

Package ‘OUTRIDER’

April 16, 2019

Title OUTRIDER - OUTlier in RNA-Seq fInDER

Type Package

Version 1.0.4

Date 2019-03-18

Maintainer Christian Mertes <mertes@in.tum.de>

URL <https://github.com/gagneurlab/OUTRIDER>

BugRepos <https://github.com/gagneurlab/OUTRIDER/issues>

Author Felix Brechtmann [aut, cre], Christian Mertes [aut, cre],
Ange Matusевичiute [aut, cre], Vicente Yezpez [aut], Julien Gagneur [aut]

Description Identification of aberrant gene expression in RNA-seq data.
Read count expectations are modeled by an autoencoder to control for confounders in the data. Given these expectations, the RNA-seq read counts are assumed to follow a negative binomial distribution with a gene-specific dispersion. Outliers are then identified as read counts that significantly deviate from this distribution. Furthermore, OUTRIDER provides useful plotting functions to analyze and visualize the results.

VignetteBuilder knitr

biocViews ImmunoOncology, RNASeq, Transcriptomics, Alignment,
Sequencing, GeneExpression, Genetics

License MIT + file LICENSE

NeedsCompilation yes

Encoding UTF-8

RoxygenNote 6.1.1

Depends R (>= 3.5), BiocParallel, GenomicFeatures,
SummarizedExperiment, data.table, methods

Imports BBmisc, Biobase, BiocGenerics, compiler, DESeq2 (>= 1.16.1),
GenomicRanges, ggplot2, ggpubr, gplots, graphics, IRanges,
matrixStats, plotly, plyr, pcaMethods, PRROC, RColorBrewer,
Rcpp, reticulate, S4Vectors, scales, splines, stats, utils

Suggests testthat, knitr, rmarkdown, BiocStyle,
TxDb.Hsapiens.UCSC.hg19.knownGene, org.Hs.eg.db, RMySQL,
AnnotationDbi, beeswarm, covr

LinkingTo Rcpp, RcppArmadillo

Collate package-OUTRIDER.R class-OutriderDataSet.R
 inputCheckerFunctions.R helperFunctions.R getNSetterFuns.R
 getNSetterFunsInternal.R autoencoder.R fitNB.R ZscoreMatrix.R
 method-evaluation.R method-counts.R method-gridSearch.R
 method-plot.R method-results.R pValMatrix.R filterExpression.R
 OUTRIDER.R sizeFactor.R RcppExports.R updateE.R updateD.R
 updateTheta.R PCAcorrections.R thetaMethodOfMoments.R
 controlForConfounders.R lossNGradientRVersion.R

git_url <https://git.bioconductor.org/packages/OUTRIDER>

git_branch RELEASE_3_8

git_last_commit 18fcd8e

git_last_commit_date 2019-03-18

Date/Publication 2019-04-15

R topics documented:

aberrant	2
computeGeneLength	3
computeLatentSpace	4
computePvalues	5
computeZscores	6
controlForConfounders	7
counts	7
estimateBestQ	8
filterExpression	9
findEncodingDim	10
fit	11
fpkm	12
getBestQ	12
makeExampleOutriderDataSet	13
normalizationFactors	14
OUTRIDER	15
OutriderDataSet-class	16
plotVolcano	17
results	20
sampleExclusionMask	21
sizeFactors	22
Index	24

aberrant

Number of aberrant events

Description

Identifies the aberrant events and returns the number of aberrant counts per gene or sample or returns a matrix indicating aberrant events.

Usage

```
aberrant(ods, padjCutoff = 0.05, zScoreCutoff = 0, by = c("none",
  "sample", "gene"))
```

Arguments

ods	An OutriderDataSet object
padjCutoff	The padjust cutoff
zScoreCutoff	The absolute Z-score cutoff, if NA or NULL no Z-score cutoff is used
by	if the results should be summarized by 'sample', 'gene' or not at all (default).

Value

The number of aberrant events by gene or sample or a TRUE/FALSE matrix of the size sample x gene of aberrant events.

Examples

```
ods <- makeExampleOutriderDataSet()
ods <- OUTRIDER(ods, implementation='pca')

aberrant(ods)[1:10,1:10]
tail(sort(aberrant(ods, by="sample")))
tail(sort(aberrant(ods, by="gene")))
```

computeGeneLength	<i>Extracting the gene length from annotations</i>
-------------------	--

Description

Computes the length for each gene based on the given GTF file or annotation. Here the length of a gene is defined by the total number of bases covered by exons.

