Package 'XBSeq'

March 30, 2021

Type Package

Title Test for differential expression for RNA-seq data

Version 1.22.0

Date 2018-03-27

Author Yuanhang Liu

Maintainer Yuanhang Liu <liuy12@uthscsa.edu>

Description We developed a novel algorithm, XBSeq, where a statistical model was established based on the assumption that observed signals are the convolution of true expression signals and sequencing noises. The mapped reads in non-exonic regions are considered as sequencing noises, which follows a Poisson distribution. Given measureable observed and noise signals from RNA-seq data, true expression signals, assuming governed by the negative binomial distribution, can be delineated and thus the accurate detection of differential expressed genes.

License GPL (>=3)

Imports pracma, matrixStats, locfit, ggplot2, methods, Biobase, dplyr, magrittr, roar

Depends DESeq2, R (>= 3.3)

Suggests knitr, DESeq, rmarkdown, BiocStyle, testthat

VignetteBuilder knitr

biocViews ImmunoOncology, RNASeq, DifferentialExpression, Sequencing, Software, ExperimentalDesign

URL https://github.com/Liuy12/XBSeq

git_url https://git.bioconductor.org/packages/XBSeq

git_branch RELEASE_3_12

git_last_commit 340f654

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

R topics documented:

`XBSeq-package` *Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads*

Description

We developed a novel algorithm, XBSeq, where a statistical model was established based on the assumption that observed signals are the convolution of true expression signals and sequencing noises. The mapped reads in non-exonic regions are considered as sequencing noises, which follows a Poisson distribution. Given measureable observed signal and background noise from RNA-seq data, true expression signals, assuming governed by the negative binomial distribution, can be delineated and thus the accurate detection of differential expressed genes.

Details

Package:	XBSeq
Type:	Package
Version:	1.1.0
Date:	2015-12-02
License:	>=GPL3
Imports:	DESeq2, Biobase, pracma, matrixStats, ggplot2,locfit, methods, BiocGenerics, dplyr, plotly, MetricsGraphics

Author(s)

Yuanhang Liu

Maintainer: Yuanhang Liu <liuy12@uthscsa.edu>

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

apaUsage *Testing differential alternative polyadenylation (apa) usage by using roar*

Description

XBSeq now offers testing differential apa usage via funcitionality provided by roar package

Usage

```
apaUsage(bamTreatment, bamControl, apaAnno, paired = NULL)
```

Arguments

bamTreatment	A list of full path of filenames of bam alignments with data for the treatment condition (by convention it is considered the 'treated' condition:
bamControl	A list of full path of filenames of bam alignments with data for the control condition to be considered.
apaAnno	full path of apa annotation used by roar package. APA annotation for several organisms of various genome build can be downloaded from [here](https://github.com/Liuy12/XBSeq_f For details regarding how to construct APA annotation, please refer to vignette
paired	a vector indicating how samples are paired, leave to NULL if the experiment is not paired.

Details

Bioconductor package roar is used to detect preferential usage of shorter isoforms via alternative poly-adenylation from RNA-seq data. The approach is based on Fisher test to detect disequilibriums in the number of reads falling over the 3' UTRs when comparing two biological conditions.

Value

The resulting data frame will have the "gene_id" of the initial annotation as row names and as columns the m/M ratio for the treatment and control conditions, the roar value and the Fisher test pvalue, expression value of treatment group, expression value of control groups (respectively: mM_treatment, mM_control, roar, pval, treatmentValue, controlValue). If more than one sample has been given for a condition the "pval" column will contain the multiplication of all the comparisons pvalue and there will be other columns containing the pvalues resulting from all the pairwise treatment vs control contrasts, with names "pvalue_X_Y" where X represent the position of the sample in the treatment list of bam files and Y the position for the control list

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

`conditions` *Accessor functions for the 'conditions' information in a XBSeqDataSet object.*

Description

Conditions extract the experimental design information similar as used in DESeq.

