Package ‘a4Core’

April 11, 2018

Type Package

Title Automated Affymetrix Array Analysis Core Package

Version 1.26.0

Date 2011-06-04

Author Willem Talloen, Tobias Verbeke

Maintainer Tobias Verbeke <`tobias.verbeke@openanalytics.eu`>, Willem Ligtenberg <`willem.ligtenberg@openanalytics.eu`>

Description Automated Affymetrix Array Analysis Core Package

Depends methods, Biobase, glmnet

License GPL-3

biocViews Microarray

NeedsCompilation no

R topics documented:

`confusionMatrix` *Generic function to produce a confusion matrix (related to a classification problem)*

Description

Generic function to produce a confusion matrix (related to a classification problem)

Usage

```
confusionMatrix(x, ...)
```

Arguments

`x`	object (usually a model fit object) that contains all information needed to produce the confusion matrix.
`...`	further arguments for a specific method

Author(s)

Tobias Verbeke

`simulateData` *Simulate Data for Package Testing and Demonstration Purposes*

Description

Simulate Data for Package Testing and Demonstration Purposes

Usage

```
simulateData(nCols = 40, nRows = 1000, nEffectRows = 5, nNoEffectCols = 5,
    betweenClassDifference = 1, withinClassSd = 0.5)
```

Arguments

`nCols`	number of samples; currently this should be an even number
`nRows`	number of features (genes)
`nEffectRows`	number of differentially expressed features
`nNoEffectCols`	number of samples for which the profile of a differentially expressed feature will be set similar to the other class
`betweenClassDifference`	Average mean difference between the two classes to simulate a certain signal in the features for which an effect was introduced; the default is set to 1
`withinClassSd`	Within class standard deviation used to add a certain noise level to the features for which an effect was introduced; the default standard deviation is set to 0.5

Value

object of class ExpressionSet with the characteristics specified

Note

The simulation assumes the variances are equal between the two classes. Heterogeneity could easily be introduced in the simulation if this would be requested by the users.

Author(s)

W. Talloen and T. Verbeke

Examples

```
someEset <- simulateData(nCols = 40, nRows = 1000, nEffectRows = 5, nNoEffectCols = 5)
someEset
```

`topTable` *S4 Generic for obtaining a top table*

Description

a top table is a rectangular object (e.g. data frame) which lists the top n most relevant variables

Usage

```
topTable(fit, n, ...)
```

Arguments

`fit`	object for which to obtain a top table, generally a fit object for a given model class
`n`	number of features (variables) to list in the top table, ranked by importance
`...`	further arguments for specific methods

Author(s)

Tobias Verbeke

`topTable-methods` *Methods for topTable*

Description

Methods for topTable. topTable extracts the top n most important features for a given classification or regression procedure

Arguments

`fit`	object resulting from a classification or regression procedure
`n`	number of features that one wants to extract from a table that ranks all features according to their importance in the classification or regression model; defaults to 10 for limma objects

Methods

glmnet and lognet

glmnet objects are produced by `lassoClass` (a4Classif) or `lassoReg` (a4Base)

fit = "glmnet", n = "numeric"fit = "lognet", n = "numeric" lognet objects are produced by `lassoClass` (a4Classif) or `lassoReg` (a4Base)

fit = "elnet", n = "numeric" lognet objects are produced by `lassoClass` (a4Classif) or `lassoReg` (a4Base)

Index