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Cristo, E.B. Metodos Estatisticos na Analise de Experimentos de Microarray. Masther's thesis, Instituto de Matematica e Estatistica - Universidade de Sao Paulo, 2003 (in portuguese).

See Also

knn.cv, classifyKNN, classifyLDAsc and classifySVMsc.

Examples

```
## Loading the dataset
data(gastro)

## Doing KNN classifier with 2 genes for the 6th gene group comparing
## the 2 categories from 'Type' sample label.
gastro.class = classifyKNNsc(gastro.summ, sLabelID="Type",
  gNameID="GeneName", nGenes=2, geneGrp=1, cliques=10)
gastro.class

## To do classifier with 3 genes for the 6th gene group comparing
## normal vs adenocarcinomas from 'Tissue' sample label
gastro.class = classifyKNNsc(gastro.summ, sLabelID="Tissue",
  gNameID="GeneName", nGenes=3, geneGrp=1, cliques=10,
  facToClass=list(Norm=c("Neso","Nest"), Ade=c("Aeso","Aest")))
```

---

classifyLDA *Function to do discrimination analysis*

Description

Function to search by groups of few genes, also called cliques, that can discriminate (or classify) between two distinct biological sample types, using the Fisher's linear discriminat analysis. This function uses exhaustive search.

Usage

```
classifyLDA(obj=NULL, sLabelID="Classification", facToClass=NULL,
            gNameID="GeneName", geneGrp=1, path=NULL, nGenes=3,
            sortBy="cv")
```

Arguments

obj	object of class maiges to search the classifiers.
sLabelID	character string with the identification of the sample label to be used.
facToClass	named list with 2 character vectors specifying the samples to be compared. If NULL (default) the first 2 types of sLabelID are used.
gNameID	character string with the identification of gene label ID.
geneGrp	character or integer specifying the gene group to be tested (colnames of GeneGrps slot). If both geneGrp and path are NULL all genes are used. Defaults to 1 (first group).
path	character or integer specifying the gene network to be tested (names of Paths slot). If both geneGrp and path are NULL all genes are used. Defaults to NULL.
nGenes	integer specifying the number of genes in the clique, or classifier.
sortBy	character string with field to sort the result. May be 'cv' (default) or 'svd' for cross validation by leave-one-out or the singular value decomposition, respectively.

Details

Pay attention with the arguments geneGrp and path, if both of them is NULL an exhaustive search for all dataset will be done, and this search may be extremely computational intensive, which may result in a process running during some weeks or months depending on the number of genes in your dataset.

If you want to construct classifiers from a group of several genes, the *search and choose* (SC) method may be an interesting option. It is implemented in the function classifyLDAsc. This function uses the function lda from package *MASS* to search by classifiers using Fisher's linear discriminant analysis. The functions classifySVM and classifyKNN were also dedined to construct classifiers by support vector machines ans k-neighbours, respectively.

Value

The result of this function is an object of class maigesClass.

Author(s)

Elier B. Cristo, adapted by Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

lda, classifySVM, classifyKNN, classifyLDAsc.

Examples

```
## Loading the dataset
data(gastro)

## Doing LDA classifier with 2 genes for the 6th gene group comparing
## the 2 categories from 'Type' sample label.
gastro.class = classifyLDA(gastro.summ, sLabelID="Type",
  gNameID="GeneName", nGenes=2, geneGrp=6)
gastro.class
```

```
## To do classifier with 3 genes for the 6th gene group comparing
## normal vs adenocarcinomas from 'Tissue' sample label
gastro.class = classifyLDA(gastro.summ, sLabelID="Tissue",
  gNameID="GeneName", nGenes=3, geneGrp=6,
  facToClass=list(Norm=c("Neso","Nest"), Ade=c("Aeso","Aest")))
```

`classifyLDAsc` *Function to do discrimination analysis, by the search and choose method*

Description

Function to search by groups of few genes, also called cliques, that can discriminate (or classify) between two distinct biological sample types, using the Fisher's linear discriminant analysis. This function uses the search and choose method.

Usage

```
classifyLDAsc(obj=NULL, sLabelID="Classification", func="wilcox.test",
              facToClass=NULL, gNameID="GeneName", geneGrp=1, path=NULL,
              nGenes=3, cliques=100, sortBy="cv")
```

Arguments

`obj`	object of class `maiges` to search the classifiers.
`sLabelID`	character string with the identification of the sample label to be used.
`func`	string specifying the function to be used to search by the initial one-dimensional classifiers, like 'wilcox.test' or 't.test'.
`facToClass`	named list with 2 character vectors specifying the samples to be compared. If NULL (default) the first 2 types of sLabelID are used.
`gNameID`	character string with the identification of gene label ID.
`geneGrp`	character or integer specifying the gene group to be tested (`colnames` of `GeneGrps` slot). If both `geneGrp` and `path` are NULL all genes are used. Defaults to 1 (first group).
`path`	character or integer specifying the gene network to be tested (`names` of `Paths` slot). If both `geneGrp` and `path` are NULL all genes are used. Defaults to NULL.
`nGenes`	integer specifying the number of genes in the clique, or classifier.
`cliques`	integer specifying the number of cliques or classifiers to be generated.
`sortBy`	character string with the field to be sorted. May be 'cv' (default) or 'svd'.

Details

This function implements the method known as Search and choose proposed by Cristo (2003). If you want to use an exhaustive search use the function `classifyLDA`.

This method uses the function `lda` from package *MASS* to search by classifiers using Fisher's linear discriminant analysis. It is possible to search classifiers by Support Vector Machines and k-nearest neighbour classifiers using the functions `classifySVMsc` and `classifyKNNsc`, respectively.

Value

The result of this function is an object of class maigesClass.

Author(s)

Elier B. Cristo, adapted by Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Cristo, E.B. Metodos Estatisticos na Analise de Experimentos de Microarray. Masther's thesis, Instituto de Matematica e Estatistica - Universidade de Sao Paulo, 2003 (in portuguese).

See Also

lda, classifyLDA, classifySVMsc and classifyKNNsc.

Examples

```
## Loading the dataset
data(gastro)

## Doing LDA classifier with 2 genes for the 6th gene group comparing
## the 2 categories from 'Type' sample label.
gastro.class = classifyLDAsc(gastro.summ, sLabelID="Type",
  gNameID="GeneName", nGenes=2, geneGrp=1, cliques=10)
gastro.class

## To do classifier with 3 genes for the 6th gene group comparing
## normal vs adenocarcinomas from 'Tissue' sample label
gastro.class = classifyLDAsc(gastro.summ, sLabelID="Tissue",
  gNameID="GeneName", nGenes=3, geneGrp=1, cliques=10,
  facToClass=list(Norm=c("Neso","Nest"), Ade=c("Aeso","Aest")))
```

---

classifySVM *Function to do discrimination analysis*

Description

Function to search by groups of few genes, also called cliques, that can discriminate (or classify) between two distinct biological sample types, using the Support Vector Machinne method. This function uses exhaustive search.

Usage

```
classifySVM(obj=NULL, sLabelID="Classification", facToClass=NULL,
            gNameID="GeneName", geneGrp=1, path=NULL, nGenes=3)
```

Arguments

obj	object of class maiges to search the classifiers.
sLabelID	character string with the identification of the sample label to be used.
facToClass	named list with 2 character vectors specifying the samples to be compared. If NULL (default) the first 2 types of sLabelID are used.
gNameID	character string with the identification of gene label ID.
geneGrp	character or integer specifying the gene group to be tested (colnames of GeneGrps slot). If both geneGrp and path are NULL all genes are used. Defaults to 1 (first group).
path	character or integer specifying the gene network to be tested (names of Paths slot). If both geneGrp and path are NULL all genes are used. Defaults to NULL.
nGenes	integer specifying the number of genes in the clique, or classifier.

Details

Pay attention with the arguments geneGrp and path, if both of them is NULL an exhaustive search for all dataset will be done, and this search may be extremely computational intensive, which may result in a process running during some weeks or months depending on the number of genes in your dataset.

If you want to construct classifiers from a group of several genes, the *search and choose* (SC) method may be an interesting option. It is implemented in the function classifySVMsc. This method uses the function svm from package *e1071* to search classifiers by Support Vector Machines. The functions classifyLDA and classifyKNN were also dedined to construct classifiers by Fisher's linear discriminant analysis ans k-neighbours, respectively.

Value

The result of this function is an object of class maigesClass.

Author(s)

Elier B. Cristo, adapted by Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

svm, classifySVMsc, classifyLDA and classifyKNN.

Examples

```
## Loading the dataset
data(gastro)

## Doing SVM classifier with 2 genes for the 6th gene group comparing
## the 2 categories from 'Type' sample label.
gastro.class = classifySVM(gastro.summ, sLabelID="Type",
  gNameID="GeneName", nGenes=2, geneGrp=6)
gastro.class

## To do classifier with 3 genes for the 6th gene group comparing
## normal vs adenocarcinomas from 'Tissue' sample label
gastro.class = classifySVM(gastro.summ, sLabelID="Tissue",
```

```
gNameID="GeneName", nGenes=3, geneGrp=6,
facToClass=list(Norm=c("Neso","Nest"), Ade=c("Aeso","Aest")))
```

`classifySVMsc` *Function to do discrimination analysis, by the search and choose method*

Description

Function to search by groups of few genes, also called cliques, that can discriminate (or classify) between two distinct biological sample types using the Support Vector Machinnes method. This function uses the search and choose method.

Usage

```
classifySVMsc(obj=NULL, sLabelID="Classification", func="wilcox.test",
              facToClass=NULL, gNameID="GeneName", geneGrp=1, path=NULL,
              nGenes=3, cliques=100)
```

Arguments

`obj`	object of class `maiges` to search the classifiers.
`sLabelID`	character string with the identification of the sample label to be used.
`func`	string specifying the function to be used to search by the initial one-dimensional classifiers, like 'wilcox.test' or 't.test'.
`facToClass`	named list with 2 character vectors specifying the samples to be compared. If NULL (default) the first 2 types of sLabelID are used.
`gNameID`	character string with the identification of gene label ID.
`geneGrp`	character or integer specifying the gene group to be tested (`colnames` of `GeneGrps` slot). If both `geneGrp` and `path` are NULL all genes are used. Defaults to 1 (first group).
`path`	character or integer specifying the gene network to be tested (`names` of `Paths` slot). If both `geneGrp` and `path` are NULL all genes are used. Defaults to NULL.
`nGenes`	integer specifying the number of genes in the clique, or classifier.
`cliques`	integer specifying the number of cliques or classifiers to be generated.

Details

This function implements the method known as Search and choose proposed by Cristo (2003). If you want to use an exhaustive search use the function `classifySVM`.

This method uses the function `svm` from package *e1071* to search classifiers by Support Vector Machines. It is possible to search by classifiers using Fisher's linear discriminant analysis and k nearest neighbours methods using the functions `classifyLDAsc` and `classifyKNNsc`, respectively.

Value

The result of this function is an object of class `maigesClass`.

Author(s)

Elier B. Cristo, adapted by Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Cristo, E.B. Metodos Estatisticos na Analise de Experimentos de Microarray. Masther's thesis, Instituto de Matematica e Estatistica - Universidade de Sao Paulo, 2003 (in portuguese).

See Also

svm, classifySVM, classifyLDAsc and classifyKNNsc.

Examples

```
## Loading the dataset
data(gastro)

## Doing SVM classifier with 2 genes for the 6th gene group comparing
## the 2 categories from 'Type' sample label.
gastro.class = classifySVMsc(gastro.summ, sLabelID="Type",
  gNameID="GeneName", nGenes=2, geneGrp=1, cliques=10)
gastro.class

## To do classifier with 3 genes for the 6th gene group comparing
## normal vs adenocarcinomas from 'Tissue' sample label
gastro.class = classifySVMsc(gastro.summ, sLabelID="Tissue",
  gNameID="GeneName", nGenes=3, geneGrp=1, cliques=10,
  facToClass=list(Norm=c("Neso","Nest"), Ade=c("Aeso","Aest")))
```

---

coerce-method *Coerce a maiges object to classes defined by packages limma and marray*

Description

Coercing methods were defined to convert maiges objects of classes maiges and maigesRaw into objects of classes marrayNorm and marrayRaw from package *marray* or classes MAList and RGList from package *limma* and vice-versa.

Usage

```
## S4 method for signature 'maiges,marrayNorm':
as(from, to)

## S4 method for signature 'marrayNorm,maiges':
as(from, to)

## S4 method for signature 'maiges,MAList':
as(from, to)

## S4 method for signature 'MAList,maiges':
as(from, to)
```

```
## S4 method for signature 'maigesRaw,marrayRaw':
as(from, to)

## S4 method for signature 'marrayRaw,maigesRaw':
as(from, to)

## S4 method for signature 'maigesRaw,RGList':
as(from, to)

## S4 method for signature 'RGList,maigesRaw':
as(from, to)
```

Arguments

from	object of class maiges, maigesRaw, MAList, RGList, marrayNorm or marrayRaw.
to	character string specifying the class of object to which the object from will be coerced

Details

When converting from objects of classes RGList or marrayRaw to class maigesRaw, the slot Sf and Sb will always receive the channel 2 (red) values, and Rf and Rb will always receive channel 1 (green) values. For the normalized ojects, the slot W will be equivalent to the M values.

When converting from objects of classes maigesRaw or maiges to classes RGList, marrayRaw, MAList or marrayNorm the correct values of the two channels or of M values are calculated.

Methods

from = maiges, to = marrayNorm convert an object of class maiges into an object of class marrayNorm.

from = marrayNorm, to = maiges convert an object of class marrayNorm into an object of class maiges.

from = maiges, to = MAList convert an object of class maiges into an object of class MAList.

from = MAList, to = maiges convert an object of class MAList into an object of class maiges.

from = maigesRaw, to = marrayRaw convert an object of class maigesRaw into an object of class marrayRaw.

from = marrayRaw, to = maigesRaw convert an object of class marrayRaw into an object of class maigesRaw.

from = maigesRaw, to = RGList convert an object of class maigesRaw into an object of class RGList.

from = RGList, to = maigesRaw convert an object of class RGList into an object of class maigesRaw.

Note

It is possible to use Package *convert* to convert objects between classes defined in packages *limma*, *marray* and *Biobase*.