Usage

```
computeGeneLength(ods, gtfFile, format = "gtf", mapping = NULL, ...)
```

Arguments

ods	An OutriderDataSet for which the gene length should be computed.
gtfFile	Can be a GTF file or an txDb object with annotation.
format	The format parameter from makeTxDbFromGFF
mapping	If set, it is used to map gene names between the GFT and the ods object. This should be a 2 column data.frame: 1. column GTF names and 2. column ods names.
...	further arguments to makeTxDbFromGFF

Value

An `OutriderDataSet` containing a `basepairs` column with the calculated gene length. Accessable through `mcols(ods)['basepairs']`

Examples

```
ods <- makeExampleOutriderDataSet(dataset="GTExSkinSmall")
annotationFile <- system.file("extdata", "gencode.v19.genes.small.gtf.gz",
  package="OUTRIDER")
ods <- computeGeneLength(ods, annotationFile)

mcols(ods)['basepairs']
fpkm(ods)[1:10,1:10]
```

computeLatentSpace *Extracting the latent space*

Description

Extracts the latent space from the `OutriderDataSet` object determined by the autoencoder.

Usage

```
computeLatentSpace(ods)
```

Arguments

ods An `OutriderDataSet`

Value

A matrix containing the latent space determined by the autoencoder.

Examples

```
ods <- makeExampleOutriderDataSet()

ods <- estimateSizeFactors(ods)
ods <- controlForConfounders(ods, implementation="pca")
computeLatentSpace(ods)[,1:6]
```

computePvalues	<i>Calculate P-values</i>
----------------	---------------------------

Description

This function computes the P-values based on the fitted negative binomial model. It computes two matrices with the same dimension as the count matrix (samples x genes), which contain the corresponding P-values and adjusted P-values of every count.

Usage

```
computePvalues(object, ...)  
  
## S4 method for signature 'OutriderDataSet'  
computePvalues(object,  
  alternative = c("two.sided", "greater", "less"), method = "BY",  
  BPPARAM = bpparam())
```

Arguments

object	An OutriderDataSet
...	additional params, currently not used.
alternative	Can be one of "two.sided", "greater" or "less" to specify the alternative hypothesis used to calculate the P-values, defaults to "two.sided"
method	Method used for multiple testing
BPPARAM	Can be used to parallelize the computation, defaults to bpparam()

Value

An OutriderDataSet object with computed nominal and adjusted P-values

See Also

p.adjust

Examples

```
ods <- makeExampleOutriderDataSet()  
  
ods <- estimateSizeFactors(ods)  
ods <- fit(ods)  
  
ods <- computePvalues(ods)  
  
assays(ods)[['pValue']][1:10, 1:10]
```

computeZscores	<i>Z score computation</i>
----------------	----------------------------

Description

Computes the z scores for every count in the matrix. The z score is defined in the log2 space as follows:

$$z_{ij} = \frac{l_{ij} - \mu_j^l}{\sigma_j^l}$$

, where l is the log2 transformed normalized count and μ and σ the mean and standard deviation for gene j , respectively.

Usage

```
computeZscores(ods, ...)

## S4 method for signature 'OutriderDataSet'
computeZscores(ods, peerResiduals = FALSE,
  ...)
```

Arguments

ods	OutriderDataSet
...	Further arguments passed on to ZscoreMatrix.
peerResiduals	If TRUE, PEER residuals are used to compute Z scores

Value

An OutriderDataSet containing the Z score matrix "zScore" and the log2 fold changes "l2fc" as assays.

Examples

```
ods <- makeExampleOutriderDataSet()
ods <- estimateSizeFactors(ods)

ods <- controlForConfounders(ods, implementation="pca")
ods <- computeZscores(ods)

zScore(ods)[1:10,1:10]
assay(ods, "l2fc")[1:10,1:10]
```

controlForConfounders *Autoencoder function to correct for confounders.*

Description

This is the wrapper function for the autoencoder implementation. It can be used to call the standard R implementation or the experimental Python implementation.

Usage

```
controlForConfounders(ods, q, implementation = c("autoencoder", "pca"),
  BPPARAM = bpparam(), ...)
```

Arguments

ods	An <code>OutriderDataSet</code> object
q	The encoding dimensions
implementation	"autoencoder", the default, will use the autoencoder implementation. Also 'pca' and 'peer' can be used to control for confounding effects
BPPARAM	A BiocParallelParam instance to be used for parallel computing.
...	Further arguments passed on to the specific implementation method.