Usage

```
## S4 method for signature 'XBSeqDataSet'
conditions(object,...)
## S4 replacement method for signature 'XBSeqDataSet'
conditions(object,...) <- value
```

Arguments

`object`	a XBSeqDataSet
`value`	experimental design information
`...`	Further arguments will be ignored

Value

The experimental design information for a XBSeqDataSet object

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

`XBSeqDataSet`

Examples

```
data(ExampleData)
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
XB <- XBSeqDataSet(Observed, Background, conditions)
conditions(XB)
```

counts — *Accessor functions for the 'counts' information in a XBSeqDataSet object.*

Description

The 'counts' function extract a certian assay element from XBSeqDataSet object. The normalized assay element can be extracted by specifying 'normalized = TRUE'.

Usage

```
## S4 method for signature 'XBSeqDataSet'
counts(object,slot = 3, normalized = FALSE)
```

Arguments

object	a XBSeqDataSet
slot	a integer value to specify which assay element to extract (default to 3)
normalized	whether the normalized assay element should be returned

Details

counts is a function to access an array elemen which is specified by the end user. The difference between this function and the counts function for DESeqDataSet is that this function can be used to access a specific array elememt rather than a pre-defined array element "counts" in the case of DESeqDataSet. By default, the first array element contains information of observed signal. The second array element contains information of background noise. The third array element contains information of estimated true signal after calling the function estimateRealCount.

Value

Either normalized or un-normalized assay element

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

XBSeqDataSet, DESeqDataSet

Examples

```
data(ExampleData)
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
XB <- XBSeqDataSet(Observed, Background, conditions)
str(counts(XB, 1))
```

dispEst *Function to access the dispersion estimation for each gene*

Description

The dispersion estimated for each gene are stored as a data.frame after user called `estimateSCV`

Usage

```
dispEst(object, varname = NA)
dispEst(object, varname = NA) <- value
```

Arguments

`object`	XBSeqDataSet object
`varname`	variable name of dispersion estimates
`value`	The dispersion estimates for each gene

Value

A data.frame which contains the dispersion estimates for each gene

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

`estimateSCV`, `dispTable`, `XBSeqDataSet`

Examples

```
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
data(ExampleData)
XB <- XBSeqDataSet(Observed, Background, conditions)
XB <- estimateRealCount(XB)
XB <- estimateSizeFactors(XB)
XB <- estimateSCV(XB, fitType='local')
str(dispEst(XB))
```

`dispTable` *Access the dispersion table information for a XBSeqDataSet object*

Description

A method adopted from DESeq to examine the dispersion table information for a XBSeqDataSet object

Usage

```
dispTable(object, ...)
```

Arguments

`object`	a XBSeqDataSet
`...`	further argumnts are ignored

Value

Dispersion table information for a XBSeqDataSet object

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

`estimateSCV`, `dispEst`, `XBSeqDataSet`

Examples

```
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
data(ExampleData)
XB <- XBSeqDataSet(Observed, Background, conditions)
XB <- estimateRealCount(XB)
XB <- estimateSizeFactors(XB)
XB <- estimateSCV(XB, fitType='local')
dispTable(XB)
```

`estimateRealCount` *Preliminary step to estimate the true signal based on observed signal and background noise*

Description

Based on the observed signal as well as the background noise, estimate the true signal for each gene.

Usage

```
estimateRealCount(object)
```

Arguments

`object`	A XBSeqDataSet object

Details

The observed signal can be achieved by using HTSeq to count the reads map to exonic regions. The background noise can be extracted by using HTSeq the second time to count the reads map to non-exonic regions, the regions we defined by excluding potential functional elements. The the underneath true signal is estimated by the simple subtraction of observed signal and background noise. The true signal of genes with background noise larger than observed signal will be assigned as 0.

Value

A matrix contains the estimated true signal for each gene with the same length as observed signal.

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

`counts`

Examples

```
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
data(ExampleData)
XB <- XBSeqDataSet(Observed, Background, conditions)
XB <- estimateRealCount(XB)
str(counts(XB, 3))
```

`estimateSCV` *Estimate squared coefficient of variation for each gene*

Description

A similar method is applied to estimate the SCV for each gene based on the method used in DESeq

Usage

```
## S4 method for signature 'XBSeqDataSet'
estimateSCV( object, method = c( "pooled", "per-condition", "blind" ), sharingMode = c( "maximum", "
   fitType = c("local","parametric"),
   locfit_extra_args=list(), lp_extra_args=list(), ... )
```

Arguments

`object` a `XBSeqDataSet` with size factors.