See Also

as in the *methods* package.

Examples

```
## Loading the dataset
data(gastro)

## Converting a maigesRaw class object into marrayRaw object
as(gastro.raw, "marrayRaw")

## Converting a maigesRaw class object into RGList
as(gastro.raw, "RGList")

## Converting a maiges class object into marrayNorm object
as(gastro.norm, "marrayNorm")

## Converting a maiges class object into MAList object
as(gastro.summ, "MAList")
```

---

compCorr *Compute correlation differences and their p-values*

Description

This function takes two correlation values (or matrices of correlation values) and calculate the differences between these values (term by term) and their respective p-values by a Fisher's Z transformation.

Usage

```
compCorr(n1, r1, n2, r2)
```

Arguments

n1	numerical or matrix of sample sizes for group 1.
r1	numerical or matrix of correlation values for group 1.
n2	numerical or matrix of sample sizes for group 2.
r2	numerical or matrix of correlation values for group 2.

Details

This function use Fisher's Z transformation from scripts adapted from the Internet:
http://ftp.sas.com/techsup/download/stat/compcorr.html
http://www.fon.hum.uva.nl/Service/Statistics/Two_Correlations.html

Value

The result of this function is a list with two numerical items.

diff	matrix (or a single number) of differences between correlation values from two groups
pval	matrix (or a single number) of p-values of differences between the correlation values

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

Examples

```
compCorr(12, 0.9, 16, 0.73)
```

---

contrastsFitM *Compute Contrasts from Linear Model Fit*

Description

Given a linear model fit to microarray data, compute estimated coefficients and standard errors for a given set of contrasts. This function was adapted from `contrasts.fit` of the *limma* package.

Usage

```
contrastsFitM(fit, contrasts)
```

Arguments

`fit`	an `MArrayLM` object or a list object produced by the function `lm.series` or equivalent.
`contrasts`	numeric matrix with row corresponding to coefficients in fit and columns containing contrasts. May be a vector if there is only one contrast.

Details

This function was adapted from the equivalent `contrasts.fit` limma's function to do the linear model fit without use the empirical Bayes method given by the function `eBayes`.

Value

The result of this function is an object of class `MArrayLM`.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>, adapted from the limma's function.

See Also

`lmFit`, `contrasts.fit`, `eBayes`, `MArrayLM`.

createMaigesRaw *Function to create objects of class maigesRaw*

Description

Function to create objects of class maigesRaw from objects of class maigesPreRaw.

Usage

```
createMaigesRaw(PreRaw, greenDataField, greenBackDataField, redDataField,
                redBackDataField, flagDataField, gLabelGrp, gLabelPath)
```

Arguments

PreRaw	object of class maigesPreRaw to be used to generate another object of class maigesRaw.
greenDataField	character string specifying the name of the Data slot from PreRaw that will be used to read the spot intensity values for green channel.
greenBackDataField	character string specifying the name of the Data slot from PreRaw that will be used to read the background intensity values for green channel.
redDataField	character string specifying the name of the Data slot from PreRaw that will be used to read the spot intensity values for red channel.
redBackDataField	character string specifying the name of the Data slot from PreRaw that will be used to read the background intensity values for red channel.
flagDataField	character string specifying the name of the Data slot from PreRaw that will be used to read the flag values.
gLabelGrp	character string with the gene label to match gene groups.
gLabelPath	character string with the gene label to match gene networks.

Value

The result of this function is an object of class maigesRaw.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

loadData, maigesPreRaw,

Examples

```
## Load a little dataset
data(gastro)

## See the object
gastro

## Transform gastro into a maigesRaw object
gastro.raw = createMaigesRaw(gastro, greenDataField="Ch1.Mean",
  greenBackDataField="Ch1.B.Mean", redDataField="Ch2.Mean",
  redBackDataField="Ch2.B.Mean", flagDataField="Flags",
  gLabelGrp="GeneName", gLabelPath="GeneName")
```

---

createTDMS *Create a tab delimited file for TIGR MeV*

Description

This function gets an object of class `maiges` an save a tab delimited file containing the W values to be load by TIGR MeV software, in the format TDMS file.

Usage

```
createTDMS(data=NULL, sLabelID=names(data@Slabels)[1], file="data.txt")
```

Arguments

`data`	object of class `maiges` to be saved as a TDMS file.
`sLabelID`	character string giving the sample label ID to be used to label the samples in the TDMS file. Defaults to the first sample label ID.
`file`	character string specifying the file name to where TDMS file must be saved. Defaults to 'data.txt'.

Value

This function save an ASCII file and do not return any object or value.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

deGenes2by2BootT *Function to do differential expression analysis, comparing only two samples*

Description

This function takes an object of class maiges and do differential expression analysis for the genes onto dataset, comparing only two samples by a bootstrap of t statistics method.

Usage

```
deGenes2by2BootT(data=NULL, sLabelID=names(data@Slabels)[1], sTypeComp=NULL,
                 doClust=TRUE, ...)
```

Arguments

data	object of class maiges.
sLabelID	character string giving the sample label ID to be used.
sTypeComp	list with character vectors specifying the two sample types to be compared.
doClust	logical indicating if the object generated from this analysis will be used for cluster analysis. Defaults to TRUE.
...	additional parameters for functions t.test, wilcox.test or bootstrapT.

Details

This function calculate t statistics and p-values by re-sampling of the data using the function bootstrapT.

There is the option to do the t test directly, using the function deGenes2by2Ttest, or to do the non-parametric Wilcox test using the function deGenes2by2Wilcox.

Value

The result of this function is an object of class maigesDE if doClust if FALSE or of class maigesDEcluster if it is TRUE.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

bootstrapT, deGenes2by2Ttest and deGenes2by2Wilcox.

Examples

```
## Loading the dataset
data(gastro)

## Doing bootstrap from t statistic test fot 'Type' sample label, k=1000
## specifies one thousand bootstraps
gastro.boot = deGenes2by2BootT(gastro.summ, sLabelID="Type", k=1000)
gastro.boot
```

deGenes2by2Ttest *Function to do differential expression analysis, comparing only two samples*

Description

This function takes an object of class maiges and do differential expression analysis for the genes onto dataset, comparing only two samples, by t test.

Usage

```
deGenes2by2Ttest(data=NULL, sLabelID=names(data@Slabels)[1], sTypeComp=NULL,
                 doClust=TRUE, ...)
```

Arguments

data	object of class maiges.
sLabelID	character string giving the sample label ID to be used.
sTypeComp	list with character vectors specifying the two sample types to be compared.
doClust	logical indicating if the object generated from this analysis will be used for cluster analysis. Defaults to TRUE.
...	additional parameters for function t.test.

Details

This function calculate t statistics and p-values for the test of difference in the means of the groups using the function t.test.

There are other two functions, deGenes2by2Wilcox and deGenes2by2BootT, to do non-parametric tests by the Wilcox test an t statistic bootstrap, respectively.

Value

The result of this function is an object of class maigesDE if doClust if FALSE or of class maigesDEcluster if it is TRUE.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

t.test, deGenes2by2Wilcox and deGenes2by2BootT.

Examples

```
## Loading the dataset
data(gastro)

## Doing t test fot 'Type' sample label
gastro.ttest = deGenes2by2Ttest(gastro.summ, sLabelID="Type")
gastro.ttest
```

deGenes2by2Wilcox *Function to do differential expression analysis, comparing only two samples*

Description

This function takes an object of class maiges and do differential expression analysis for the genes onto dataset, comparing only two samples, by the Wilcox (Mann-Withney) test.

Usage

```
deGenes2by2Wilcox(data=NULL, sLabelID=names(data@Slabels)[1], sTypeComp=NULL,
                  doClust=TRUE, ...)
```

Arguments

data	object of class maiges.
sLabelID	character string giving the sample label ID to be used.
sTypeComp	list with character vectors specifying the two sample types to be compared.
doClust	logical indicating if the object generated from this analysis will be used for cluster analysis. Defaults to TRUE.
...	additional parameters for functions t.test, wilcox.test or bootstrapT.

Details

This function calculate Wilcox statistics and p-values for the test comparing the equality of means using the function wilcox.test.

There another function to do parametric t test , deGenes2by2Ttest, and another option of non-parametric test doing bootstrap of t statistics, deGenes2by2BootT.

Value

The result of this function is an object of class maigesDE if doClust if FALSE or of class maigesDEcluster if it is TRUE.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

wilcox.test, deGenes2by2Ttest and deGenes2by2BootT.

Examples

```
## Loading the dataset
data(gastro)

## Doing wilcoxon test fot 'Type' sample label
gastro.wilcox = deGenes2by2Wilcox(gastro.summ, sLabelID="Type")
gastro.wilcox
```

deGenesANOVA *Function to do differential expression analysis, using ANOVA models*

Description

This function takes an object of class maiges and do differential expression analysis for the genes onto dataset, comparing more than two samples using ANalysis Of VAriance (ANOVA) models.

Usage

```
deGenesANOVA(data=NULL, eBayes=FALSE, retOrig=FALSE,
             retF=FALSE, doClust=TRUE, ...)
```

Arguments

data	object of class maigesANOVA.
eBayes	logical indicating the use or not (default) of empirical Bayes statistics implemented in *limma* package.
retOrig	logical indicating if the object of class MArrayLM from *limma* package must be returned. Defaults to FALSE.
retF	logical asking to return the results associated with the F test (TRUE) or with the individual contrasts (FALSE - default).
doClust	logical indicating if the object generated from this analysis will be used for cluster analysis. Defaults to TRUE.
...	additional parameters to function lmFit.

Details

The object of class maigesANOVA of the argument data is created by the function designANOVA. This function calculate statistics and p-values using the function lmFit from package *limma*.

Value

The result of this function is an object of class MArrayLM when retOrig is TRUE. When it is FALSE, the result is an object of class maigesDE if doClust if FALSE or of class maigesDEcluster if it is TRUE.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

designANOVA, lmFit, MArrayLM.

Examples

```
## Loading the dataset
data(gastro)

## Constructing a maigesANOVA object for the 'Tissue' sample label using
## default model (simple linear model with intercept) and contrasts (all
## parameters are equal between themselves)
gastro.ANOVA = designANOVA(gastro.summ, factors="Tissue")

## Fitting the ANOVA model designed by the above command
gastro.ANOVAfit = deGenesANOVA(gastro.ANOVA, retF=TRUE)
gastro.ANOVAfit
```

---

designANOVA *Function to construct design an contrasts matrices for ANOVA models*

Description

This function takes an object of class `maiges` together with other arguments and construct the matrices of design and contrasts for adjusting ANOVA models. The design matrix are generated using the function `model.matrix`.

Usage

```
designANOVA(data=NULL, factors=names(data@Slabels), model=NULL,
            contrasts=NULL, ...)
```

Arguments

`data`	object of class `maiges`.
`factors`	vector of character strings specifying the sample labels IDs to be used as factors in the models.
`model`	a formula specifying the model to be fitted.
`contrasts`	character vector specifying the contrasts to be done. This is done by the function `makeContrasts` from package *limma*. Pay attention that we use the treatment-control parametrisation.
`...`	additional parameters for function `model.matrix`.

Value

The result of this function is an object of class `maigesANOVA`.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`model.matrix`, `makeContrasts`, `deGenesANOVA`.

Examples

```
## Loading the dataset
data(gastro)

## Constructing a maigesANOVA object for the 'Tissue' sample label using
## default model (simple linear model with intercept) and contrasts (all
## parameters are equal between themselves)
gastro.ANOVA = designANOVA(gastro.summ, factors="Tissue")
gastro.ANOVA
```

---

dim *Retrieve the dimension of microarray objects*

Description

Generic function dim to retrieve the number of rows (genes) and columns (arrays) for classes maigesPreRaw, maigesRaw, maiges and maigesANOVA.

Usage

```
## S3 method for class 'maigesPreRaw':
dim(x)

## S3 method for class 'maigesRaw':
dim(x)

## S3 method for class 'maiges':
dim(x)

## S3 method for class 'maigesANOVA':
dim(x)
```

Arguments

x	an object of class maigesPreRaw, maigesRaw, maiges or maigesANOVA

Details

This is a function to retrieve the dimensions of the dataset - number of genes (rows) and samples (columns) studied. Note that the commands nrow(x), ncol(x) and commands related with matrix dimensions on also work.

Value

Vector with 2 numbers, the first one is the number of rows (genes) and the second one the number of columns (samples).

Author(s)

modified from the marray package

See Also

dim in the base package.

Examples

```
## Loading the dataset
data(gastro)

dim(gastro)
dim(gastro.raw)
dim(gastro.norm)
dim(gastro.summ)
```

---

gastro *Gastro-esophagic dataset*

Description

This dataset is composed of five objects of classes maigesPreRaw, maigesRaw and maiges containing a piece of the dataset that was analysed by the MAIGES (http://www.maiges.org/) and published at *Cancer Research* (see the reference below). In this dataset we evaluated the profiles of expression for 72 gastro-esophagical samples. Were used cDNA microarrays with slides containing 4800 spots representing approximately 4400 unique genes. The original dataset was composed from 86 samples with dye swap (what makes 172 chips). The dataset available here contain a subgroup of 20 samples (40 chips) and 500 spots (486 unique genes).

The objects available are: - gastro: maigesPreRaw class containing the raw dataset. - gastro.raw: maigesRaw class containing the dataset ready for filtration and normalization. - gastro.raw2: same to the above, after filtration. - gastro.norm: maiges containing the normalized data. - gastro.summ: same to the above, after summary of replicates.

Usage

```
data(gastro)
```

References

Gomes, L.I.; Esteves, G.H.; Carvalho, A.F.; Cristo, E.B.; Hirata Jr., R.; Martins, W.K.; Marques, S.M.; Camargo, L.P.; Brentani, H.; Pelosof, A.; Zitron, C.; Sallum, R.A.; Montagnini, A.; Soares, F.A.; Neves, E.J. & Reis, L.F. Expression Profile of Malignant and Nonmalignant Lesions of Esophagus and Stomach: Differential Activity of Functional Modules Related to Inflammation and Lipid Metabolism, **Cancer Research**, 65, 7127-7136, 2005 (http://cancerres.aacrjournals.org/cgi/content/abstract/65/16/7127)

getLabels *Method getLabels to pick gene and sample labels*

Description

Generic function getLabels to extract labels given an ID to genes or samples.

