Package 'covRNA'

April 14, 2017

Type Package

Title Multivariate Analysis of Transcriptomic Data

Version 1.0.0

Author Lara Urban <lara.h.urban@ebi.ac.uk>

Maintainer Lara Urban <lara.h.urban@ebi.ac.uk>

Description This package provides the analysis methods fourthcorner and RLQ analysis for large-scale transcriptomic data.

License GPL (>= 2)

LazyData TRUE

Depends ade4, Biobase

Imports parallel, genefilter, grDevices, stats, graphics

biocViews GeneExpression, Transcription

Suggests BiocStyle, knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

R topics documented:

`covRNA-package` *The covRNA package*

Description

covRNA (covariate analysis of RNA-Seq data) is a fast and user-friendly R package which implements fourthcorner analysis and RLQ of transcriptomic data.

Gene expression data normally comes with covariates of the samples and of the genes. To analyze associations between sample and gene covariates, the fourthcorner analysis tests the statistical significance of the associations by permutation tests while the RLQ visualizes associations within and be-tween the covariates.

The fourthcorner analysis and RLQ implemented in the ade4 package are adapted to easily analyze large-scale transcriptomic data. (1) Runtime and storage space are significantly reduced, (2) the analysis accounts for tran-scriptome-specific shapes of the empirical permutation distributions, (3) the analysis is rendered user-friendly by supplying automation, simple design-ing of plots and unsupervised gene filtering.

To cite covRNA, please use citation("covRNA"). For further details, please refer to the vignette by openVignette("covRNA") and the man pages.

Details

Package:	covRNA
Type:	Package
License:	GPL (>=2)
LazyLoad:	yes

Author(s)

Lara Urban

Maintainer: Lara Urban <lara.h.urban@ebi.ac.uk>

References

To be announced soon.

`Baca dataset` *The Baca dataset*

Description

The integrated Baca dataset contains the ExpressionSet Baca; its assayData contains deep sequenced RNA-Seq data of Bacillus anthracis under four stress conditions (with four replicates per stress conditions). The raw sequence reads derive from Passalacqua et al. (2012) and are availaible at Gene Expression Omnibus (GEO, accession number GSE36506). We have already mapped,

counted and DESeq2 normalised these counts. The phenoData assigns the stress condition, i.e. ctrl, cold, salt and alcohol stress, to the samples. The featureData contains COG annotations of the genes.

Usage

```
Baca
```

Format

ExpressionSet

Value

ExpressionSet

Source

GEO GSE36506

References

Passalacqua, K. D., Varadarajan, A., Weist, C., Ondov, B. D., Byrd, B. et al. (2012) *Strand-Specific RNA-Seq Reveals Ordered Patterns of Sense and Antisense Transcription in Bacillus anthracis*. PLoS ONE, 7(8):e43350.

Examples

```
data(Baca)
fData(Baca)
pData(Baca)
exprs(Baca)
```

`ord` *RLQ for transcriptomic data*

Description

The RLQ visualises the association between and within sample and gene covariates by ordination. It applies generalized singular value decomposition (GSVD) to the fourthcorner matrix, which contains the associations between the sample and gene covariates. This is realised by eigendecomposition of the covariance matrices of the fourthcorner matrix. The name RLQ refers to the three dataframes R, L and Q to be analyzed. The function ‘ord‘ automates the ‘rlq‘ function of the ‘ade4‘ package.

The input has to be given as dataframe or matrix. Dataframe/matrix L [n x p] contains transcriptomic data of p samples across n genes, dataframe/matrix R [n x m] contains m gene covariates across the n genes and dataframe/matrix Q [p x s] contains s sample covariates across the p samples. Alternatively, objects of the class ExpressionSet (with assayData, phenoData and featureData) can be used as input. If the argument ExprSet is missing, the function will use the dataframes/matrices R, L and Q as input.

Genes can be filtered with respect to their expression variance before analysis (argument exprvar); the function will automatically discard the gene covariates which do not annotate any of the remaining genes.

Warning: If R and Q are given as matrices, they will be converted to dataframes at the beginning of the function.

Warning: If R or Q is missing, it will be replaced by an identity matrix. Then, a principal component analysis of this matrix will be performed what might be time-consuming, depending on the size of the identity matrix.

Usage

```
ord(ExprSet, R=NULL, L=NULL, Q=NULL, exprvar=1, nf=2)
```

Arguments

ExprSet	An ExpressionSet of the *Biobase* package. The ExpressionSet is used as default input. If no ExpressionSet is given, the individual dataframes/matrices R, L and Q can be used as input.
R	A dataframe/matrix containing information about each gene. The number of rows in R must match the number of rows in L. If R is missing, it will be replaced by an identity matrix [n x n].
L	A dataframe/matrix of gene expression values of genes across samples.
Q	A dataframe/matrix containing information about each sample. The number of rows in Q must match the number of columns in L. If Q is missing, it will be replaced by an identity matrix [p x p].
exprvar	The fraction of most variably expressed genes to take into account. If the functions 'stat' and 'ord' shall be combined, this value has to be the same in both analyses.
nf	The number of axes to be considered by ordination.