`method` There are three ways how the empirical dispersion can be computed:

- `pooled` - Use the samples from all conditions with replicates to estimate a single pooled empirical dispersion value, called "pooled", and assign it to all samples.
- `per-condition` - For each condition with replicates, compute a gene's empirical dispersion value by considering the data from samples for this condition. For samples of unreplicated conditions, the maximum of empirical dispersion values from the other conditions is used.
- `blind` - Ignore the sample labels and compute a gene's empirical dispersion value as if all samples were replicates of a single condition. This can be done even if there are no biological replicates. This method can lead to loss of power.

`sharingMode` After the empirical dispersion values have been computed for each gene, a dispersion-mean relationship is fitted for sharing information across genes in order to reduce variability of the dispersion estimates. After that, for each gene, we have two values: the empirical value (derived only from this gene's data), and the fitted value (i.e., the dispersion value typical for genes with an average expression similar to those of this gene). The `sharingMode` argument specifies which of these two values will be written to the `dispEst` and hence will be used by the functions `XBSeqTest`

- `fit-only` - use only the fitted value, i.e., the empirical value is used only as input to the fitting, and then ignored. Use this only with very *few* replicates, and when you are not too concerned about false positives from dispersion outliers, i.e. genes with an unusually high variability.
- `maximum` - take the maximum of the two values. This is the conservative or prudent choice, recommended once you have at least three or four replicates and maybe even with only two replicates.
- `gene-est-only` - No fitting or sharing, use only the empirical value. This method is preferable when the number of replicates is large and the empirical dispersion values are sufficiently reliable. If the number of replicates is small, this option may lead to many cases where the dispersion of a gene is accidentally underestimated and a false positive arises in the subsequent testing.

`fitType`
- `parametric` - Fit a dispersion-mean relation of the form `dispersion = asymptDisp + extraPois / mean` via a robust gamma-family GLM. The coefficients `asymptDisp` and `extraPois` are given in the attribute `coefficients` of the `dispFunc` in the `fitInfo`.
- `local` - Use the locfit package to fit a dispersion-mean relation, as described in the DESeq paper.

`locfit_extra_args, lp_extra_args`
(only for `fitType=local`) Options to be passed to the `locfit` and to the `lp` function of the locfit package. Use this to adjust the local fitting. For example, you may pass a value for `nn` different from the default (0.7) if the fit seems too smooth or too rough by setting `lp_extra_agrs=list(nn=0.9)`. As another example, you can set `locfit_extra_args=list(maxk=200)` if you get the error that locfit ran out of nodes. See the documentation of the `locfit` package for details. In most cases, you will not need to provide these parameters, as the defaults seem to work quite well.

`...` extra arguments are ignored

Details

The details regarding which option to choose can be found in the DESeq help page. Generally speaking, if you have less number of replicates (<=3), set `method="pooled"`. Otherwise, try `method="per-condition"`. We revised the code to estimate the variance of the true signal by using variance sum law rather than calculate the variance directly.

Value

The `XBSeqDataSet` cds, with the slots `fitInfo` and `dispEst` updated.

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

`XBSeqDataSet`

Examples

```
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
data(ExampleData)
XB <- XBSeqDataSet(Observed, Background, conditions)
XB <- estimateRealCount(XB)
XB <- estimateSizeFactors(XB)
XB <- estimateSCV(XB, fitType='local')
str(fitInfo(XB))
```

`ExampleData` *Example Datasets used in the manual pages as well as in vignette*

Description

Example datasets used in manual pages and vignette by carrying out HTSeq procedure for exonic mapped reads (Observed) and non-exonic mapped reads (Background) and gene length information (genelength).

Usage

```
data(ExampleData)
```

Format

ExampleData contains three data.frames. Two of them are expression matrix. One is called 'Observed'. One is called 'Background'. For the two data.frames, rows represent exonic or non-exonic region mapped reads for each gene. Columns represent each sample. Both the two data.frames have total of 22609 number of rows and 6 number of columns. There is also another data.frame containing the gene length information.