Usage

```
getLabels(obj, labelID=NULL, sLabel=TRUE)

## Default S3 method:
getLabels(obj=NULL, labelID=NULL, sLabel=TRUE)

## S3 method for class 'maigesDE':
getLabels(obj=NULL, labelID=NULL, sLabel=TRUE)

## S3 method for class 'maigesDEcluster':
getLabels(obj=NULL, labelID=NULL, sLabel=TRUE)

## S3 method for class 'RGList':
getLabels(obj=NULL, labelID=NULL, sLabel=TRUE)

## S3 method for class 'MAList':
getLabels(obj=NULL, labelID=NULL, sLabel=TRUE)

## S3 method for class 'marrayRaw':
getLabels(obj=NULL, labelID=NULL, sLabel=TRUE)

## S3 method for class 'marrayNorm':
getLabels(obj=NULL, labelID=NULL, sLabel=TRUE)
```

Arguments

obj	object to look for labels. Methods defined for classes maigesRaw, maiges, maigesDE, maigesDEcluster, RGList, MAList, marrayRaw and marrayNorm.
labelID	character string with label name to be searched.
sLabel	logical indicating search in the sample labels, defaults to TRUE. If FALSE search is done for gene labels.

Details

The name of gene labels are done by the names of the slot Glabels in objects of classes maigesRaw or maiges, the slot GeneInfo in objects of classes maigesDE or maigesDEcluster, the slot genes in objects of classes RGList or MAList and the slot maGnames@maInfo in objects of classes marrayRaw or marrayNorm. Equivalently, the name of sample labels are done by the names of the slots Slabels, SampleInfo, targets and maTargets@maInfo.

Author(s)

Gustavo H. Esteves <gesteves@vision.ime.usp.br>

Examples

```
## Loading dataset
data(gastro)

## Getting the 'Tissue' label for samples in maigesRaw class object
getLabels(gastro.raw, "Tissue", sLabel=TRUE)

## Getting the 'Type' label for samples in maigesNorm class object
getLabels(gastro.summ, "Type", sLabel=TRUE)

## Getting the 'GeneName' label for genes (spots) in maigesRaw class object
getLabels(gastro.raw, "GeneName", sLabel=FALSE)

## Getting the 'Annot' label for samples in maigesNorm class object
getLabels(gastro.summ, "Annot", sLabel=FALSE)
```

---

heatmapsM — *Function to plot heatmaps separating groups generated by SOM or k-means*

Description

This function display individual heatmaps for each group constructed by divisive clustering techniques, like SOM or k-means. It is possible to add hierarchical dendrograms to each group individually.

Usage

```
heatmapsM(data, groups, sampleT=NULL, doHier=FALSE, distfun=dist,
          hclustfun=hclust, ...)
```

Arguments

`data`	numeric matrix to be displayed, including all observations from all groups.
`groups`	numeric or character vector (with length equal to columns or rows of data) with identification of the groups to be separated. If numeric must contain the indexes and if character must contain the colnames or rownames.
`sampleT`	list with 2 vectors, the first one specifying the first character of each sample label to be coloured according to the colours specified in the second vector.
`doHier`	logical indicating to plot or not the hierarchical branch in the other dimension of the matrix.
`distfun`	the function to be used for distance calculation.
`hclustfun`	the function for to be used for hierarchical cluster calculation.
`...`	additional parameters for `image` function.

Details

This function is used internally by the functions hierM, somM and kmeansM. If you want to the hierarchical dendrogram in the other dimension, pay attention to the number of observations in this dimension, because the calculation of the hierarchical branch may be very time consuming!

Value

This function display the heatmaps and don't return any object or value.

Author(s)

Elier B. Cristo, addapted by Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

hierM, somM, kmeansM

---

hierMde *Function to do hierarchical cluster analysis*

Description

This is a function to do hierarchical clustering analysis for objects of classe maigesDEcluster.

Usage

```
hierMde(data, group=c("C", "R", "B")[1], distance="correlation",
        method="complete", doHeat=TRUE, sLabelID="SAMPLE",
        gLabelID="GeneName", idxTest=1, adjP="BH",
        nDEgenes=0.05, ...)
```

Arguments

data	object of class maigesDEcluster.
group	character string giving the type of grouping: by rows 'R', columns 'C' (default) or both 'B'.
distance	char string giving the type of distance to use. Here we use the function Dist and the possible values are 'euclidean', 'maximum', 'manhattan', 'canberra', 'binary', 'pearson', 'correlation' (default) and 'spearman'.
method	char string specifying the linkage method for the hierarchical cluster. Possible values are 'ward', 'single', 'complete' (default), 'average', 'mcquitty', 'median' or 'centroid'
doHeat	logical indicating to do or not the heatmap. If FALSE, only the dendrogram is displayed.
sLabelID	character string specifying the sample label ID to be used to label the samples.
gLabelID	character string specifying the gene label ID to be used to label the genes.
idxTest	numerical index of the test to be used to sort the genes when clustering objects of class maigesDEcluster.

`adjP`	string specifying the method of p-value adjustment. May be 'none', 'Bonferroni', 'Holm', 'Hochberg', 'SidakSS', 'SidakSD', 'BH', 'BY'.
`nDEgenes`	number of DE genes to be selected. If a real number in (0,1) all genes with p.value <= `nDEgenes` will be used. If an integer, the `nDEgenes` genes with smaller p-values will be used.
`...`	additional parameters for `heatmap` function.

Details

This function implements the hierarchical clustering method for objects resulted from differential expression analysis. The default function for hierarchical clustering is the `hclust`. For the adjustment of p-values in the selection of genes differentially expressed, we use the function `mt.rawp2adjp` from package *multtest*.

Value

This function display the heatmaps and don't return any object or value.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`somM` and `kmeansM` for displaying SOM and k-means clusters, respectively.

Examples

```
## Loading the dataset
data(gastro)

## Doing bootstrap from t statistic test fot 'Type' sample label, k=1000
## specifies one thousand bootstraps
gastro.ttest = deGenes2by2Ttest(gastro.summ, sLabelID="Type")

## Hierarchical cluster adjusting p-values by FDR, and showing all genes
## with p-value < 0.05
hierMde(gastro.ttest, sLabelID="Type", adjP="BH", nDEgenes=0.05)
```

`hierM` *Function to do hierarchical cluster analysis*

Description

This is a function to do hierarchical clustering analysis for objects of classes `maiges`, `maigesRaw` and `maigesANOVA`. Use the function `hierMde` for objects of class `maigesDEcluster`.

Usage

```
hierM(data, group=c("C", "R", "B")[1], distance="correlation",
      method="complete", doHeat=TRUE, sLabelID="SAMPLE",
      gLabelID="GeneName", rmGenes=NULL, rmSamples=NULL,
      rmBad=TRUE, geneGrp=NULL, path=NULL, ...)
```

Arguments

data	object of class maigesRaw, maiges, maigesANOVA or maigesDEcluster.
group	character string giving the type of grouping: by rows 'R', columns 'C' (default) or both 'B'.
distance	char string giving the type of distance to use. Here we use the function Dist and the possible values are 'euclidean', 'maximum', 'manhattan', 'canberra', 'binary', 'pearson', 'correlation' (default) and 'spearman'.
method	char string specifying the linkage method for the hierarchical cluster. Possible values are 'ward', 'single', 'complete' (default), 'average', 'mcquitty', 'median' or 'centroid'
doHeat	logical indicating to do or not the heatmap. If FALSE, only the dendrogram is displayed.
sLabelID	character string specifying the sample label ID to be used to label the samples.
gLabelID	character string specifying the gene label ID to be used to label the genes.
rmGenes	char list specifying genes to be removed.
rmSamples	char list specifying samples to be removed.
rmBad	logical indicating to remove or not bad spots (slot BadSpots in objects of class maiges, maigesRaw or maigesANOVA).
geneGrp	numerical or character specifying the gene group to be clustered. This is given by the columns of the slot GeneGrps in objects of classes maiges, maigesRaw and maigesANOVA.
path	numerical or character specifying the gene network to be clustered. This is given by the items of the slot Paths in objects of classes maiges, maigesRaw and maigesANOVA.
...	additional parameters for heatmap function.

Details

This function implements the hierarchical clustering method for objects of microarray data defined in this package. The default function for hierarchical clustering is the hclust.

Value

This function display the heatmaps and don't return any object or value.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

somM and kmeansM for displaying SOM and k-means clusters, respectively.

Examples

```
## Loading the dataset
data(gastro)

## Doing a hierarchical cluster using all genes, for maigesRaw class
hierM(gastro.raw, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
```

```
      sLabelID="Sample", gLabelID="Name", doHeat=FALSE)

## Doing a hierarchical cluster using all genes, for maigesNorm class
hierM(gastro.norm, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
      sLabelID="Sample", gLabelID="Name", doHeat=FALSE)

## If you want to show the heatmap do
hierM(gastro.norm, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
      sLabelID="Sample", gLabelID="Name", doHeat=TRUE)

## If you want to show the hierarchical branch in both margins do
hierM(gastro.summ, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
      sLabelID="Sample", gLabelID="Name", doHeat=TRUE, group="B")

## If you want to use euclidean distance only into rows (spots or genes)
hierM(gastro.summ, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
      sLabelID="Sample", gLabelID="Name", doHeat=FALSE, group="R", distance="euclidean")
```

`image` *Method image for objects defined in this package*

Description

Generic function `image` to display colour maps of numerical values stored in objects defined in this package.

Usage

```
## S3 method for class 'maigesRaw':
image(x, ...)

## S3 method for class 'maiges':
image(x, ...)

## S3 method for class 'maigesANOVA':
image(x, ...)

## S3 method for class 'maigesRelNetB':
image(x=NULL, name=NULL, ...)

## S3 method for class 'maigesRelNetM':
image(x=NULL, names=NULL, ...)

## S3 method for class 'maigesActMod':
image(x, type=c("S","C")[2], keepEmpty=FALSE, ...)

## S3 method for class 'maigesActNet':
image(x, type=c("score","p-value")[1], ...)
```

Arguments

`x`	an object of class `maigesRaw`, `maiges`, `maigesANOVA`, `maigesRelNetB`, `maigesRelNetM`, `maigesActMod` or `maigesActNet` defined in this package.

name	character string giving a name for sample type tested to be plotted as a name in the method for class maigesRelNetB.
names	similar to the previous one, but it is a vector of length 3 for class maigesRelNetM.
type	string specifying the type of colour map to be plotted. For class maigesActMod it must be 'S' or 'C' for samples or biological conditions, respectively. For class maigesActNet it must be 'score' or 'p-value' for the statistics or p-values of the tests, respectively.
keepEmpty	logical, if true the results of all gene groups are displayed, else only the gene groups that present at least one significant result are displayed.
...	additional arguments for the generic method image from *graphics* package or maImage defined in package *marray* (for maigesRaw, maiges or maigesANOVA classes), in this case the additional parameters must not be named, because these names conflict with the boxplot generic function definition.

Details

This method uses the function maImage from marray package to display colour maps of accessor methods defined into *marray* package for objects of class maiges or maigesRaw and maigesANOVA.

For objects of class maigesRelNetM the method displays 3 colour maps representing the correlation values for the two groups tested and the p-values of the tests.

For class maigesRelNetB it displays the correlation values for the type tested.

In objects of class maigesActMod it displays the fraction of genes induced or repressed for each gene group, by samples or biological type.

Finally, for class maigesActNet, the method display the matrix of statistics or p-values of the tests.

Pay attention, if you specify the parameter *x* (but not named) for maImage) it will plot the M values instead of W (default).

Author(s)

Gustavo H. Esteves <gesteves@vision.ime.usp.br>

See Also

image in the *graphics* package and maImage in *marray* package.

Examples

```
## Loading the dataset
data(gastro)

## Doing image plots (using package marray), default method showing the
## W values (for 1st chip), after showing the A values (2nd chip) and
## red background (20th chip).
image(gastro.raw[,1])
image(gastro.raw[,2], "maA")
image(gastro.raw[,20], "maRb")

## Example for normalized objects (showing A values for the 5th chip).
image(gastro.norm[,5], "maA")
```

```
## Example for object of class maigesRelNetB

## Constructing the relevance network (Butte's method) for sample
## 'Tissue' equal to 'Neso' for the 1st gene group
gastro.net = relNetworkB(gastro.summ, sLabelID="Tissue",
  samples="Neso", geneGrp=1, type="Rpearson")

image(gastro.net)



## Example for object of class maigesRelNetM

## Constructing the relevance network for sample
## 'Tissue' comparing 'Neso' and 'Aeso' for the 1st gene group
gastro.net = relNetworkM(gastro.summ, sLabelID="Tissue",
  samples = list(Neso="Neso", Aeso="Aeso"), geneGrp=11,
  type="Rpearson")

image(gastro.net)
```

`kmeansMde` *Function to do k-means cluster analysis*

Description

This is a function to do k-means clustering analysis for objects of class `maigesDEcluster`.

Usage

```
kmeansMde(data, group=c("C", "R")[1], distance="correlation",
          method="complete", sampleT=NULL, doHier=FALSE, sLabelID="SAMPLE",
          gLabelID="GeneName", idxTest=1, adjP="none", nDEgenes=0.05, ...)
```

Arguments

`data`	object of class `maigesDEcluster`.
`group`	character string giving the type of grouping: by rows 'R' or columns 'C' (default).
`distance`	char string giving the type of distance to use. Here we use the function `Dist` and the possible values are 'euclidean', 'maximum', 'manhattan', 'canberra', 'binary', 'pearson', 'correlation' (default) and 'spearman'.
`method`	char string specifying the linkage method for the hierarchical cluster. Possible values are 'ward', 'single', 'complete' (default), 'average', 'mcquitty', 'median' or 'centroid'
`sampleT`	list with 2 vectors. The first one specify the first letter of different sample types to be coloured by distinct colours, that are given in the second vector. If NULL (default) no colour is used.

doHier	logical indicating if you want to do the hierarchical branch in the opposite dimension of clustering. Defaults to FALSE.
sLabelID	character string specifying the sample label ID to be used to label the samples.
gLabelID	character string specifying the gene label ID to be used to label the genes.
idxTest	numerical index of the test to be used to sort the genes when clustering objects of class maigesDEcluster.
adjP	string specifying the method of p-value adjustment. May be 'none', 'Bonferroni', 'Holm', 'Hochberg', 'SidakSS', 'SidakSD', 'BH', 'BY'.
nDEgenes	number of DE genes to be selected. If a real number in (0,1) all genes with p.value <= nDEgenes will be used. If an integer, the nDEgenes genes with smaller p-values will be used.
...	additional parameters for Kmeans function.