Details

The function automates the following steps. Firstly, Correspondence Analysis is applied to gene expression table L. Either Principal Component Analysis (only quantitative variables), Multiple Correspondence Analysis (only categorical variables) or Hillsmith analysis (quantitative and categorical variables) are applied to the covariate tables R and Q. Secondly, RLQ is applied to the results of these ordination methods.

Value

The function returns a list ob class ord where:

call	gives the original call of the function.
rank	gives the rank.
nf	gives number of axes to be considered by ordination.
RV	gives the RV coefficient.
eig	gives a vector of the eigenvalues.
variance	gives the variance explained by the axes.
lw	gives the row weights of the fourthcorner table.

cw	gives the column weights of the fourthcorner table.
lw	gives the row weights of the fourthcorner table.
tab	gives the fourthcorner table.
li	gives the coordinates of the covariates of R.
l1	gives the normed scores of the covariates of R.
co	gives the coordinates of the covariates of Q.
c1	gives the normed scores of the covariates of Q.
lR	gives the row coordinates of R.
mR	gives the normed row scores of R.
lQ	gives the row coordinates of Q.
mQ	gives the normed row scores of Q.
aR	gives projection of axis onto co-inertia axis of R.
aR	gives projection of axis onto co-inertia axis of Q.
ngenes	gives the number of analysed genes.

Author(s)

Lara Urban

Examples

```
data(Baca)
ordBaca <- ord(ExprSet = Baca, exprvar = 1, nf = 2)
ls(ordBaca)
plot(ordBaca)
```

---

plot.ord *Plot RLQ for transcriptomic data*

Description

The function plot can visualise different features of an ord object by adjusting the argument "feature". By default, a barplot of the variance explained by the axes of the RLQ is plotted (see arguments).

Usage

```
## S3 method for class 'ord'
plot(x, feature="variance", xaxis=1, yaxis=2, cex=1, range=2, ...)
```

Arguments

x	An object of class ord that shall be visualised by ordination.
feature	Defines which features of the object shall be visualised: "columns L","rows L", "columns R" and "columns Q" visualise the respective variables as oridnation, "variance" shows a barplot of the variance explained by the axes, "correlation circle R" and "correlation circle Q" visualise the projection of the original space into the ordination space.
xaxis, yaxis	Define which axes of ordination shall be shown by x- and y-axis, respectively.
cex	Defines size of covariate text.
range	The range of the axes can be extended or reduced, e.g. for the case that not all covariates are visible in the default setting.
...	More plotting parameters can be added.

Value

Plot of RLQ.

Author(s)

Lara Urban

Examples

```
ordBaca <- ord(Baca)
plot(ordBaca)
```

---

plot.stat *Plot the fourthcorner analysis for transcriptomic data*

Description

The function plot produces a cross table of the gene and sample covariates of a stat object. Colours indicate positive/negative significance or absence of significance of the assciations (per default: white for non-significant, red for negative significant and red for positive significant associations).

Usage

```
## S3 method for class 'stat'
plot(x, col=c("lightgrey","deepskyblue","red"), sig=TRUE,
                    alpha=0.05, show=c("adj","non-adj"), cex=1,
                    ynames, xnames, ytext=1, xtext=1, shiftx=0, shifty=0, ...)
```

Arguments

x	An object of class stat that shall be visualised as a cross table.
col	A vector of three colours. The first colour represents non-significant, the second positive significant, the third negative significant associations in the cross table.
sig	If TRUE (default), only covariates involved in at least one significant association are plotted.

alpha	The significance level.
show	'adj' or 'non-adj' indicate if adjusted or raw p-values shall be plotted, respectively.
cex	The magnitude of the text in the cross table.
ynames, xnames	Row and column names of the cross table. By default, the column names of R and Q are used, respectively.
ytext, xtext	Rotation of the row and column names of the cross table.
shifty, shiftx	Shift of the row and column names to the right or to the left.
...	More plotting parameters can be added.

Value

Plot of fourthcorner analysis.

Author(s)

Lara Urban

Examples

```
statBaca <- stat(Baca, nrcor = 2)
plot(statBaca)
```

---

stat *Fourthcorner analysis for transcriptomic data*

Description

The fourthcorner analysis tests for significant associations between each sample covariate and each gene covariate by statistical permutation tests. The sample and gene covariates can be categorical and/or quantitative.