Details

In order to use XBSeq for testing DE, we need to run HTSeq twice to measure the reads mapped to exonic regions (observed signal) and non-exonic regions (background noise). Firstly, we need to construct the gtf annotation file to measure the background noise:

- Download refFlat table from UCSC database (http://genome.ucsc.edu) and create the preliminary list of gene-free regions,
- Download tables of (a) all_mrna; (b) ensGene; (c) pseudoYale60Gene; (d) vegaGene;, (e)xenoMrna, and (f) xenoRefGene from UCSC database and remove regions appear in any of them from the gene-free regions,
- To guarantee gene-free regions are far enough from exonic regions, trim 100 bps from both sides of intronic regions and 1,000 bps from both sides of inter-genic regions,
- Shift each exon of a gene to the right nearest gene-free region. Most of the shifted genes remain the same as the original structures of the genes,
- If the nearby gene-free region is too short, we may only preserve the exon size features but not the whole gene structure. The priority of shifting a region is: i) nearest right gene-free region, 2) nearest left gene-free region; 3) the second right nearest gene-free region and so on until the shift region of the original exon fits, and
- Shift each exon of a gene to the right nearest gene-free region. Most of the shifted genes remain the same as the original structures of the genes,
- At last, we considered the shifted regions as the non-exonic regions for each gene and a final .gtf file was created

We carried out HTSeq procedure twice by using a a mouse RNA-seq dataset, which contains 3 replicates of wild type mouse liver tissues (WT) and 3 replicates of Myc transgenic mouse liver tissues (MYC).The dataset is obtained from Gene Expression Omnibus (GSE61875) (http://www.ncbi.nlm.nih.gov/geo/query/ac
. The two datasets can be loaded via `data(ExampleData)` after loading the XBSeq library.

The annotation for measuring the background noise can be generated by following the previous steps. Firstly, generate preliminary gene-free regions by calling the function `exonFreeRegionShift.pl <-EX exon-GTF file > <-FR gene free region>`.

Then remove the potential functional elements by calling the function `GEFRshift.pl <-G gene-GTF.gtf > <-I intronRegion.tsv> <-T integenicRegion.tsv> optional: -m mRNA.bed -x xenoMrna.bed -z xenoRefGene.bed -e ensGene.bed -p pseudoGene.bed -v vegaGene.bed -b`.

We have already generated gtf files for human (hg18 and hg19) and mouse (mm9 and mm10) and deposited in github. If you would like to generate your own gtf files, the scripts to generate the files ,which are written in perl, are available in the package subfolder XBSeq\inst\scripts\. The scripts are also deposited in github (https://github.com/Liuy12/XBSeq).

Value

Three data.frames as described in format section.

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

---

fitInfo *Accessor function for the fitInfo objects in a XBSeqDataSet*

Description

Same method is adopted from DESeq to access the fit information from a XBSeqDataSet

Usage

```
fitInfo( object, name)
```

Arguments

`object`	a XBSeqDataSet
`name`	if `estimateSCV` was called with `method="per-condition"` a name hasd to specified. Try `ls(XB@fitInfo`.

Value

A list containing fitting information for a XBSeqDataSet object:

`perGeneSCVEsts`	SCV estimates for each gene, which has the same length as the number of rows as the assay elements in an object
`SCVFunc`	The function used to predict the fitted SCV
`fittedSCVEsts`	The fitted SCV estimates for each gene, which is of the same length as perGeneSCVEsts
`df`	Integer value indicating the degree of freedom
`sharingMode`	The sharing mode argument specified by the user

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

estimateSCV, XBSeqDataSet

Examples

```
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
data(ExampleData)
XB <- XBSeqDataSet(Observed, Background, conditions)
XB <- estimateRealCount(XB)
XB <- estimateSizeFactors(XB)
XB <- estimateSCV(XB, fitType='local')
str(fitInfo(XB))
```

---

getSignalVars *Estimate variance of the signal based on variance summation law*

Description

Based on variance of observed signal as well as background noise, estimate the variance of the true signal

Usage

```
getSignalVars(counts, bgcounts)
```

Arguments

counts	A data frame or matrix which contains the observed signal (expression level) information for an experiment. Rows represent genes and Columns represent samples. Please refer details for more information.
bgcounts	A data frame or matrix which contains the background noise information for an experiment. Rows represent genes and Columns represent samples. Please refer details for more information.

Details

Observed signal are the reads mapped to the exonic regions which can be obtained by applying HTSeq procedure with GTF files of exonic regions. Background noise are the reads mapped to the non-exonic regions which can be obtained by applying HTSeq procedure with GTF files of non-exonic regions we defined by certain criteria. Details regarding how to carry out the HTSeq procedure for observed signal as well as background noise can be found in the vignette of XBSeq. One example dataset is provided in ExampleData.