Details

This function implements the k-means clustering method for objects resulted from differential analysis. The method uses the function Kmeans from package *amap*. For the adjustment of p-values in the selection of genes differentially expressed, we use the function mt.rawp2adjp from package *multtest*.

Value

This function display the heatmaps and return invisibly a list resulted from the function Kmeans.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

Kmeans from package *amap*. mt.rawp2adjp from package *multtest*. somM and hierM for displaying SOM and hierarchical clusters, respectively.

Examples

```
## Loading the dataset
data(gastro)

## Doing bootstrap from t statistic test fot 'Type' sample label, k=1000
## specifies one thousand bootstraps
gastro.ttest = deGenes2by2Ttest(gastro.summ, sLabelID="Type")

## K-means cluster with 2 groups adjusting p-values by FDR, and showing all genes
## with p-value < 0.05
kmeansMde(gastro.ttest, sLabelID="Type", adjP="BH", nDEgenes=0.05, centers=2)

## K-means cluster with 3 groups adjusting p-values by FDR, and showing all genes
## with p-value < 0.05
kmeansMde(gastro.ttest, sLabelID="Type", adjP="BH", nDEgenes=0.05, centers=3)

dev.off()
```

kmeansM *Function to do k-means cluster analysis*

Description

This is a function to do k-means clustering analysis for objects of classes maiges, maigesRaw and maigesANOVA. Use the function kmeansMde for objects of class maigesDEcluster.

Usage

```
kmeansM(data, group=c("C", "R")[1], distance="correlation",
        method="complete", sampleT=NULL, doHier=FALSE, sLabelID="SAMPLE",
        gLabelID="GeneName", rmGenes=NULL, rmSamples=NULL, rmBad=TRUE,
        geneGrp=NULL, path=NULL, ...)
```

Arguments

data	object of class maigesRaw, maiges or maigesANOVA.
group	character string giving the type of grouping: by rows 'R' or columns 'C' (default).
distance	char string giving the type of distance to use. Here we use the function Dist and the possible values are 'euclidean', 'maximum', 'manhattan', 'canberra', 'binary', 'pearson', 'correlation' (default) and 'spearman'.
method	char string specifying the linkage method for the hierarchical cluster. Possible values are 'ward', 'single', 'complete' (default), 'average', 'mcquitty', 'median' or 'centroid'
sampleT	list with 2 vectors. The first one specify the first letter of different sample types to be coloured by distinct colours, that are given in the second vector. If NULL (default) no colour is used.
doHier	logical indicating if you want to do the hierarchical branch in the opposite dimension of clustering. Defaults to FALSE.
sLabelID	character string specifying the sample label ID to be used to label the samples.
gLabelID	character string specifying the gene label ID to be used to label the genes.
rmGenes	char list specifying genes to be removed.
rmSamples	char list specifying samples to be removed.
rmBad	logical indicating to remove or not bad spots (slot BadSpots in objects of class maiges, maigesRaw or maigesANOVA).
geneGrp	numerical or character specifying the gene group to be clustered. This is given by the columns of the slot GeneGrps in objects of classes maiges, maigesRaw and maigesANOVA.
path	numerical or character specifying the gene network to be clustered. This is given by the items of the slot Paths in objects of classes maiges, maigesRaw and maigesANOVA.
...	additional parameters for Kmeans function.

Details

This function implements the k-means clustering method for objects of microarray data defined in this package. The method uses the function `Kmeans` from package *amap*.

Value

This function display the heatmaps and return invisibly a list resulted from the function `Kmeans`.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`Kmeans` from package *amap*. `somM` and `hierM` for displaying SOM and hierarchical clusters, respectively.

Examples

```
## Loading the dataset
data(gastro)

## Doing a K-means cluster with 2 groups using all genes, for maigesRaw class
kmeansM(gastro.raw, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
        sLabelID="Sample", gLabelID="Name", centers=2)

## The same as above, but for maigesNorm class
kmeansM(gastro.norm, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
        sLabelID="Sample", gLabelID="Name", centers=2)

## Another example with 3 groups
kmeansM(gastro.norm, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
        sLabelID="Sample", gLabelID="Name", centers=3)

## If you want to use euclidean distance to group genes (or spots) with
## 4 groups
kmeansM(gastro.summ, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
        sLabelID="Sample", gLabelID="Name", centers=4, group="R", distance="euclidean")
```

---

loadData *Load cDNA microarray data tables*

Description

This function loads a cDNA microarray dataset into a temporary `maigesPreRaw` object.

Usage

```
loadData(fileConf=paste(R.home(), "library/maiges/doc/gastro/load_gastro.conf",
```

Arguments

`fileConf` string specifying a file name containing the parameters to load data. This file must contain all the information necessary to load the data, which are the following:

***dataDir*:** specify a folder name containing the data files to be loaded. The function tests the presence or not of the final bar.

***ext*:** string specifying the extension of the tables (if the *sampleFile* below don't contain this information). You don't need to put the *dot* onto string beginning, the function tests this automatically.

***sampleFile*:** string containing the file name with the descriptions of the biological samples hybridised, including the respective intensity data files. This file must be spreadsheet-like separated by tabs in a plain text format. The column fields **'File'** and **'Ref'** are mandatory (with exactly these names). The first one describes the files containing the numerical data and the second one describes the channel used to label the reference sample, must be 'green' or 'red' and they are not case sensitive.

***datasetId*:** string with a dataset identification.

***geneMap*:** as in *sampleFile*, this item is a character string giving a file name. This file must describe the genes on the slides. Also it must be a plain text spreadsheet-like separated by tabs. There are no mandatory field, but it is strongly recommended that you specify some fields containing gene names, genbank ID, cluster ID and gene annotations for a nice gene identification.

***headers*:** character string (in the R format) specifying the column fields from data files you want to load.

***skip*:** number of lines to be skipped in the numeric tables.

***sep*:** character that separates the fields on the numeric tables.

***gridR*:** number of print tip rows inside the slide.

***gridC*:** number of print tip columns inside the slide.

***printTipR*:** number of rows inside each print tip.

***printTipC*:** number of columns inside each print tip.

You can see an example of this configuration file in RHOME/library/maiges/doc/gastro/load_gastro.c

Details

This function takes the file name with initial arguments and load the dataset specified by this config file. It generate a `maigesPreRaw` object. During the process the function writes a file named load.out on your working folder, that is a log of the process, that you can check and verify if all was done correctly. Obviously, the parameters *dataDir*, *sampleFile*, *geneMap*, *sep*, *gridR*, *gridC*, *printTipR*, *printTipC* and *headers* must be specified. All other parameters may be specified as NULL and, if so, they are ignored. It is possible to specify any fields that you want in the headers parameter, but it is strongly recommended that you specify the fields of spot intensity and background for both channels and the filed giving quality weights for all spots.

Value

This function returns a `maigesPreRaw` object containing the dataset loaded.

Once an object of class `maigesPreRaw` was generated, you may use the functions `addGeneGrps` and `addPaths` to load informaation about gene groups and gene networks, respectively.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

Examples

```
## Don't run because you don't have data tables.
## Not run:
gastro = loadData(fileConf="load_gastro.conf")

## End(Not run)
```

---

maigesActMod-class *maigesActMod class, store results of functional classification of gene groups*

Description

This class defines a structure to store the results of functional classification of gene groups (or modules) that was proposed by Segal et al. (2004), see the reference below. Objects of this type are generated by calling the function `activeMod`.

Details

Objects of this class are produced by calling the function `activeMod` over an object of class `maiges` to search for gene groups that present numbers of genes induced or repressed greater than the expected by chance in biological sample types of interest. This is done according to the model proposed by Segal et al. (2004), described below.

Slots

`modBySamp`**:** numerical matrix containing the fraction of genes activated (or negative fraction of genes repressed) for all chips. The rows and columns of the matrix represents the chips and the gene groups used, respectively.

`modByCond`**:** numerical matrix storing the fraction (or negative fraction) of different sample types that presents alteration in the gene groups tested. The rows and columns represents the sample types and gene groups, respectively.

`globalScore`**:** list with the same length as the number of gene groups containing matrices with the genes as rows and 2 columns. The first column gives a global score that measure the consistency of the classification of the gene and the group, the second column gives the p-value for this score, as described in Segal et al. (2004).

`tissueScore`**:** a list similar to the previous one, but having arrays of 3 dimensions that gives scores similar to that one described above, but relating only to each specific sample type. Note that this new score was implemented in this package and not in the original Segal's work.

`Date`**:** character string giving the date and time that the object was generated.

`V.info`**:** list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

image `signature(x = 'maigesActMod')`: image method for `maigesActMod` class. Display colour representation of the fractions of gene groups induced and repressed.

plot `signature(x = 'maigesActMod')`: plot method for `maigesActMod` class. Do the same as image.

print `signature(x = 'maigesActMod')`: print method for `maigesActMod` class.

show `signature(x = 'maigesActMod')`: show method for `maigesActMod` class.

summary `signature(x = 'maigesActMod')`: summary method for `maigesActMod` class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Segal, E.; Friedman, N.; Koller, D. and Regev, A. A module map showing conditional activity of expression modules in cancer. **Nature Genetics**, 36, 1090-1098, 2004. (http://www.nature.com/ng/journal/v36/n10/abs/ng1434.html)

See Also

`activeMod`.

---

maigesActNet-class *maigesActNet class, store results of functional classification of gene networks*

Description

This class defines a structure to store the results of functional classification of gene networks that was proposed and implemented in this package. Objects of this type are generated by calling the function `activeNet`.

Details

Objects of this class are produced by calling the function `activeNet` over an object of class `maiges` to search for gene networks that present evidence of activation in different biological types. This is done according to the model proposed in the PhD thesis of the author of this package.

Slots

`scores`: numerical matrix storing the results of a statistic to test the functional activation of the networks studied. The rows and columns of the matrix represents the biological sample types and the networks, respectively.

`Pvalues`: numerical matrix storing the p-values of the statistical test. As in the previous slot, matrix rows and columns represents the biological sample types and the networks, respectively.

`Date`: character string giving the date and time that the object was generated.

`V.info`: list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

image `signature(x = 'maigesActNet')`: image method for `maigesActNet` class. Display colour representation of the statistics or p-values of gene networks.

plot `signature(x = 'maigesActNet')`: plot method for `maigesActNet` class. Do the same as image.

print `signature(x = 'maigesActNet')`: print method for `maigesActNet` class.

show `signature(x = 'maigesActNet')`: show method for `maigesActNet` class.

summary `signature(x = 'maigesActNet')`: summary method for `maigesActNet` class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`activeNet`.

---

maigesANOVA-class *maigesANOVA class, extend maiges class to fit ANOVA models*

Description

This class extends the class `maiges` adding two numerical matrices to fit ANOVA models and estimate parameters of interest. Additional to the existing slots of class `maiges` this class defines other two, described in *Slots* section.

Details

This class of objects is exactly the same as `maiges` with the two slots described in *Slots* section. Objects of this class are generated from `maiges` class using the function `designANOVA`. So, it is possible to fit the model and estimate parameters or contrasts using the function `deGenesANOVA`.

Slots

`Dmatrix`**:** numeric matrix describing the design matrix to fit an ANOVA model.

`Cmatrix`**:** numeric matrix describing the contrasts matrix to estimate parameters and contrasts of interest, after the model fitting.

Methods

[`signature(x = 'maigesANOVA')`: sub-setting operator for spots on the array or arrays in the batch, ensures that all slots are subset properly.

boxplot `signature(x = 'maigesANOVA')`: boxplot method for `maigesANOVA` class. Display boxplots of the slides and print tip groups using package *marray* or boxplots of one gene previously defined.

dim `signature(x = 'maigesANOVA', value = 'numeric')`: get the dimensions of the object, numeric vector of length two.

image `signature(x = 'maigesANOVA')`: image method for `maigesANOVA` class. Display colour representation of the slides using package *marray*.

plot `signature(x = 'maigesANOVA')`: plot method for `maigesANOVA` class. Display 'MA' plots.

print `signature(x = 'maigesANOVA')`: print method for `maigesANOVA` class.

show `signature(x = 'maigesANOVA')`: show method for `maigesANOVA` class.

summary `signature(x = 'maigesANOVA')`: summary method for `maigesANOVA` class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`designANOVA`, `deGenesANOVA`.

`maigesClass-class` *maigesClass class, store results of discrimination (or classification) analysis*

Description

This class defines a structure to store the results from discrimination analysis. This type of analysis can be done using the functions `classifyLDA`, `classifySVM`, `classifyKNN`, `classifyLDAsc`, `classifySVMsc` or `classifyKNNsc`.

Details

Objects of this class are produced by calling the functions `classifyLDA`, `classifySVM`, `classifyKNN`, `classifyLDAsc`, `classifySVMsc` or `classifyKNNsc` over an object of class `maiges` to search for cliques satisfying the criteria specified for classification.

Slots

`W`: numeric matrix giving the W values of the genes tested. This information is useful for doing plots of the cliques.

`CV`: numeric vector that store the number of correct classifications in the leave-one-out cross validation procedure.

`SVD`: numeric vector that store the singular value decomposition from Fisher linear discriminant analysis.

`cliques`: character matrix that gives the genes that constitute the cliques returned. The rows of the matrix represent the cliques while the columns represent the genes that form the clique.

`cliques.idx`: numeric matrix similar to the above, storing the indexes (onto `W` slot) of the genes.

`method`: character string giving the method of discrimination analysis used.

`Date`: character string giving the date and time that the object was generated.

`V.info`: list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

plot `signature(x = 'maigesClass')`: plot method for `maigesClass` class. Display dispersion plots.

print `signature(x = 'maigesClass')`: print method for `maigesClass` class.

show `signature(x = 'maigesClass')`: show method for `maigesClass` class.

summary `signature(x = 'maigesClass')`: summary method for `maigesClass` class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`classifyLDA`, `classifySVM`, `classifyKNN`, `classifyLDAsc`, `classifySVMsc` and `classifyKNNsc`.