Value

An ods object including the control factors

Examples

```
ods <- makeExampleOutriderDataSet()
implementation <- 'autoencoder'

ods <- estimateSizeFactors(ods)
ods <- controlForConfounders(ods, implementation=implementation)

plotCountCorHeatmap(ods, normalized=FALSE)
plotCountCorHeatmap(ods, normalized=TRUE)
```

counts *Accessors for the 'counts' slot of an OutriderDataSet object.*

Description

The counts slot holds the count data as a matrix of non-negative integer count values, one row for each observational unit (eg.: gene), and one column for each sample.

Usage

```
## S4 method for signature 'OutriderDataSet'
counts(object, normalized = FALSE,
        minE = 0.5)

## S4 replacement method for signature 'OutriderDataSet,matrix'
counts(object) <- value
```

Arguments

object	OutriderDataSet
normalized	TRUE/FALSE whether counts should be normalized
minE	minimal expected count.
value	An integer matrix containing the counts

Details

By default this function returns the raw counts. If control factors are computed or provided the normalized counts can be returned using `normalized = TRUE`. The offset parameter can be used to add a pseudocount to the count before dividing by the normalization. This can be useful when the `log(counts)` are computed and in case the control values are in the same order of magnitude as the counts.

Value

A matrix containing the counts

See Also

[sizeFactors](#), [normalizationFactors](#)

Examples

```
ods <- makeExampleOutriderDataSet()
counts(ods)[1:10,1:10]

ods <- estimateSizeFactors(ods)
counts(ods, normalized=TRUE)[1:10,1:10]
```

estimateBestQ

Estimation of Q

Description

Estimating the best q for the given data set

Usage

```
estimateBestQ(ods)
```


Arguments

ods An OutriderDataSet object

Value

The estimated dimension of hidden confounders

Examples

```
ods <- makeExampleOutriderDataSet()
estimateBestQ(ods)
```

filterExpression	<i>Filter expression</i>
------------------	--------------------------

Description

To filter out non expressed genes this method uses the FPKM values to get a comparable value over genes. To calculate the FPKM values the user needs to provide a GTF file or the basepair parameter as described in [fpkm](#).

Usage

```
filterExpression(x, ...)

## S4 method for signature 'OutriderDataSet'
filterExpression(x, gtfFile, fpkmCutoff = 1,
  filterGenes = TRUE, savefpkm = FALSE, minCounts = FALSE, ...)
```

Arguments

x An OutriderDataSet object

... Additional arguments passed to computeGeneLength

gtfFile A txDb object or a GTF/GFF file to be used as annotation

fpkmCutoff The threshold for filtering based on the FPKM value

filterGenes If TRUE, the default, the object is subsetted.

savefpkm If TRUE the FPKM values are saved as assay

minCounts Filter for required read counts per gene

Value

An OutriderDataSet containing the passedFilter column, which indicates if the given gene passed the filtering threshold. If filterGenes is TRUE the object is already subsetted.

Examples

```
ods <- makeExampleOutriderDataSet(dataset="GTEXSkinSmall")
annotationFile <- system.file("extdata",
  "gencode.v19.genes.small.gtf.gz", package="OUTRIDER")
ods <- filterExpression(ods, annotationFile, filterGenes=FALSE)

mcols(ods)['passedFilter']
fpkm(ods)[1:10,1:10]
dim(ods)

ods <- ods[mcols(ods)[['passedFilter']]
dim(ods)
```

findEncodingDim	<i>Find the optimal encoding dimension</i>
-----------------	--

Description

Finds the optimal encoding dimension for a given data set by running a grid search based on the provided parameter set.

Usage

```
findEncodingDim(ods, params = seq(2, min(100, ncol(ods) - 1, nrow(ods) -
  1), 2), freq = 0.01, zScore = 3, sdlog = log(1.6), lnorm = TRUE,
  inj = "both", ..., BPPARAM = bpparam())

findInjectZscore(ods, freq = 0.01, zScoreParams = c(seq(1.5, 4, 0.5),
  "lnorm"), encDimParams = c(seq(3, 40, 3), seq(45, 70, 5), 100, 130,
  160), inj = "both", ..., BPPARAM = bpparam())
```

Arguments

ods	An OutriderDataSet
params, encDimParams	Set of possible q values.
freq	Frequency of outlier, defaults to 1E-2
zScore, zScoreParams	Set of possible injection Z-score, defaults to 3.
sdlog	Standard deviation of the sitribution on the log scale.
lnorm	If TRUE, the default, Z-scores are drawn from a log normal distribution with a mean of log(zScore) in log-scale.
inj	Injection strategy, by default 'both'.
...	Further arguments passed on to the controlForConfounders function.
BPPARAM	BPPARAM object by default bpparam().