By assuming that the true signal and background noise are independent, the variance of the underneath signal (σ_s^2) can be estimated by applying variance summation law:

$$\sigma_s^2 = \sigma_x^2 + \sigma_b^2 - 2\rho\sigma_x\sigma_b$$

where σ_x^2 and σ_b^2 are variance for observed signal and background noise respectively.

Value

A matrix with the same number of rows as `counts`. Rows represent the estimated variance of true signal for each gene.

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

`estimateSCV`

Examples

```
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
data(ExampleData)
data_var <- getSignalVars(Observed, Background)
```

`MAplot` *Generate maplot after differential expression test*

Description

Generate maplot after differential expression test based on ggplot2

Usage

```
MAplot(stats, ylim, padj = TRUE, pcuff = 0.1, lfccuff = 1,
   linecol = "red3", xlab = "mean of normalized counts",
   ylab = expression(log[2] ~ fold ~ change), shape)
```

Arguments

`stats`	The output of `XBSeqTest`
`ylim`	Range of limit for y axis
`padj`	Whether to use adjusted p value or not
`pcuff`	Threshold for pvalue
`lfccuff`	Log fold change cutoff

linecol	Colour of horizontal line
xlab	Lable for x axis
ylab	Lable for y axis
shape	The shape of the points used

Details

Generate classic MAplot for DE analysis using ggplot2, where A and M are from slot baseMean and slot log2FoldChange of the test statistics aftering calling XBSeqTest. The ggplot2 package generally generate figures of better quality as well as give user better control of the plotting system compared with the base plotting system.

Value

MAplot of test statistics

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

XBSeqTest

Examples

```
conditions <- c(rep('C1', 3), rep('C2', 3))
data(ExampleData)
Stats <- XBSeq(Observed, Background, conditions)
MAplot(Stats)
```

---

plotSCVEsts *Plot estimated squared coefficient of variation*

Description

Plot estimated SCV based on ggplot2

Usage

```
plotSCVEsts(XB, name = NULL, ymin, linecol = "red3",
   xlab = "mean of normalized counts", ylab = "SCV")
```

Arguments

XB	A XBSeqDataSet object
name	The name of the fit information. Only specify this if you choose method="per-condition"
ymin	The limit of y axis
linecol	The linecolour of the SCV-mean trend
xlab	The lable of x axis
ylab	The lable of y axis

Value

Summary plot for the fitting and estimation of scv

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

estimateSCV

Examples

```
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
data(ExampleData)
XB <- XBSeqDataSet(Observed, Background, conditions)
XB <- estimateRealCount(XB)
XB <- estimateSizeFactors(XB)
XB <- estimateSCV(XB, fitType='local')
plotSCVEsts(XB)
```

XBplot *Examine the distribution of observed signal and background noise*

Description

Function to viewlize the distribution of observed signal X and background noise B across all genes for one specified sample

Usage

```
XBplot(XB, Samplenum = NULL, unit = c('counts', 'LogTPM'), Libsize = NULL, Genelength = NULL, xlab =
```

Arguments

XB	An XBSeqDataSet object
Samplenum	An integer number to specify which sample to examine
unit	Whether to examine the distrbution in 'counts' unit or 'LogTPM' unit. 'LogTPM' is generally recommended
Libsize	A single integer indicating the library size of the sample. By default, the sum of all reads mapped to exonic regions are used.
Genelength	A numeric vector containing genelength information. Please make sure the length and order of the gene length information is the same as arrays in the XB object.
xlab	lab for x axis
ylab	Lable for y axis
col	A vector of two colours for observed signal and background noise
alpha	A vector of two numeric numbers indicating transparency

Details

We strongly recommended users to apply XBplot to their datasets before differential expression analysis. According to our experience, for XBplot in 'logTPM' unit, the peak of distribution of background noise generally coinsides with the left hump of distribution of observed signal.

Value

Plot of distribution of observed signal and background noise.