---

maiges-class *maiges class, store normalised microarray datasets*

Description

This class describes objects to handle ratio of intensities (*W*) and average of intensities (*A*) values values and information about genes and samples used in the data. Objects of this class are created from class `maigesRaw` using the functions `normLoc`, `normOLIN`, `normRepLoess`, `normScaleLimma` and/or `normScaleMarray` to do the normalisation.

Here, the *M=log(R/G)* value of intensity ratio was redefined as *W=log(Test/Ref)*, where *Test* and *Ref* are the test and reference samples.

Details

This defines the main class of objects defined in this package. It is created from `maigesRaw` class using the normalisation functions `normLoc`, `normOLIN`, `normRepLoess`, `normScaleLimma` and `normScaleMarray`. From this class of objects it is possible to do any type of analysis defined by several functions in this package. Also, it is possible to summarise spots (or samples) information using the function `summarizeReplicates`.

Slots

`W`: numeric matrix containing the ratio values (in log2 scale) between the test and reference sample intensities (W values). Spots are indexed by rows and samples by columns.

`A`: numeric matrix containing the mean intensity values between test and reference samples (also in log2 scale). Spots corresponding to rows and samples (or chips) corresponding to columns, too.

`SD`: numeric matrix containing the standard deviation of W values when the lowess step is repeated several times.

`IC1`: numeric matrix containing the left margin of confidence interval defined by repeated lowess during the normalisation step.

`IC2`: numeric matrix containing the right margin of confidence interval defined above.

`BadSpots`: logical vector specifying spots that was judged as bad ones. By default this slot is created as a vector of FALSEs with same length as number of spots.

`UseSpots`: logical matrix indexing the spots to be used for normalisation.

`GeneGrps`: a logical matrix with rows representing the spots and columns representing different gene groups. Each column give the index of spots in that gene group.

`Paths`: list containing `graphNEL` objects specifying gene regulatory networks (or pathways). The first object in this list is a char string giving the gene label used to match the genes.

`Layout`: a list containing the number of rows (`gridR`) and columns (`gridC`) of grids, the number of rows (`spotR`) and columns (`spotC`) of spots inside each grid and the total number of spots.

`Glabels`: data frame giving the gene labels. These labels are generally used during the data analysis.

`Slabels`: data frame giving the sample labels. These labels are generally used during the data analysis.

`Notes`: char string that receives any comment about the dataset. The dataset description is stored in this slot.

`Date`: char string giving the date and hour that the object was created.

`V.info`: list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

[`signature(x = 'maiges')`: subsetting operator for spots on the array or arrays in the batch, ensures that all slots are subset properly.

boxplot `signature(x = 'maiges')`: boxplot method for `maiges` class. Display boxplots of the slides and print tip groups using package *marray* or boxplots of one gene previously defined.

dim `signature(x = 'maiges', value = 'numeric')`: get the dimensions of the object, numeric vector of length two.

image `signature(x = 'maiges')`: image method for `maiges` class. Display colour representation of the slides using package *marray*.

plot `signature(x = 'maiges')`: plot method for `maiges` class. Display 'MA' plots.

print `signature(x = 'maiges')`: print method for `maiges` class.

show `signature(x = 'maiges')`: show method for `maiges` class.

summary `signature(x = 'maiges')`: summary method for `maiges` class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`normLoc`, `normOLIN`, `normRepLoess`, `normScaleLimma`, `normScaleMarray` and `summarizeReplicate`

maigesDE-class *maigesDE class, store results of differential gene expression analysis*

Description

This class defines a structure to store the results from differential expression analysis. This type of analysis can be done using the functions deGenes2by2Ttest, deGenes2by2Wilcox and deGenes2by2BootT for analysis between two biological sample types, or function deGenesANOVA, for analysis between more than two sample types using ANOVA models.

Details

Objects of this class are produced by calling the functions deGenes2by2Ttest, deGenes2by2Wilcox and deGenes2by2BootT over an object of class maiges or by calling the function deGenesANOVA over an object of class maigesANOVA fitting an ANOVA model to the microarray dataset.

Slots

GeneInfo: data frame containing the information regarding the genes from the dataset.

SampleInfo: data frame similar to the GeneInfo above, but containing information about the biological sample types used in the analysis.

fold: numerical matrix containing the fold values (mean difference between the sample types) when two sample types were compared. Each matrix column gives one specific test.

stat: numerical matrix giving the statistic of the tests that were done. Each column represents on test.

p.value: numerical matrix giving the p-values of the statistical tests. Again, each column represents different tests.

factors: character string giving the biological factors used in ANOVA model (when ANOVA models were used).

test: character string describing the test done.

Date: character string giving date and time that the object were created.

V.info: list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

plot signature(x = 'maigesDE'): plot method for maigesDE class. Display volcano plots.

print signature(x = 'maigesDE'): print method for maigesDE class.

show signature(x = 'maigesDE'): show method for maigesDE class.

summary signature(x = 'maigesDE'): summary method for maigesDE class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

deGenes2by2Ttest, deGenes2by2Wilcox, deGenes2by2BootT, deGenesANOVA.

maigesDEcluster-class *maigesDEcluster class, store results of differential gene expression analysis*

Description

This class extends the class maigesDE to store results from differential expression analysis. It is produced in the same way as the latter, but has one more slot containing the W values. This is useful to do cluster analysis. Together the slots of the maigesDE class, it has one more given in *Slots* section.

Details

Objects of this class are produced in the same way as maigesDE. The addition of the slot W turn possible to do cluster analysis in this class of objects using the functions hierMde, somMde and kmeansMde, selecting genes according to the results of gene expression analysis.

Slots

W: numeric matrix containing the W values for the genes and samples used in the analysis.

Methods

boxplot signature(x = 'maigesDEcluster'): boxplot method for maigesDEcluster class. Display boxplots of one gene previously defined.

plot signature(x = 'maigesDEcluster'): plot method for maigesDEcluster class. Display *MA* plots.

print signature(x = 'maigesDEcluster'): print method for maigesDEcluster class.

show signature(x = 'maigesDEcluster'): show method for maigesDEcluster class.

summary signature(x = 'maigesDEcluster'): summary method for maigesDEcluster class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

deGenes2by2Ttest, deGenes2by2Wilcox, deGenes2by2BootT, deGenesANOVA, hierMde, somMde, kmeansMde.

maigesPreRaw-class *maigesPreRaw class, store pre raw microarray datasets*

Description

This class describes objects to handle data values and information about genes and samples used in the data. Here, it is possible to put any field from data tables that you want.

Details

This class of objects is intended to define an intermediate step between the external world (data tables, gene groups, gene networks, etc) and real raw objects given by the class maigesRaw. In this intermediate step the user can do any type of exploratory analysis and set bad spots (at slot BadSpots), or to do any type of calculation they judge important in the fields of Data slot.

This class of objects is created by a call from loadData. The functions addGeneGrps and addPaths may also be used to add information for gene groups and gene networks. After the exploratory analysis it must be converted in an object of class maigesRaw using the function createMaigesRaw.

If you have the package *Rgraphviz* installed and working it is possible to display the graphs stored in the slot Paths using the command plot.

Slots

Data: contains a list with matrices. Each matrix has nrow = number of spots and ncol = number of data tables (or chips) with the numerical values of each data field specified by the user in the configuration file used as parameter for loadData function.

GeneGrps: a list containing character vectors. These vectors specify genes from the gene groups that must be studied. Each vector is used as a gene group. The names of the groups are catch from the names of the vectors from the list, that is catch from file names.

Paths: list containing graphNEL objects specifying gene regulatory networks (or pathways). As in the previous slot, the names are catch from file names and stored as names of the elements from the list.

Layout: a list containing the number of rows (gridR) and columns (gridC) of grids, the number of rows (spotR) and columns (spotC) of spots inside each grid and the total number of spots.

Glabels: data frame giving the gene labels. These labels are generally used during the data analysis.

Slabels: data frame giving the sample labels. These labels are generally used during the data analysis.

BadSpots: logical vector specifying spots that was judged as bad ones. By default this slot is created as a vector of FALSEs with same length as number of spots.

Notes: char string that receives any comment about the dataset. The dataset description is stored in this slot.

Date: char string giving the date and hour that the object was created.

V.info: list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

print `signature(x = 'maigesPreRaw')`: print method for `maigesPreRaw` class.

show `signature(x = 'maigesPreRaw')`: show method for `maigesPreRaw` class.

summary `signature(x = 'maigesPreRaw')`: summary method for `maigesPreRaw` class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`loadData`, `addGeneGrps`, `addPaths`, `createMaigesRaw`.

---

maigesRaw-class *maigesRaw class, store raw microarray datasets*

Description

This class describes objects to handle intensity values and information about genes and samples used in the data. Objects of this class are obtained from class `maigesPreRaw` using function `createMaigesRaw`.

Details

This class of objects defines a real raw object that is used to do the normalisation. Objects of this class are generated from objects of class `maigesPreRaw` using the function `createMaigesRaw`. Here it is possible to do several plots for exploratory analysis using functions from *marray* package. Using the function `selSpots`, you select spots to use in the normalisation method, that is done by the functions `normLoc`, `normOLIN`, `normRepLoess`, `normScaleLimma` and `normScaleMarray`.

Slots

`Sf`: numeric matrix containing the test samples spots intensity values, spots corresponding to rows and data tables (or chips) corresponding to columns.

`Sb`: numeric matrix containing the test samples local background values, spots corresponding to rows and data tables (or chips) corresponding to columns.

`Sdye`: character vector with length equal the length of data files (chips), specifying the channel ('ch1' or 'ch2') used to label each test sample in each chip.

`Rb`: numeric matrix containing the reference samples spots intensity values, spots corresponding to rows and data tables (or chips) corresponding to columns.

`Rf`: numeric matrix containing the reference samples spots intensity values, spots corresponding to rows and data tables (or chips) corresponding to columns.

`Rdye`: character vector with length equal the length of data files (chips), specifying the channel ('ch1' or 'ch2') used to label each reference sample in each chip.

`Flag`: matrix specifying the flags for the spots, as in the matrices above, rows and columns correspond to genes and samples, respectively. Type o value may be numeric or character, depending of the flags used.

`BadSpots`: logical vector specifying spots that was judged as bad ones. By default this slot is created as a vector of FALSEs with same length as number of spots.

`UseSpots`: logical matrix indexing the spots to be used for normalisation.

`GeneGrps`: a logical matrix with rows representing the spots and columns representing different gene groups. Each column give the index of spots in that gene group.

`Paths`: list containing `graphNEL` objects specifying gene regulatory networks (or pathways). The first object in this list is a char string giving the gene label used to match the genes.

`Layout`: a list containing the number of rows (`gridR`) and columns (`gridC`) of grids, the number of rows (`spotR`) and columns (`spotC`) of spots inside each grid and the total number of spots.

`Glabels`: data frame giving the gene labels. These labels are generally used during the data analysis.

`Slabels`: data frame giving the sample labels. These labels are generally used during the data analysis.

`Notes`: char string that receives any comment about the dataset. The dataset description is stored in this slot.

`Date`: char string giving the date and hour that the object was created.

`V.info`: list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

[`signature(x = 'maigesRaw')`: subsetting operator for spots on the array or arrays in the batch, ensures that all slots are subset properly.

boxplot `signature(x = 'maigesRaw')`: boxplot method for `maigesRaw` class. Display boxplots of the slides and print tip groups using package *marray*.

dim `signature(x = 'maigesRaw', value = 'numeric')`: get the dimensions of the object, numeric vector of length two.

image `signature(x = 'maigesRaw')`: image method for `maigesRaw` class. Display colour representation of the slides using package *marray*.

plot `signature(x = 'maigesRaw')`: plot method for `maigesRaw` class. Display *MA* plots.

print `signature(x = 'maigesRaw')`: print method for `maigesRaw` class.

show `signature(x = 'maigesRaw')`: show method for `maigesRaw` class.

summary `signature(x = 'maigesRaw')`: summary method for `maigesRaw` class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`createMaigesRaw`, `selSpots`, `normLoc`, `normOLIN`, `normRepLoess`, `normScaleLimma` and `normScaleMarray`.

maigesRelNetB-class *maigesRelNetB class, store results of relevance network analysis (Butte's method)*

Description

This class defines a structure to store the results of relevance network analysis, using a method that was proposed by Butte et al. (2000), see the reference below. Objects of this type are generated calling the function relNetworkB.

Details

Objects of this class are produced by calling the function relNetworkB over an object of class maiges to search for pairs of genes with significant correlation values in a specific sample type, according to the method proposed by Butte et al. (2000).

Slots

W: numeric matrix giving the W values of the genes tested. This information is useful for doing plots representing the correlation coefficient values.

Corr: numerical square matrix that store the correlation coefficient values between all pair of genes tested.

Pval: numerical square matrix that store the p-value for each correlation value between the pairs of genes.

maxB: numerical square matrix that store the maximum (in absolute value) of the correlation values calculated by permutation values, as described by Butte et al. (2000).

type: character vector giving the name of sample type tested.

Slabel: string with the sample label used to define the sample types tested.

Date: character string giving the date and time that the object was generated.

V.info: list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

image signature(x = 'maigesRelNetB'): image method for maigesRelNetB class. Display colour representation of the correlation matrix.

plot signature(x = 'maigesRelNetB'): plot method for maigesRelNetB class. Display a circular graph representing the relevance network.

print signature(x = 'maigesRelNetB'): print method for maigesRelNetB class.

show signature(x = 'maigesRelNetB'): show method for maigesRelNetB class.

summary signature(x = 'maigesRelNetB'): summary method for maigesRelNetB class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Butte, A.J.; Tamayo, P.; Slonim, D.; Golub, T.R. and Kohane, I.S. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks. **PNAS**, 97, 12182-12186, 2000. (http://www.pnas.org/cgi/content/full/97/22/12182)

See Also

relNetworkB.

`maigesRelNetM-class`

maigesRelNetM class, store results of relevance network analysis (comparing two sample types)

Description

This class defines a structure to store the results of relevance network analysis, using a method that compare the correlation values between pairs of genes in two different sample types by a z-transformation. Objects of this type are generated calling the function relNetworkM.