Value

The optimal encoding dimension

Examples

```
ods <- makeExampleOutriderDataSet()
encDimSearchParams <- c(5, 8, 10, 12, 15)
zScoreParams <- c(2, 3, 5, 'lnorm')
implementation <- 'autoencoder'
register(MulticoreParam(4))

ods1 <- findEncodingDim(ods, params=encDimSearchParams,
  implementation=implementation)
plotEncDimSearch(ods1)

ods2 <- findInjectZscore(ods, zScoreParams=zScoreParams,
  encDimParams=encDimSearchParams, implementation=implementation)
plotEncDimSearch(ods2)
```

fit

*Fit the negative binomial distribution***Description**

Fit a negative binomial (NB) distribution to the counts per gene over all samples using, if available, the precomputed control factors. If no normalization factors are provided only the sizeFactors are used.

Usage

```
fit(object, ...)

## S4 method for signature 'OutriderDataSet'
fit(object, BPPARAM = bpparam())
```

Arguments

object	An OutriderDataSet
...	additional arguments, currently not used.
BPPARAM	by default bpparam()

Value

An OutriderDataSet object with the fitted model. Accessible through: `mcols(ods)[,c('mu', 'theta')]`.

Examples

```
ods <- makeExampleOutriderDataSet()
ods <- estimateSizeFactors(ods)
ods <- fit(ods)

mcols(ods)[1:10,c('mu', 'theta')]
```

 fpkm

Calculate FPM and FPKM values

Description

This is the fpm and fpkm function from DESeq2. For more details see: [fpkm](#) and [fpm](#)

See Also

[fpkm](#) [fpm](#)

Examples

```
ods <- makeExampleOutriderDataSet()
mcols(ods)['basepairs'] <- round(rnorm(nrow(ods), 1000, 500))

mcols(ods)['basepairs']
fpkm(ods)[1:10,1:10]
fpm(ods)[1:10,1:10]
```

 getBestQ

Getter/Setter functions

Description

This is a collection of small accessor/setter functions for easy access to the values within the OUT-RIDER model.

Usage

```
getBestQ(ods)

zScore(ods)

pValue(ods)

padj(ods)

## S4 method for signature 'OutriderDataSet'
dispersions(object, ...)

theta(ods)
```

Arguments

ods, object An OutriderDataSet object.
 ... Further arguments currently not in use.

Value

A matrix or vector dependent on the type of data retrieved.

Examples

```
ods <- makeExampleOutriderDataSet(10, 10)
ods <- OTRIDER(ods)

zScore(ods)
pValue(ods)
padj(ods)
theta(ods)
theta(ods) == 1/dispersions(ods)
getBestQ(ods)
```

```
makeExampleOutriderDataSet
```

Create example data sets for OTRIDER

Description

Creates an example data set from a file or simulates a data set based on random counts following a negative binomial distribution with injected outliers with a fixed z score away from the mean of the gene.

Usage

```
makeExampleOutriderDataSet(n = 200, m = 80, q = 10, freq = 0.001,
  zScore = 6, inj = c("both", "low", "high"), sf = rnorm(m, mean = 1,
  sd = 0.1), dataset = c("none", "GTEXSkinSmall", "KremerNBaderSmall"))
```

Arguments

n	Number of simulated genes
m	Number of simulated samples
q	number of simulated latent variables.
freq	Frequency of in-silico outliers
zScore	Absolute z score of in-silico outliers (default 6).
inj	Determines whether counts are injected with the strategy ('both', 'low', 'high'), default is 'both'.
sf	Artificial Size Factors
dataset	If "none", the default, an example data set is simulated. One can also use example data set included in the package by specifying 'GTEXSkinSmall' or 'KremerNBaderSmall'

Value

An OutriderDataSet containing an example dataset. Depending on the parameters it is based on a real data set or it is simulated

Examples

```
# A generic dataset
ods1 <- makeExampleOutriderDataSet()
ods1

# A generic dataset with specified sample size and injection method
ods2 <- makeExampleOutriderDataSet(n=200, m=50, inj='low')
ods2

# A subset of a real world dataset from GTEx
ods3 <- makeExampleOutriderDataSet(dataset="GTExSkinSmall")
ods3
```

normalizationFactors *Accessor functions for the normalization factors in an Outrider-DataSet object.*

Description

To normalize raw count data normalization factors can be provided as a matrix. When running [controlForConfounders](#) the normalization factors are stored within the OutriderDataset object. This normalization factors are then used to compute the normalized counts.