The input has to be given as dataframe or matrix. Dataframe/matrix L [n x p] contains transcriptomic data of p samples across n genes, dataframe/matrix R [n x m] contains m gene covariates across the n genes and dataframe/matrix Q [p x s] contains s sample covariates across the p samples. Alternatively, objects of the class ExpressionSet (with assayData, phenoData and featureData) can be used as input. If the argument ExprSet is missing, the function will use the dataframes/matrices R, L and Q as input.

The number of permutations is set to 9999 per default to assure significance of p-values after multiple testing correction. As computation time increases with size of the matrices/dataframes and with number of permutations, parallelization across multiple cores is highly recommended. Per default, all except one CPU cores on the current host are used.

Genes can be filtered with respect to their expression variance before analysis (argument exprvar); the function will automatically discard the gene covariates which do not annotate any of the remaining genes.

Warning: If R and Q are given as matrices, they will be converted to dataframes at the beginning of the function.

Warning: If R or Q is missing, it will be replaced by an identity matrix.

Usage

```
stat(ExprSet, R=NULL, L=NULL, Q=NULL, npermut=9999, padjust="BH",
         nrcor=detectCores()-1, exprvar=1)
```

Arguments

ExprSet	An ExpressionSet of the *Biobase* package. The ExpressionSet is used as default input. If no ExpressionSet is given, the individual dataframes/matrices R, L and Q can be used as input.
R	A dataframe/matrix containing information about each gene. The number of rows in R must match the number of rows in L. If R is missing, it will be replaced by an identity matrix [n x n].
L	A dataframe/matrix of gene expression values of genes across samples.
Q	A dataframe/matrix containing information about each sample. The number of rows in Q must match the number of columns in L. If Q is missing, it will be replaced by an identity matrix [p x p].
npermut	The number of permutations.
padjust	The method of multiple testing adjustment of the pvalues, see p.adjust.methods for all methods implemented in R.
nrcor	The number of cores to be used.
exprvar	The fraction of most variably expressed genes to take into account. If the functions 'stat' and 'ord' shall be combined, this value has to be the same in both analyses.

Details

Dependent on the covariate combination, a statistic is calculated based on matrix multiplication of the three tables. This statistic amounts to a correlation coefficient for the association between quantitative-quantitative and quantitative-categorical variables and to a Chi2-related statistic for the association between categorical-categorical variables.

Value

The function returns a list of class stat where:

stat	is a cross table (m x s) with the values of the original statistical tests per covariate combination.
pvalue, adj.pvalue	are cross tables (m x s) which contain the p-values and adjusted p-values, respectively, of the permutation tests per covariate combination.
adjust.method	shows the applied multiple testing adjustment method.
npermut	gives the number of permutations per permutation test.
ngenes	gives the number of analysed genes ("all" in the case of no filtering of the genes).
call	gives the original call of the function.

Author(s)

Lara Urban

Examples

```
data(Baca)
statBaca <- stat(ExprSet = Baca, npermut = 999, padjust = "BH", nrcor = 2, exprvar = 1)
statBaca$adj.pvalue
plot(statBaca)
```

`vis` *Simultaneous visualisation of transcriptomic data by combining fourthcorner analysis and RLQ*

Description

The vis function simultaneously visualizes the results of the functions stat and ord. Firstly, all covariates of R and Q are visualized by ordination in one plot; covariates involved in at least one significant association are shown in black, other covariates are shown in gray. Then, all covariates that are significantly associated according to stat are connected by lines which color represents the character of their significance.

Usage

```
vis(Stat, Ord=NULL, alpha=0.05, xaxis=1, yaxis=2, col=c("gray", transblue, transred),
    alphatrans=0.5, cex=1, rangex=2, rangey=2, ...)
```

Arguments

`Stat`	An object of class stat.
`Ord`	An object of class ord. The objects stat and ord should have the same value ngenes.
`alpha`	The significance level.
`xaxis, yaxis`	Define which axes of ordination shall be shown by x- and y-axis, respectively.
`col`	A vector of three colors. The first color represents non-significant variables, the second positive significant, the third negative significant associations.
`alphatrans`	Defines degree of transparency of the second and third color.
`cex`	The magnitude of the text in the ordination.
`rangex, rangey`	The range of the x axis and y axis can be extended or reduced, e.g. for the case that not all covariates are visible in the default setting.
`...`	More plotting parameters can be added.

Value

Plot of fourthcorner analysis and RLQ.

Author(s)

Lara Urban

Examples

```
data(Baca)
statBaca <- stat(Baca, nrcor = 2)
ordBaca <- ord(Baca)
vis(Stat = statBaca, Ord = ordBaca)
vis(Ord = ordBaca)
```

Index