Details

Objects of this class are produced by calling the functions relNetworkM over an object of class maiges to search for pairs of genes with altered correlation values between two sample types tested.

Slots

`W`: numeric matrix giving the W values of the genes tested. This information is useful for doing plots representing the correlation coefficient values.

`Corr1`: numerical square matrix that store the correlation coefficient values between all pair of genes tested for the first sample type.

`Corr2`: numerical matrix similar to the previous one, storing the correlation values for the second sample type.

`DifP`: numerical matrix that store the p-value for the comparison between the correlation values of both sample types tested.

`types`: character vector giving the names of sample types tested.

`Slabel`: string with the sample label used to define the two sample types tested.

`Date`: character string giving the date and time that the object was generated.

`V.info`: list containg three characters. The first one is a string containing the R version used when the object was created. The second is a char vector with base packages and the last one is another char vector with additional packages and version numbers.

Methods

image `signature(x = 'maigesRelNetM')`: image method for `maigesRelNetM` class. Display colour representation of the correlation and p-values matrices.

plot `signature(x = 'maigesRelNetM')`: plot method for `maigesRelNetM` class. Display circular graphs representing the relevance networks.

print `signature(x = 'maigesRelNetM')`: print method for `maigesRelNetM` class.

show `signature(x = 'maigesRelNetM')`: show method for `maigesRelNetM` class.

summary `signature(x = 'maigesRelNetM')`: summary method for `maigesRelNetM` class.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`relNetworkM`.

MI *Calculate Mutual Information*

Description

Function to calculate the mutual information of 2 random variables, or between all pairs of rows of a numerical matrix.

Usage

```
MI(x, y=NULL, k=1)
```

Arguments

`x`	numerical matrix to calculate the MI between all pairs of rows from `x`. Also, `x` must be a numerical vector and `y` must be specified as another numerical vector of same lenght as `x` and the MI value between both them are calculated.
`y`	optional numerical vector that must be specified if `x` is a vector. Defaults to NULL.
`k`	integer specifying the number of the neighbours to be used in the calculation of the MI value.

Details

This function implements an algorithm proposed by Kraskov et al. (2004) that don't use estimator of the entropy.

Value

If `x` is a matrix, the function return a square matrix with lenght equal to the number of rows of `x` with MI values between all pairs of rows from `x`. If `x` is a numerical vector, `y` must be specified and the function returns a positive real number with the MI value between the two vectors.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Kraskov, A.; Stogbauer, H. and Grassberger, P. Estimating mutual information, **Physical Review E**, 69, 066138, 2004 (http://scitation.aip.org/getabs/servlet/GetabsServlet?prog=normal&id=PLEEE8000069000006066138000001&idtype=cvips&gifs=yes).

Examples

```
x <- runif(50, 0, 1)
y <- rbeta(50, 1, 2)
MI(x, y)

z <- matrix(rnorm(100, 0, 1), 4, 25)
MI(z)
```

---

normLoc *Normalise a cDNA Microarray Object*

Description

This function loads a maigesRaw object and corrects for location bias.

Usage

```
normLoc(obj=NULL, ...)
```

Arguments

obj	object of type maigesRaw to be normalised.
...	additional parameters for function normalizeWithinArrays from *limma* package.

Details

This function for normalisation is entirely based on the function normalizeWithinArrays from *limma* package. See their help page to known how to setup the parameters correctly. The parameters layout and weights, are automatically specified by the object obj and must not be specified.

Value

This function returns a maiges object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

normalizeWithinArrays from *limma* package.

Examples

```
## Loading the dataset
data(gastro)

## Do the normalization by loess method and span 0.4
gastro.norm = normLoc(gastro.raw2, span=0.4, method="loess")

## Do the same normalization without background subtraction
gastro.norm = normLoc(gastro.raw2, span=0.4, method="loess", bc.method="none")
```

---

normOLIN *Normalise a cDNA Microarray Object*

Description

This function loads a maigesRaw object and normalise it using methods OLIN or OSLIM.

Usage

```
normOLIN(obj=NULL, ...)
```

Arguments

obj	object of type maigesRaw to be normalised.
...	additional parameters for function olin.

Details

This function for normalisation is entirely based on function olin from *OLIN* package. This function implements the methods OLIN and OSLIM, proposed by Futschich and Crompton (2004). See help page for this function to discover how to set the parameters.

Value

This function returns a maiges object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Futschik, M. and Crompton, T. Model selection and efficiency testing for normalization of cDNA microarray data, **Genome Biology**, 5, R60, 2004 (http://genomebiology.com/2004/5/8/R60).

See Also

olin from *OLIN* package.

Examples

```
## Loading the dataset
data(gastro)

## Doing the OLIN normalization. Pay attention because, this methods are
## very time consuming!
## Not run:
gastro.norm = normOLIN(gastro.raw2) ## Without scale adjustment

gastro.norm = normOLIN(gastro.raw2, OSLIN=TRUE) ## With scale adj

## End(Not run)
```

---

normRepLoess *Bootstrap of LOWESS normalisation*

Description

This function normalises a microarray object re-doing the LOWESS fitting several times, selecting a pre-specified proportion of points aleatorily.

Usage

```
normRepLoess(raw, span=0.4, propLoess=0.5, nRep=50, func="none",
             bkgSub="none", ...)
```

Arguments

raw	an object of class maigesRaw to be normalised.
span	real number in (0,1) representing the proportion of points to use in the loess regression.
propLoess	real number in (0,1) representing the proportion of points (spots) to be used in each iteration of loess.
nRep	number of repetitions for loess procedure.
func	character string giving the function to estimate the final W value. You must use 'mean', 'median' or 'none' (default).
bkgSub	character with background subtraction method, using the function backgroundcorrect from *limma* package.
...	additional parameters for function loessFit from *limma* package.

Details

The LOWESS fitting for normalising microarray data is a computational intensive task, so pay attention to not specify a very large argument in nRep. If you do so, your process will take so much time to conclude.

Value

The result of this function is an object of class maiges.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

loessFit.

Examples

```
## Loading the dataset
data(gastro)

## Doing the repetition loess with default parameters. Be carefull, this
## is very time consuming
## Not run:
gastro.norm = normRepLoess(gastro.raw2)

## End(Not run)

## Do the same normalization selecting 60% dos spots with 10
## repetitions and estimating the W by the mean value.
## Not run:
gastro.norm = normRepLoess(gastro.raw2, propLoess=0.6, nRep=10, func="mean")

## End(Not run)
```

---

normScaleLimma *Scale adjust a cDNA Microarray Object*

Description

This function loads a maigesRaw or maiges object and scale adjust (normalise between arrays) the data using functions from *limma* package.

Usage

```
normScaleLimma(obj=NULL, ...)
```

Arguments

obj	object of type maigesRaw or maiges to be normalised.
...	additional parameters for function normalizeBetweenArrays.

Details

This function for scale adjustment is entirely based on function normalizeBetweenArrays from *limma* package. See the help page for this function to see how to set the parameters. Pay attention to the 'vsn' method of scale adjustment, that must be used alone.

Value

This function returns a maiges object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

`normalizeBetweenArrays` from *limma* package.

Examples

```
## Loading the dataset
data(gastro)

## Doing the scale adjustment from median-absolute-value method (from
## limma)
gastro.norm = normScaleLimma(gastro.norm, method="scale")
boxplot(gastro.norm) ## To see the efect of adjustment

## To do VSN scale adjustment (from vsn package) use the command. Be
## carefull that this method adjust the variance along A values and not
## between chips!!
gastro.norm = normScaleLimma(gastro.raw2, method="vsn")
boxplot(gastro.norm) ## See the effect
```

`normScaleMarray` *Scale adjust a cDNA Microarray Object*

Description

This function loads a `maigesRaw` or `maiges` object and scale adjust (normalise between arrays) the data using functions from *marray* package.

Usage

```
normScaleMarray(obj=NULL, ...)
```

Arguments

`obj`	object of type `maigesRaw` or `maiges` to be normalised.
`...`	additional parameters for function `maNormScale`.

Details

This function for scale adjustment is entirely based on function `maNormScale` from *marray* package. See the help page for this function to see how to set the parameter. Pay attention to the `subset` argument that is fixed directly from the `UseSpots` and `BadSpots` from `obj` object, and must not be specified in the additional arguments.

The functionality of the scale adjustment function from *marray* package was added because it uses an estimator of MAD different from that one used in *limma* package. Also, using `maNormScale` function it is possible to do print tip scale adjustment.

Value

This function returns a maiges object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

maNormScale from *marray* package.

Examples

```
## Loading the dataset
data(gastro)

## Doing global MAD scale adjustment
gastro.norm = normScaleMarray(gastro.norm, norm="globalMAD")
boxplot(gastro.norm) ## To see the effect of MAD adjustment

## For print tip MAD use the following command
## Not run:
gastro.norm = normScaleMarray(gastro.norm, norm="printTipMAD")

## End(Not run)
```

---

plotGenePair *Scatter plots for pair of genes*

Description

This function displays scatter plots for pair of genes that presented altered correlation values in Relevance Network analysis.

Usage

```
plotGenePair(obj, gene1, gene2, posL=NULL, rCor=TRUE)
```

Arguments

obj	object of class maigesRelNetM.
gene1	character string giving the first gene identification.
gene2	character string giving the first gene identification.
posL	numerical vector of length 2, specifying the x and y position of the legend.
rCor	logical specifying if the correlation are robust (calculated by the function robustCorr. Defaults to TRUE.

Details

This function only picks the result of the relNetworkM and display scatter plots for a pair of genes giving the regression lines and the correlation values for the two biological groups tested.

Value

This function don't return any object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

maigesRelNetM, robustCorr, relNetworkM.

Examples

```
## Loading the dataset
data(gastro)

## Constructing the relevance network for sample
## 'Tissue' comparing 'Neso' and 'Aeso' for the 1st gene group
gastro.net = relNetworkM(gastro.summ, sLabelID="Tissue",
  samples = list(Neso="Neso", Aeso="Aeso"), geneGrp=11,
  type="Rpearson")

## As the sample size is small, because we used a small fraction of the
## genes from the original dataset, this isn't so reliable.
plotGenePair(gastro.net, "KLK13", "EVPL")
```

---

plot *Method plot for objects defined in this package*

Description

Generic function plot to display scatter plots or other types of graphical representation for objects defined in this package.

Usage

```
## S3 method for class 'maigesRaw':
plot(x, bkgSub="subtract", z=NULL, legend.func=NULL,
    ylab="W", ...)

## S3 method for class 'maiges':
plot(x, z=NULL, legend.func=NULL, ylab="W", ...)

## S3 method for class 'maigesANOVA':
plot(x, z=NULL, legend.func=NULL, ylab="W", ...)

## S3 method for class 'maigesDE':
plot(x, adjP="none", idx=1, ...)

## S3 method for class 'maigesDEcluster':
plot(x, adjP="none", idx=1, ...)
```

```
## S3 method for class 'maigesClass':
plot(x, idx=1, ...)

## S3 method for class 'maigesRelNetB':
plot(x=NULL, cutPval=0.05, cutCor=NULL,
name=NULL, ...)

## S3 method for class 'maigesRelNetM':
plot(x=NULL, cutPval=0.05, names=NULL, ...)

## S3 method for class 'maigesActMod':
plot(x, type=c("S", "C")[2], keepEmpty=FALSE, ...)

## S3 method for class 'maigesActNet':
plot(x, type=c("score", "p-value")[1], ...)
```

Arguments

x	an object of any class defined in this package, except maigesPreRaw.
bkgSub	string specifying the method for background subtraction. See function backgroundcorrect to find the available options.
z	accessor method for stratifying data, see maPlot.
legend.func	string specifying options to show legend in the figure.
ylab	character string specifying the label to y axis.
adjP	type of p-value adjustment, see function mt.rawp2adjp in package multtest.
idx	index of the test statistic to be plotted in case of objects of classes maigesDE and maigesDEcluster or the index of the clique to be plotted in case of object with class maigesClass.
cutPval	real number in [0,1] specifying a cutoff p-value to show significant results from relevance network analysis. For class maigesRelNetB, if this parameter is specified the argument cutCor isn't used.
cutCor	real number in [0,1], specifying a coefficient correlation value cutoff (in absolute value) to show only absolute correlation values greater than this value. Pay attention, to use this cutoff it is necessary to specify cutPval as NULL.
name	character string giving a name for sample type tested to be plotted as a name in the method for class maigesRelNetB.
names	similar to the previous one, but it is a vector of length 3.
type	string specifying the type of colour map to be plotted. For class maigesActMod it must be 'S' or 'C' for samples or biological conditions, respectively. For class maigesActNet it must be 'score' or 'p-value' for the statistics or p-values of the tests, respectively.
keepEmpty	logical, if true the results of all gene groups are displayed, else only the gene groups that present at least one significant result are displayed.
...	additional arguments for method maPlot or plot

Details

This method uses the function maPlot to display scatter plots ratio vs mean values for objects of class maiges, maigesRaw or maigesANOVA. For objects of class maigesDE or maigesDEcluster,

this method display volcano plots. For objects of class maigesClass it do 2 or 3 dimensions scatter plots that facilitate the visualisation of good classifying cliques of genes For objects of class maigesRelNetM the method displays 3 circular graphs representing the correlation values for the two groups tested and the p-values of the tests. For class maigesRelNetB it displays only one circular graph showing the correlation values for the type tested. In objects of class maigesActMod and maigesActNet the method do the same job as image.

Pay attention that, even using the method maPlot from *marray* package, we plot *W* values against *A* values instead of *MA* plots.

Author(s)

Gustavo H. Esteves <gesteves@vision.ime.usp.br>

See Also

mt.rawp2adjp, backgroundcorrect, maPlot in the package marray, plot in the base package.