Usage

```
## S4 method for signature 'OutriderDataSet'
normalizationFactors(object)

## S4 replacement method for signature 'OutriderDataSet,matrix'
normalizationFactors(object) <- value

## S4 replacement method for signature 'OutriderDataSet,DataFrame'
normalizationFactors(object) <- value

## S4 replacement method for signature 'OutriderDataSet,data.frame'
normalizationFactors(object) <- value

## S4 replacement method for signature 'OutriderDataSet,`NULL`'
normalizationFactors(object) <- value
```

Arguments

object	An OutriderDataSet object.
value	The matrix of normalization factors
...	additional arguments

Value

A numeric matrix containing the normalization factors or the `OutriderDataSet` object with an updated `normalizationFactors` assay.

See Also

`DESeq2::normalizationFactors`
[sizeFactors](#)

Examples

```
ods <- makeExampleOutriderDataSet()

normFactors <- matrix(runif(nrow(ods)*ncol(ods),0.5,1.5),
  ncol=ncol(ods),nrow=nrow(ods))

# the normalization factors matrix should not have 0's in it
# it should have geometric mean near 1 for each row
normFactorsRM <- normFactors / exp(rowMeans(log(normFactors)))
normalizationFactors(ods) <- normFactorsRM
normalizationFactors(ods)[1:10,1:10]

normalizationFactors(ods) <- NULL
ods <- estimateSizeFactors(ods)
normalizationFactors(ods) <- normFactors
all(normalizationFactors(ods) == t(sizeFactors(ods) * t(normFactors)))
```

 OUTRIDER

OUTRIDER - Finding expression outlier events

Description

The `OUTRIDER` function runs the default `OUTRIDER` pipeline combining the fit, the computation of Z scores and P-values. All computed values are returned as an `OutriderDataSet` object.

To have more control over each analysis step, one can call each function separately.

1. [estimateSizeFactors](#) to calculate the `sizeFactors`
2. [controlForConfounders](#) to control for confounding effects
3. [fit](#) to fit the negative binomial model (only needed if the autoencoder is not used)
4. [computePvalues](#) to calculate the nominal and adjusted P-values
5. [computeZscores](#) to calculate the Z scores

Usage

```
OUTRIDER(ods, q, controlData = TRUE, implementation = "autoencoder",
  BPPARAM = bpparam(), ...)
```

Arguments

ods	An OutriderDataSet object
q	The encoding dimensions
controlData	If TRUE, the default, the raw counts are controlled for confounders by the autoencoder
implementation	"autoencoder", the default, will use the autoencoder implementation. Also 'pca' and 'peer' can be used to control for confounding effects
BPPARAM	A BiocParallelParam instance to be used for parallel computing.
...	Further arguments passed on to <code>controlForConfounders</code>

Value

OutriderDataSet with all the computed values. The values are stored as assays and can be accessed by: `assay(ods, 'value')`. To get a full list of calculated values run: `assayNames(ods)`

Examples

```
ods <- makeExampleOutriderDataSet()
implementation <- 'autoencoder'

ods <- OUTRIDER(ods, implementation=implementation)

pValue(ods)[1:10,1:10]
res <- results(ods, all=TRUE)
res

plotAberrantPerSample(ods)
plotVolcano(ods, 1)
```

OutriderDataSet-class *OutriderDataSet class and constructors*

Description

The OutriderDataSet class is designed to store the whole OUTRIDER data set needed for an analysis. It is a subclass of RangedSummarizedExperiment. All calculated values and results are stored as assays or as annotation in the mcols structure provided by the RangedSummarizedExperiment class.

Usage

```
OutriderDataSet(se, countData, colData, ...)
```

Arguments

se	A RangedSummarizedExperiment object or any object which inherits from it and contains a count matrix as the first element in the assay list.
countData	A simple count matrix. If dim names are provided, they have to be unique. This is only used if no se object is provided.

colData Additional to the count data a DataFrame can be provided to annotate the samples.

... Further arguments can be passed to [DESeqDataSet](#), which is used to parse the user input and create the initial RangedSummarizedExperiment object.

Value

An OutriderDataSet object.

Author(s)

Christian Mertes <mertes@in.tum.de>, Felix Brechtmann <brechtma@in.tum.de>

Examples

```
ods <- makeExampleOutriderDataSet()
ods

ods <- makeExampleOutriderDataSet(dataset="Kremer")
ods
```

plotVolcano

Visualization functions for OTRIDER

Description

The OTRIDER package provides multiple functions to visualize the data and the results of a full data set analysis.

This is the list of all plotting function provided by OTRIDER:

- `plotAberrantPerSample()`
- `plotVolcano()`
- `plotExpressionRank()`
- `plotQQ()`
- `plotCountCorHeatmap()`
- `plotFPKM()`
- `plotDispEsts()`
- `plotPowerAnalysis()`
- `plotEncDimSearch()`

For a detailed description of each plot function please see the details. Most of the functions share the same parameters.

Usage

```
plotVolcano(ods, sampleID, main, padjCutoff = 0.05, zScoreCutoff = 0,
  pch = 16, basePlot = FALSE, col = c("gray", "firebrick"))
```

```
plotQQ(ods, geneID, main, global = FALSE, padjCutoff = 0.05,
  zScoreCutoff = 0, samplePoints = TRUE, legendPos = "topleft",
  outlierRatio = 0.001, conf.alpha = 0.05, pch = 16, xlim = NULL,
  ylim = NULL, col = NULL)
```

```
plotExpressionRank(ods, geneID, main, padjCutoff = 0.05,
  zScoreCutoff = 0, normalized = TRUE, basePlot = FALSE,
  log = TRUE, col = c("gray", "firebrick"), ...)
```

```
plotCountCorHeatmap(ods, normalized = TRUE, rowCentered = TRUE,
  rowCoFactor = NULL, rowColSet = "Set1", colCoFactor = NULL,
  colColSet = "Set2", nCluster = 4,
  main = "Count correlation heatmap", dendrogram = "both",
  basePlot = TRUE, names = c("both", "row", "col", "none"), ...)
```

```
plotAberrantPerSample(ods, main, padjCutoff = 0.05, zScoreCutoff = 0,
  outlierRatio = 0.001, col = brewer.pal(3, "Dark2")[c(1, 2)],
  yadjust = c(1.2, 1.2), labLine = c(3.5, 3),
  ylab = "#Aberrantly expressed genes", labCex = par()$cex,
  ymax = NULL, ...)
```

```
plotFPKM(ods)
```

```
## S4 method for signature 'OutriderDataSet'
plotDispEsts(object, compareDisp, xlim, ylim,
  main = "Dispersion estimates versus mean expression", ...)
```

```
plotPowerAnalysis(ods)
```

```
plotEncDimSearch(ods)
```

Arguments

ods, object	An OutriderDataSet object.
sampleID, geneID	A sample or gene ID, which should be plotted. Can also be a vector. Integers are treated as indices.
main	Title for the plot, if missing a default title will be used.
padjCutoff, zScoreCutoff	Significance or Z-score cutoff to mark outliers
pch	Integer or character to be used for plotting the points
basePlot	if TRUE, use the R base plot version, else use the plotly framework, which is the default
col	Set color for the points. If set, it must be a character vector of length 2. (1. normal point; 2. outlier point)
global	Flag to plot a global Q-Q plot, default FALSE

samplePoints	Sample points for Q-Q plot, defaults to max 30k points
legendPos	Set legendpos, by default topleft.
outlierRatio	The fraction to be used for the outlier sample filtering
conf.alpha	If set, a confidence interval is plotted, defaults to 0.05
xlim, ylim	The x/y limits for the plot or NULL to use the full data range
normalized	If TRUE, the normalized counts are used, the default, otherwise the raw counts
log	If TRUE, the default, counts are plotted in log10.
...	Additional parameters passed to plot() or plot_ly() if not stated otherwise in the details for each plot function
rowCentered	If TRUE, the counts are row-wise (gene-wise) centered
rowCoFactor	A vector of co-factors for color coding the rows
rowColSet	A vector of colors or a color set from RColorBrewer
colCoFactor	A vector of co-factors for color coding the columns
colColSet	A vector of colors or a color set from RColorBrewer
nCluster	An integer to be used for cutting the dendrogram into groups. If this argument is set the resulting clusters are saved in the returned OutriderDataSet.
dendrogram	A character string indicating whether to draw 'none', 'row', 'column' or 'both' dendrograms.
names	character string indicating whether to draw 'none', 'row', 'col', or 'both' names.
yadjust	Option to adjust position of Median and 90 percentile labels.
labLine	Option to move axis labels
ylab	The y axis label
labCex	The label cex parameter
ymax	If set, ymax is the upper bound for the plot range on the y axis.
compareDisp	If TRUE, the default, and if the autoCorrect normalization was used it computes the dispersion without autoCorrect and plots it for comparison.