Examples

```
## Loading the dataset
data(gastro)

## Example with an object of class maigesRaw, without and with backgound
## subtraction, also we present a plot with normexp (from limma package)
## subtract algorithm.
plot(gastro.raw[,1], bkgSub="none")
plot(gastro.raw[,1], bkgSub="subtract")
plot(gastro.raw[,1], bkgSub="normexp")

## Example with an object of class maigesNorm.
plot(gastro.norm[,1])



## Example for objects of class maigesDE.

## Doing bootstrap from t statistic test fot 'Type' sample label, k=1000
## specifies one thousand bootstraps
gastro.ttest = deGenes2by2Ttest(gastro.summ, sLabelID="Type")

plot(gastro.ttest) ## Volcano plot


## Example for object of class maigesClass.

## Doing LDA classifier with 3 genes for the 6th gene group comparing
## the 2 categories from 'Type' sample label.
gastro.class = classifyLDA(gastro.summ, sLabelID="Type",
  gNameID="GeneName", nGenes=3, geneGrp=6)

plot(gastro.class) ## plot the 1st classifier
plot(gastro.class, idx=7) ## plot the 7th classifier
```

```
## Example for object of class maigesActNet

## Doing functional classification of gene groups for 'Tissue' sample label
gastro.mod = activeMod(gastro.summ, sLabelID="Tissue", cutExp=1,
  cutPhiper=0.05)

plot(gastro.mod, "S", margins=c(15,3)) ## Plot for individual samples
plot(gastro.mod, "C", margins=c(21,5)) ## Plot for unique biological conditions



## Example for object of class maigesRelNetB

## Constructing the relevance network (Butte's method) for sample
## 'Tissue' equal to 'Neso' for the 1st gene group
gastro.net = relNetworkB(gastro.summ, sLabelID="Tissue",
  samples="Neso", geneGrp=1, type="Rpearson")

plot(gastro.net, cutPval=0.05)




## Example for object of class maigesRelNetM

## Constructing the relevance network for sample
## 'Tissue' comparing 'Neso' and 'Aeso' for the 1st gene group
gastro.net = relNetworkM(gastro.summ, sLabelID="Tissue",
  samples = list(Neso="Neso", Aeso="Aeso"), geneGrp=11,
  type="Rpearson")

plot(gastro.net, cutPval=0.05)
plot(gastro.net, cutPval=0.01)



## Example for objects of class maigesActNet

## Doing functional classification of gene networks for sample Label
## given by 'Tissue'
gastro.net = activeNet(gastro.summ, sLabelID="Tissue")

plot(gastro.net, type="score", margins=c(21,5))
plot(gastro.net, type="p-value", margins=c(21,5))
```

`print` *Method to print a nice visualisation of the objects defined in this package*

Description

Generic function `print` to display a nice (and simple) visualisation of the objects define in this package.

Usage

```
## S3 method for class 'maigesPreRaw':
print(x, ...)

## S3 method for class 'maigesRaw':
print(x, ...)

## S3 method for class 'maiges':
print(x, ...)

## S3 method for class 'maigesDE':
print(x, ...)

## S3 method for class 'maigesDEcluster':
print(x, ...)

## S3 method for class 'maigesClass':
print(x, ...)

## S3 method for class 'maigesRelNetB':
print(x, ...)

## S3 method for class 'maigesRelNetM':
print(x, ...)

## S3 method for class 'maigesActMod':
print(x, ...)

## S3 method for class 'maigesActNet':
print(x, ...)
```

Arguments

x	an object of any class defined in this package
...	further arguments passed to or from other methods

Author(s)

Gustavo H. Esteves <gesteves@vision.ime.usp.br>

See Also

print in the base package.

Examples

```
## Loading the dataset
data(gastro)

print(gastro)
print(gastro.raw)
print(gastro.norm)
print(gastro.summ)
```

relNet2TGF *Transform Relevance Network analysis in TGF output*

Description

This function receive an object of class maigesRelNetB or maigesRelNetM and write TGF files with the relevance networks produced.

Usage

```
relNet2TGF(...)

## S3 method for class 'maigesRelNetB':
relNet2TGF(data, dir = "./",
  filename="group.tgf", corC=NULL, pValue=0.05, ...)

## S3 method for class 'maigesRelNetM':
relNet2TGF(data, dir = "./",
  filenames=c("group1.tgf", "group2.tgf", "difPvalue.tgf"),
  pValue=0.05, ...)
```

Arguments

data	object of class maigesRelNetB or maigesRelNetM.
dir	character string specifying the folder to save the TGF files.
filename	character string specifying the file name, for objects of class maigesRelNetB.
filenames	character vector of length 3 with the file names to be saved, for objects of class maigesRelNetM.
corC	numeric in [0,1] specifying the cutoff for selecting absolute correlation. May also be 'max' to select the maximum correlation values in a permutation bootstrap strategy, as proposed by Butte et al. (2000).
pValue	numeric in [0,1] specifying the cutoff for selecting correlation values by p-values.
...	additional parameters.

Details

This function only picks the result of the relNetworkB or relNetworkM and display write TGF files. This files are interesting to be used with *Yed* graph visualisation and editing tool, wrote in Java (http://www.yworks.com/en/products_yed_about.htm).

Value

This function don't return any object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Butte, A.J.; Tamayo, P.; Slonim, D.; Golub, T.R. and Kohane, I.S. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks, **PNAS**, 97, 12182-12186, 2000 (http://www.pnas.org/cgi/content/full/97/22/12182)

See Also

relNetworkB, relNetworkM, maigesRelNetB, maigesRelNetM.

Examples

```
## Loading the dataset
data(gastro)

## Constructing the relevance network for sample
## 'Tissue' comparing 'Neso' and 'Aeso' for the 1st gene group
## The same is also true for objects of class maigesRelNetB
gastro.net = relNetworkM(gastro.summ, sLabelID="Tissue",
  samples = list(Neso="Neso", Aeso="Aeso"), geneGrp=11,
  type="Rpearson")

relNet2TGF(gastro.net)
```

---

relNetworkB *Relevance Network analysis*

Description

Function to construct Relevance Networks for one biological type (Butte's Relevance Network).

Usage

```
relNetworkB(data=NULL, gLabelID="GeneName", sLabelID="Classification",
            geneGrp=NULL, path=NULL, samples=NULL,
            type="Rpearson", bRep=1000, ...)
```

Arguments

data	object of class maiges.
gLabelID	character string giving the identification of gene label ID.
sLabelID	character string giving the identification of sample label ID.
geneGrp	character string (or numeric index) specifying the gene group to calculate the correlation values between them. If NULL (together with path) all genes are used.
path	character string (or numeric index) specifying the gene network to calculate the correlation values between them. If NULL (together with geneGrp) all genes are used.
samples	a character vector specifying the group to be compared.

type	type of correlation to be calculated. May be 'Rpearson' (default), 'pearson', 'kendall', 'spearman' or 'MI'.
bRep	integer specifying the number of bootstrap permutation to calculate the significance of correlation values.
...	additional parameters for functions robustCorr or cor.

Details

This method uses the function cor to calculate the usual correlation values, robustCorr to calculate a robust correlation using an idea similar to the leave-one-out or MI to calculate mutual information values.

Value

The result of this function is an object of class maigesRelNetB.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Butte, A.J. and Kohane, I.S. Unsupervised Knowledge discovery in medical databases using relevance networks. In Proc. AMIA Symp., 711-715, 1999 (http://www.amia.org/pubs/symposia/D005550.HTM)

Butte, A.J.; Tamayo, P.; Slonim, D.; Golub, T.R. and Kohane, I.S. Discovering functional relationships between RNA expression and chemotherapeutic susceptibility using relevance networks, **PNAS**, 97, 12182-12186, 2000 (http://www.pnas.org/cgi/content/full/97/22/12182)

Butte, A.J. and Kohane, I.S. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. In Pacific Symposium on Biocomputing, 5, 415-426, 2000 (http://psb.stanford.edu/psb-online/proceedings/psb00/)

See Also

cor, robustCorr, MI maigesRelNetB, plot.maigesRelNetB, image.maigesRelNetB.

Examples

```
## Loading the dataset
data(gastro)

## Constructing the relevance network (Butte's method) for sample
## 'Tissue' equal to 'Neso' for the 1st gene group
gastro.net = relNetworkB(gastro.summ, sLabelID="Tissue",
  samples="Neso", geneGrp=1, type="Rpearson")

## Constructing the relevance network (Butte's method) for sample
## 'Type' equal to 'Col' for the 1st gene group using the conventional
## pearson correlation
gastro.net = relNetworkB(gastro.summ, sLabelID="Type",
  samples="Col", geneGrp=1, type="pearson")
```

relNetworkM *Relevance Network analysis*

Description

Function to construct Relevance Networks comparing two distinct biological types.

Usage

```
relNetworkM(data=NULL, gLabelID="GeneName", sLabelID="Classification",
            geneGrp=NULL, path=NULL, samples=NULL,
            type="Rpearson", ...)
```

Arguments

data	object of class maiges.
gLabelID	character string giving the identification of gene label ID.
sLabelID	character string giving the identification of sample label ID.
geneGrp	character string (or numeric index) specifying the gene group to calculate the correlation values between them. If NULL (together with path) all genes are used.
path	character string (or numeric index) specifying the gene network to calculate the correlation values between them. If NULL (together with geneGrp) all genes are used.
samples	a named list with two character vectors specifying the two groups that must be compared.
type	type of correlation to be calculated. May be 'Rpearson' (default), 'pearson', 'kendall' or 'spearman'.
...	additional parameters for functions robustCorr or cor.

Details

This method uses the function cor to calculate

the usual correlation values or robustCorr to calculate

a robust correlation using an idea similar to the leave-one-out.

The correlation values are calculated for pairs of genes in the two groups specified by the argument samples, then a Fisher's Z transformation are done to calculate the significance for the difference between the two correlation values, this is implemented in the function compCorr. This method was first used in the work from Gomes et al. (2005).

Value

The result of this function is an object of class maigesRelNetM.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

References

Gomes, L.I.; Esteves, G.H.; Carvalho, A.F.; Cristo, E.B.; Hirata Jr., R.; Martins, W.K.; Marques, S.M.; Camargo, L.P.; Brentani, H.; Pelosof, A.; Zitron, C.; Sallum, R.A.; Montagnini, A.; Soares, F.A.; Neves, E.J. & Reis, L.F. Expression Profile of Malignant and Nonmalignant Lesions of Esophagus and Stomach: Differential Activity of Functional Modules Related to Inflammation and Lipid Metabolism, **Cancer Research**, 65, 7127-7136, 2005 (http://cancerres.aacrjournals.org/cgi/content/abstract/65/16/7127)

See Also

cor, robustCorr compCorr, maigesRelNetM, plot.maigesRelNetM, image.maigesRelNetM.

Examples

```
## Loading the dataset
data(gastro)

## Constructing the relevance network for sample
## 'Tissue' comparing 'Neso' and 'Aeso' for the 1st gene group
gastro.net = relNetworkM(gastro.summ, sLabelID="Tissue",
  samples = list(Neso="Neso", Aeso="Aeso"), geneGrp=11,
  type="Rpearson")
```

---

robustCorr *Calculate a robust correlation value*

Description

This function is intended to calculate robust correlation values between pairs of rows of numerical matrix or between two numerical vectors.

Usage

```
robustCorr(x, y=NULL)
```

Arguments

x	numerical matrix or vector. If a matrix the method calculates the robust correlations between all pairs of rows. If x is a vector, y must be specified as another vector of same length as x and the robust correlation between them is calculate.
y	optional numeric vector, must be specified if x is a vector.

Details

This function calculates a robust correlation value in a procedure similar to the leave-one-out used for cross-validation of classification results. The algorithm removes one point at a time and calculates a usual Pearson correlation value. Them, with a vector r of correlation values that has the same length as the columns of x (or vectors x and y), the algorithm decides by the min(**r**) or max(**r**), according with that one that is more distant from the median value.

Value

If x is a matrix, the method return a list with two square matrices, the first one containing the robust correlation values between all pairs of rows from x and the second containing the index of the point removed from calculation. If x is a vector, y must be specified and the function return a list with the robust correlation value between them and the index of the point removed.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

cor from package *stats*.

Examples

```
x <- runif(50, 0, 1)
y <- rbeta(50, 1, 2)
robustCorr(x, y)

z <- matrix(rnorm(100, 0, 1), 4, 25)
robustCorr(z)
```

---

selSpots *Select spots to use in normalisation*

Description

Function to select spots to be used in microarray normalisation.

Usage

```
selSpots(obj=NULL, sigNoise=1, rmFlag=NULL, gLabelsID=c("Name"),
         remove=list(c("BLANK","DAP","LYS","PHE","Q_GENE","THR","TRP")),
         badSpots=NULL, badLabel=NULL)
```

Arguments

obj	object of class maigesRaw.
sigNoise	positive real number indicating the cutoff to remove spots with signal-to-noise ratio below it.
rmFlag	vector of flag symbols to be removed (for normalisation). These flags are stored in the slot Flags from maigesRaw class.
gLabelsID	character vector indicating the gene labels to be searched to exclude that ones specified in remove argument.
remove	list of same length as GlabelsID containing character vector indicating the symbols of spots to be removed, according to the GlabelsID argument.
badSpots	index of bad spots (numeric or logical) identifying bad spots. May be the gene labels, with label ID specified by the argument badLabel.
badLabel	character string specifying the gene label ID for remove badSpots.

Details

This function takes the object of class maigesRaw and actualise the slot UseSpots according with the arguments passed to the function. This slot is read by the normalisation functions normLoc, normOLIN, normRepLoess, normScaleLimma and normScaleMarray to use only the spots that passed the criteria specified here.

Value

This function returns another object of class maigesRaw with the UseSpots slot actualised.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

Examples

```
## Loading the dataset
data(gastro)

## Filtering all spots with signal2noise ratio (Sf/Sb or Rf/Rb) greater
## or equal to 1 and that have 'Name' label as 'BLANK', 'DAP', ..., 'TRP'.
gastro.raw2 = selSpots(gastro.raw, sigNoise=1, rmFlag=NULL, gLabelsID="Name",
  remove=list(c("BLANK","DAP","LYS","PHE","Q_GENE","THR","TRP")))

## To see the number of spots that suvived the filtering above do
apply(gastro.raw2@UseSpots, 2, sum)

## To do the same filtering as above, also filtering flags marcke as 1
## and 4 do
gastro.raw2 = selSpots(gastro.raw, sigNoise=1, rmFlag=c(1,4), gLabelsID="Name",
  remove=list(c("BLANK","DAP","LYS","PHE","Q_GENE","THR","TRP")))

apply(gastro.raw2@UseSpots, 2, sum)
```

---

show-method *Show a nice visualisation of the objects defined in this package*

Description

S4 method to show the objects defined in this package without list all the slots of the object.