Details

plotAberrantPerSample: The number of aberrant events per sample are plotted sorted by rank. The ... parameters are passed on to the [aberrant](#) function.

plotVolcano: the volcano plot is sample-centric. It plots for a given sample the negative log10 nominal P-values against the Z-scores for all genes.

plotExpressionRank: This function plots for a given gene the expression level against the expression rank for all samples. This can be used with normalized and unnormalized expression values.

plotQQ: the quantile-quantile plot for a given gene or if global is set to TRUE over the full data set. Here the observed P-values are plotted against the expected ones in the negative log10 space.

plotCountCorHeatmap: The correlation heatmap of the count data of the full data set. Default the values are log transformed and row centered. This function returns an OutriderDataSet with annotated clusters if requested. The ... arguments are passed to the [heatmap.2](#) function.

plotFPKM: The distribution of FPKM values. If the OutriderDataSet object contains the passedFilter column, it will plot both FPKM distributions for the expressed genes and for the filtered genes.

plotDispEsts: Plots the dispersion of the OutriderDataSet model against the normalized mean count. If autoCorrect is used it will also estimate the dispersion without normalization for comparison.

`plotPowerAnalysis`: The power analysis plot should give the user a ruff estimate of the events one can be detected with OTRIDER. Based on the dispersion of the provided OTRIDER data set the theoretical P-value over the mean expression is plotted. This is done for different expression levels. The curves are smooths to make the reading of the plot easier.

Value

If base R graphics are used nothing is returned else the plotly or the gplot object is returned.

Examples

```
ods <- makeExampleOutriderDataSet(dataset="Kremer")
implementation <- 'autoencoder'

ods <- filterExpression(ods, minCounts=TRUE)
ods <- OTRIDER(ods, implementation=implementation)

plotAberrantPerSample(ods)

plotVolcano(ods, 1)
plotVolcano(ods, 'MUC1404', basePlot=TRUE)

plotExpressionRank(ods, 1)
plotExpressionRank(ods, "FAAH", normalized=FALSE,
  log=FALSE, main="Over expression outlier", basePlot=TRUE)

plotQQ(ods, 1)
plotQQ(ods, global=TRUE, outlierRatio=0.001)

sex <- sample(c("female", "male"), dim(ods)[2], replace=TRUE)
colData(ods)$sex <- sex
ods <- plotCountCorHeatmap(ods, colCoFactor="sex", normalized=FALSE)
ods <- plotCountCorHeatmap(ods, nCluster=4)
head(colData(ods)$clusterNumber)

mcols(ods)$basepairs <- 1
mcols(ods)$passedFilter <- rowMeans(counts(ods)) > 10
plotFPKM(ods)

plotDispEsts(ods, compareDisp=FALSE)

plotPowerAnalysis(ods)

## Not run:
ods <- findEncodingDim(ods)
plotEncDimSearch(ods)

## End(Not run)
```

Description

This function assembles a results table of significant outlier events based on the given filter criteria. The table contains various information accumulated over the analysis pipeline.

Usage

```
results(object, ...)

## S4 method for signature 'OutriderDataSet'
results(object, padjCutoff = 0.05,
        zScoreCutoff = 0, round = 2, all = FALSE, ...)
```

Arguments

object	An OutriderDataSet
...	Additional arguments, currently not used
padjCutoff	The significant threshold to be applied
zScoreCutoff	If provided additionally a z score threshold is applied
round	Can be TRUE, defaults to 2, or an integer used for rounding with <code>round</code> to make the output more user friendly
all	By default FALSE, only significant read counts are listed in the results. If TRUE all results are assembled resulting in a data.table of length samples x genes

Value

A data.table where each row is an outlier event and the columns contain additional information about this event. Eg padj, l2fc

Examples

```
ods <- makeExampleOutriderDataSet()
ods <- OUTRIDER(ods)
res <- results(ods, all=TRUE)
res
```

sampleExclusionMask *Sample exclusion*

Description

To exclude a sample from the fit process, one can use this function to mask specific samples. This can be used if replicates are present.

Usage

```
sampleExclusionMask(ods, aeMatrix = FALSE)

sampleExclusionMask(ods) <- value
```

Arguments

ods	An OutriderDataSet object
aeMatrix	If TRUE, it returns a 0-1 matrix for the internal autoencoder functions
value	A logical vector of the length of the samples. If TRUE, the corresponding sample will be excluded from the autoencoder fit.

Value

The exclusion vector/matrix.

Examples

```
ods <- makeExampleOutriderDataSet()
sampleExclusionMask(ods) <- sample(c(FALSE, TRUE), ncol(ods), replace=TRUE)

sampleExclusionMask(ods)
```

sizeFactors

SizeFactors accessor and estimation function

Description

Accessor functions for the 'sizeFactors' information in a OutriderDataSet object.

Usage

```
## S4 method for signature 'OutriderDataSet'
sizeFactors(object)

## S4 replacement method for signature 'OutriderDataSet,numeric'
sizeFactors(object) <- value

## S4 replacement method for signature 'OutriderDataSet,`NULL`'
sizeFactors(object) <- value

## S4 method for signature 'OutriderDataSet'
estimateSizeFactors(object)
```

Arguments

object	OutriderDataSet
value	A numeric vector of sizeFactors

Details

The estimation of the size factors can also make use of the existing log geometric means in the object. Those can be loaded from an existing model.

Value

An `OutriderDataSet` with the estimated `sizeFactors`, or with the getter function it returns a numeric vector containing the `sizeFactors`.

See Also

[estimateSizeFactors](#)

Examples

```
ods <- makeExampleOutriderDataSet()
ods <- estimateSizeFactors(ods)
head(sizeFactors(ods))

sizeFactors(ods) <- runif(dim(ods)[2], 0.5, 1.5)
sizeFactors(ods)
counts(ods, normalized=TRUE)[1:10,1:10]
```

Index

- 'sampleExclusionMask<-'
(sampleExclusionMask), 21
- aberrant, 2, 19
- BiocParallelParam, 7, 16
- computeGeneLength, 3
- computeLatentSpace, 4
- computePvalues, 5, 15
- computePvalues, OutriderDataSet-method
(computePvalues), 5
- computeZscores, 6, 15
- computeZscores, OutriderDataSet-method
(computeZscores), 6
- controlForConfounders, 7, 14, 15
- counts, 7
- counts, OutriderDataSet-method (counts),
7
- counts<-, OutriderDataSet, matrix-method
(counts), 7
- DESeqDataSet, 17
- dispersion, (getBestQ), 12
- dispersions, OutriderDataSet-method
(getBestQ), 12
- estimateBestQ, 8
- estimateSizeFactors, 15, 23
- estimateSizeFactors (sizeFactors), 22
- estimateSizeFactors, OutriderDataSet-method
(sizeFactors), 22
- filterExpression, 9
- filterExpression, OutriderDataSet-method
(filterExpression), 9
- findEncodingDim, 10
- findEncodingDim, (findEncodingDim), 10
- findInjectZscore (findEncodingDim), 10
- fit, 11, 15
- fit, OutriderDataSet-method (fit), 11
- fpkm, 9, 12, 12
- fpm, 12
- fpm (fpkm), 12
- getBestQ, 12
- getter_setter_functions (getBestQ), 12
- heatmap.2, 19
- makeExampleOutriderDataSet, 13
- normalizationFactors, 8, 14
- normalizationFactors, OutriderDataSet-method
(normalizationFactors), 14
- normalizationFactors<-, OutriderDataSet, data.frame-method
(normalizationFactors), 14
- normalizationFactors<-, OutriderDataSet, DataFrame-method
(normalizationFactors), 14
- normalizationFactors<-, OutriderDataSet, matrix-method
(normalizationFactors), 14
- normalizationFactors<-, OutriderDataSet, NULL-method
(normalizationFactors), 14
- OUTRIDER, 15
- OutriderDataSet
(OutriderDataSet-class), 16
- OutriderDataSet-class, 16
- padj (getBestQ), 12
- padj, (getBestQ), 12
- plotAberrantPerSample (plotVolcano), 17
- plotCountCorHeatmap (plotVolcano), 17
- plotDispEsts (plotVolcano), 17
- plotDispEsts, OutriderDataSet-method
(plotVolcano), 17
- plotEncDimSearch (plotVolcano), 17
- plotExpressionRank (plotVolcano), 17
- plotFPKM (plotVolcano), 17
- plotFunctions (plotVolcano), 17
- plotPowerAnalysis (plotVolcano), 17
- plotQQ (plotVolcano), 17
- plotVolcano, 17
- pValue (getBestQ), 12
- pValue, (getBestQ), 12
- results, 20
- results, (results), 20
- results, OutriderDataSet-method
(results), 20

results.OutriderDataSet (results), [20](#)
round, [21](#)

sampleExclusionMask, [21](#)
sampleExclusionMask,
 (sampleExclusionMask), [21](#)
sampleExclusionMask<-
 (sampleExclusionMask), [21](#)
sizeFactors, [8](#), [15](#), [22](#)
sizeFactors, OutriderDataSet-method
 (sizeFactors), [22](#)
sizeFactors<- (sizeFactors), [22](#)
sizeFactors<- , OutriderDataSet, NULL-method
 (sizeFactors), [22](#)
sizeFactors<- , OutriderDataSet, numeric-method
 (sizeFactors), [22](#)

theta (getBestQ), [12](#)
theta, (getBestQ), [12](#)

zScore (getBestQ), [12](#)
zScore, (getBestQ), [12](#)