Usage

```
## S4 method for signature 'maigesPreRaw':
show(object)

## S4 method for signature 'maigesRaw':
show(object)

## S4 method for signature 'maiges':
show(object)
```

```
## S4 method for signature 'maigesANOVA':
show(object)

## S4 method for signature 'maigesDE':
show(object)

## S4 method for signature 'maigesDEcluster':
show(object)

## S4 method for signature 'maigesClass':
show(object)

## S4 method for signature 'maigesRelNetM':
show(object)

## S4 method for signature 'maigesRelNetB':
show(object)

## S4 method for signature 'maigesActMod':
show(object)

## S4 method for signature 'maigesActNet':
show(object)
```

Arguments

object	object of any class defined in this package, to be printed in a nice way

Note

It is possible to use Package *convert* to convert objects between classes defined in packages *limma*, *marray* and *Biobase*.

See Also

show in the *methods* package.

Examples

```
## Loading the dataset
data(gastro)

gastro
gastro.raw
gastro.norm
gastro.summ
```

---

somMde *Function to do SOM cluster analysis*

Description

This is a function to do SOM (Self Organising Maps) clustering analysis for objects of classes maigesDEcluster.

Usage

```
somMde(data, group=c("C", "R")[1], distance="correlation",
       method="complete", sampleT=NULL, doHier=FALSE, sLabelID="SAMPLE",
       gLabelID="GeneName", idxTest=1, adjP="none", nDEgenes=0.05, ...)
```

Arguments

data	object of class maigesDEcluster.
group	character string giving the type of grouping: by rows 'R' or columns 'C' (default).
distance	char string giving the type of distance to use. Only two options are available here: 'euclidean' and 'correlation' (default).
method	char string specifying the linkage method for the hierarchical cluster. Possible values are 'ward', 'single', 'complete' (default), 'average', 'mcquitty', 'median' or 'centroid'
sampleT	list with 2 vectors. The first one specify the first letter of different sample types to be coloured by distinct colours, that are given in the second vector. If NULL (default) no colour is used.
doHier	logical indicating if you want to do the hierarchical branch in the opposite dimension of clustering. Defaults to FALSE.
sLabelID	character string specifying the sample label ID to be used to label the samples.
gLabelID	character string specifying the gene label ID to be used to label the genes.
idxTest	numerical index of the test to be used to sort the genes when clustering objects of class maigesDEcluster.
adjP	string specifying the method of p-value adjustment. May be 'none', 'Bonferroni', 'Holm', 'Hochberg', 'SidakSS', 'SidakSD', 'BH', 'BY'.
nDEgenes	number of DE genes to be selected. If a real number in (0,1) all genes with p.value <= nDEgenes will be used. If an integer, the nDEgenes genes with smaller p-values will be used.
...	additional parameters for som function.

Details

This function implements the SOM clustering method for objects resulted from differential expression analysis. The method uses the function som from package *som*. For the adjustment of p-values in the selection of genes differentially expressed, we use the function mt.rawp2adjp from package *multtest*.

Value

This function display the heatmaps and return invisibly an object of class som resulted from the function som.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

som from package *som*. kmeansM and hierM for displaying k-means and hierarchical clusters, respectively.

Examples

```
## Loading the dataset
data(gastro)

## Doing bootstrap from t statistic test fot 'Type' sample label, k=1000
## specifies one thousand bootstraps
gastro.ttest = deGenes2by2Ttest(gastro.summ, sLabelID="Type")

## SOM cluster with 2 groups adjusting p-values by FDR, and showing all genes
## with p-value < 0.05
somMde(gastro.ttest, sLabelID="Type", adjP="BH", nDEgenes=0.05,
       xdim=2, ydim=1, topol="rect")

## SOM cluster with 4 groups adjusting p-values by FDR, and showing all genes
## with p-value < 0.05
somMde(gastro.ttest, sLabelID="Type", adjP="BH", nDEgenes=0.05,
       xdim=2, ydim=2, topol="rect")
```

---

somM *Function to do SOM cluster analysis*

Description

This is a function to do SOM (Self Organising Maps) clustering analysis for objects of classes maiges, maigesRaw and maigesANOVA. Use the function somMde for objects of class maigesDEcluster.

Usage

```
somM(data, group=c("C", "R")[1], distance="correlation",
     method="complete", sampleT=NULL, doHier=FALSE, sLabelID="SAMPLE",
     gLabelID="GeneName", rmGenes=NULL, rmSamples=NULL, rmBad=TRUE,
     geneGrp=NULL, path=NULL, ...)
```

Arguments

data	object of class maigesRaw, maiges, or maigesANOVA.
group	character string giving the type of grouping: by rows 'R' or columns 'C' (default).
distance	char string giving the type of distance to use. Only two options are available here: 'euclidean' and 'correlation' (default).
method	char string specifying the linkage method for the hierarchical cluster. Possible values are 'ward', 'single', 'complete' (default), 'average', 'mcquitty', 'median' or 'centroid'

sampleT	list with 2 vectors. The first one specify the first letter of different sample types to be coloured by distinct colours, that are given in the second vector. If NULL (default) no colour is used.
doHier	logical indicating if you want to do the hierarchical branch in the opposite dimension of clustering. Defaults to FALSE.
sLabelID	character string specifying the sample label ID to be used to label the samples.
gLabelID	character string specifying the gene label ID to be used to label the genes.
rmGenes	char list specifying genes to be removed.
rmSamples	char list specifying samples to be removed.
rmBad	logical indicating to remove or not bad spots (slot BadSpots in objects of class maiges, maigesRaw or maigesANOVA).
geneGrp	numerical or character specifying the gene group to be clustered. This is given by the columns of the slot GeneGrps in objects of classes maiges, maigesRaw and maigesANOVA.
path	numerical or character specifying the gene network to be clustered. This is given by the items of the slot Paths in objects of classes maiges, maigesRaw and maigesANOVA.
...	additional parameters for som function.

Details

This function implements the SOM clustering method for objects of microarray data defined in this package. The method uses the function som from package *som*.

Value

This function display the heatmaps and return invisibly an object of class som resulted from the function som.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

som from package *som*. kmeansM and hierM for displaying k-means and hierarchical clusters, respectively.

Examples

```
## Loading the dataset
data(gastro)

## Doing a SOM cluster with 2 groups using all genes, for maigesRaw class
somM(gastro.raw, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
     sLabelID="Sample", gLabelID="Name", xdim=2, ydim=1, topol="rect")

## Doing a SOM cluster with 3 groups using all genes, for maigesNorm class
somM(gastro.norm, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
     sLabelID="Sample", gLabelID="Name", xdim=3, ydim=1, topol="rect")

## Another example with 4 groups
```

```
somM(gastro.norm, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
     sLabelID="Sample", gLabelID="Name", xdim=2, ydim=2, topol="rect")

## If you want to use euclidean distance to group genes (or spots), with
## 3 groups
somM(gastro.summ, rmGenes=c("BLANK","DAP","LYS","PHE", "Q_GENE","THR","TRP"),
     sLabelID="Sample", gLabelID="Name", group="R",
     distance="euclidean", xdim=3, ydim=1, topol="rect")
```

`summarizeReplicates` *Summarise microarray objects*

Description

Function to summarise measures of a `maiges` class object, both by samples and genes.

Usage

```
summarizeReplicates(object=NULL, gLabelID="GeneName", sLabelID="Sample",
                    funcS="mean", funcG="mean", rmBad=TRUE, keepEmpty=TRUE)
```

Arguments

`object`	object of class `maiges`.
`gLabelID`	character string giving the gene label ID to be used to summarise the data by rows.
`sLabelID`	character string giving the sample label ID to be used to summarise the data by columns.
`funcS`	character string specifying the function to be applied for sample replicates. Defaults to 'mean'. If NULL, no resume is done for samples.
`funcG`	character string specifying the function to be applied for genes (spots) replicates. Defaults to 'mean'. If NULL, no resume is done for genes..
`rmBad`	logical indicating if you want to remove or not bad spots, given by the slot `BadSpots` in argument `object`.
`keepEmpty`	logical indicating if you want to maintain spots with empty information.

Details

This function takes the object of class `maiges` and resume the data by spots (rows) and samples (columns).

Value

This function returns another object of class `maiges` with replicates summarised to only one observation.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

Examples

```
## Loading the dataset
data(gastro)

## Summarising the data (maigesNorm class), replicated samples will be
## summarised by mean and genes by median
gastro.summ = summarizeReplicates(gastro.norm, gLabelID="GeneName",
  sLabelID="Sample", funcS="mean", funcG="median",
  keepEmpty=FALSE, rmBad=FALSE)

## To summarise genes by mean and keep the blank spots use
gastro.summ = summarizeReplicates(gastro.norm, gLabelID="GeneName",
  sLabelID="Sample", funcS="mean", funcG="mean",
  keepEmpty=TRUE, rmBad=FALSE)
```

---

summary *Method summary for the object from this package*

Description

Generic function summary to display a nice summary of the contents from classes of objects define in this package.

Usage

```
## S3 method for class 'maigesPreRaw':
summary(object, ...)

## S3 method for class 'maigesRaw':
summary(object, ...)

## S3 method for class 'maiges':
summary(object, ...)

## S3 method for class 'maigesDE':
summary(object, ...)

## S3 method for class 'maigesDEcluster':
summary(object, ...)

## S3 method for class 'maigesClass':
summary(object, ...)

## S3 method for class 'maigesRelNetB':
summary(object, ...)

## S3 method for class 'maigesRelNetM':
summary(object, ...)

## S3 method for class 'maigesActMod':
summary(object, ...)
```

```
## S3 method for class 'maigesActNet':
summary(object, ...)
```

Arguments

object	an object of any class defined in this package
...	further arguments passed to or from other methods

Author(s)

Gustavo H. Esteves <gesteves@vision.ime.usp.br>

See Also

summary in the base package.

Examples

```
## Loading the dataset
data(gastro)

summary(gastro)
summary(gastro.raw)
summary(gastro.norm)
summary(gastro.summ)
```

---

tableClass *Save HTML or CSV tables of good classifiers (cliques)*

Description

This function takes an object of class maigesClass resulting from classification analysis and write tables (in HTML or CSV format) a table of the cliques, or classifiers.

Usage

```
tableClass(classComp=NULL, file="./class_result",
           type=c("HTML","CSV")[1], nCliques=NULL)
```

Arguments

classComp	object of class maigesClass.
file	character string giving the file name to be saved (without extension).
type	character string with the type of file to be saved. Must be 'HTML' (default) or 'CSV'.
nCliques	integer specifying the number of cliques to be saved. If NULL (default) all cliques in the object are saved.

Value

This function don't return any object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

maigesClass, classifyLDA, classifySVM, classifyKNN, classifyLDAsc, classifySVMsc and classifyKNNsc.

Examples

```
## Loading the dataset
data(gastro)


## Doing LDA classifier with 3 genes for the 6th gene group comparing
## the 2 categories from 'Type' sample label.
gastro.class = classifyLDA(gastro.summ, sLabelID="Type",
  gNameID="GeneName", nGenes=3, geneGrp=6)

tableClass(gastro.class)
```

---

tablesDE *Save HTML or CSV tables of differentially expressed genes*

Description

This function takes an object of class maigesDE or maigesDEcluster generated by functions deGenes2by2Ttest, deGenes2by2Wilcox, deGenes2by2BootT and deGenesANOVA and save HTML or CSV tables differentially expressed genes.

Usage

```
tablesDE(deComp=NULL, dir="./", filenames=NULL, dataID="Someone's",
         type=c("HTML","CSV")[1], geneID="GeneName", hsID="ClusterId",
         gbID="GeneId", annotID="Annot", genes=NULL, logFold=TRUE,
         adjP="none", sort="p.value", nDEgenes=NULL)
```

Arguments

deComp	object of class maigesDE.
dir	character specifying the directory to save the files.
filenames	character vector with file names to be saved, if NULL default names are given.
dataID	character giving an identifier for the dataset. If NULL the notes slot from DE.comp are used.
type	type of file to be saved. May be 'HTML' (default) or 'CSV'.
geneID	character giving the ID of label for gene symbol in the dataset.
hsID	character giving the ID of label for unigene code in the dataset.
gbID	character giving the ID of label for Genbank entry in the dataset.
annotID	character giving the ID of label for gene annotation in the dataset.

genes	character vector specifying the genes to be saved, mapped according to geneID label.
logFold	logical specifying if the fold change must be saved in log2 scale.
adjP	character string giving the type of p-value adjustment. May be 'Bonferroni', 'Holm', 'Hochberg', 'SidakSS', 'SidakSD', 'BH', 'BY' or 'none'. Defaults to 'none'.
sort	character specifying the field to be sorted, may be 'p.value' (default), 'fold' or 'statistic'.
nDEgenes	number of differentially genes to be saved in the file. Defaults to NULL to save all genes. If an integer value the function saves the nDEgenes with smaller p-values (most significantly DE genes). If a number in (0,1) nDEgenes is used as a cutoff for the p-values.

Details

We use the function mt.rawp2adjp from package *multtest* to adjust p-values, any information the methods implemented must be searched in their help pages. The arguments hsID, gbID and annotID are used only to improve the tables generate including links for the respective databases, but if these information are absent in the dataset they must be specified as NULL. The argument geneID must be necessarily specified, because the genes must be at least one identification.

Value

This function don't return any object.

Author(s)

Gustavo H. Esteves <<gesteves@vision.ime.usp.br>>

See Also

deGenes2by2Ttest, deGenes2by2Wilcox, deGenes2by2BootT, deGenesANOVA, maigesDE, maigesDEcluster.

Examples

```
## Loading the dataset
data(gastro)

## Doing bootstrap from t statistic test fot 'Type' sample label, k=1000
## specifies one thousand bootstraps
gastro.ttest = deGenes2by2Ttest(gastro.summ, sLabelID="Type")

tablesDE(gastro.ttest) ## Save HTML tables

## To save only tables with p-value < 0.05
tablesDE(gastro.ttest, nDEgenes=0.05)

## To save only tables with 30 most significantly genes
tablesDE(gastro.ttest, nDEgenes=30)
```

Index