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

ExampleData

Examples

```
conditions <- c(rep('C1', 3), rep('C2', 3))
data(ExampleData)
XB <- XBSeqDataSet(Observed, Background, conditions)
XBplot(XB, Samplenum = 1, unit = "LogTPM", Genelength = genelength[,2])
```

XBSeq *Express function to carry out XBSeq analysis*

Description

A wrapper function to carry out XBSeq analysis procedure

Usage

```
XBSeq(counts, bgcounts, conditions, method = "pooled",
  sharingMode = "maximum", fitType = "local", pvals_only = FALSE, paraMethod='NP', big_count = 900)
```

Arguments

counts	A data.frame or matrix contains the observed signal
bgcounts	A data.frame or matrix contains the background noise
conditions	A factor to specify the experimental design
method	Method used to estimate SCV
sharingMode	Mode of sharing of information
fitType	Option to fit mean-SCV relation
pvals_only	Logical; Specify whether to extract pvalues only
paraMethod	Method to use for estimation of distribution parameters, 'NP' or 'MLE'. See details section for details
big_count	An integer specify a count number above which where be considerred as 'big' and beta approximation will be used instead for testing differential expression

Details

This is the express function for carry out differential expression analysis. Two methods can be choosen from for paraMethod. 'NP' stands for non-parametric method. 'MLE' stands for maximum liklihood estimation method. 'NP' is generally recommended for experiments with replicates smaller than 5.

Value

A data.frame with following columns:

id	rownames of XBSeqDataSet
baseMean	The basemean for all genes
baseMeanA	The basemean for condition 'A'
baseMeanB	The basemean for condition 'B'
foldChange	The fold change compare condition 'B' to 'A'
log2FoldChange	The log2 fold change
pval	The p value for all genes
padj	The adjusted p value for all genes

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

estimateRealCount, XBSeqDataSet, estimateSCV, XBSeqTest

Examples

```
conditions <- c(rep('C1', 3), rep('C2', 3))
data(ExampleData)
Stats <- XBSeq(Observed, Background, conditions)
```

XBSeqDataSet-class *Class* "XBSeqDataSet"

Description

XBSeqDataSet is a subclass of "DESeqDataSet", used to store the input values, intermediate calculations and results of an analysis of differential expression. Different from the original DESeqDataSet class, XBSeqDataSet has some extra slots including:

- fitInfo: An object of environment class which contains the scv fitting information for a XBSeqDataSet object
- dispTable: An object of character class which indicates method used for scv fitting. Details can be found in estimateSCV.
- conditions: An object of factor class which contains the experimental design information for a XBSeqDataSet object
- dispEst: An object of list class which contains the final dispersion estimates for each gene. Details can be found in dispEst

Usage

```
XBSeqDataSet(counts, bgcounts, conditions, sizeFactors=NULL, ...)
```

Arguments

counts	A data frame or matrix which contains the observed signal for each gene across all the samples. Rows represent genes and columns represent samples.
bgcounts	A data frame or matrix which contains the background noise for each gene across all the samples. Rows represent genes and columns represent samples.
conditions	Object of class "character". The conditions for the experimental design.
sizeFactors	Numeric vector which contains normalizing factors for the data matrix. In most cases, it is recommended that you calculate sizeFactors by estimateSizeFactors. You are also able to provide sizeFactors yourself.
...	Further arguments provided will be ignored

Value

A XBSeqDataSet object.

Methods

conditions `signature(object = "XBSeqDataSet")`: ...

conditions<- `signature(object = "XBSeqDataSet")`: ...

counts `signature(object = "XBSeqDataSet")`: ...

dispTable `signature(object = "XBSeqDataSet")`: ...

dispEst `signature(object = "XBSeqDataSet")`: ...

dispEst<- `signature(object = "XBSeqDataSet")`: ...

fitInfo `signature(object = "XBSeqDataSet")`: ...

fitInfo<- `signature(object = "XBSeqDataSet")`: ...

estimateSCV `signature(object = "XBSeqDataSet")`: ...

estimateSizeFactors `signature(object = "XBSeqDataSet")`: ...

estimateRealCount `signature(object = "XBSeqDataSet")`: ...

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

`estimateSCV`, `conditions`, `dispEst`, `dispTable`, `fitInfo`, `DESeqDataSet`, `counts`, `estimateRealCount`

Examples

```
data(ExampleData)
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
XB <- XBSeqDataSet(Observed, Background, conditions)
str(XB)
```

XBSeqTest *XBSeq test for differential expression*

Description

The same method is adopted from DESeq for testing differential expression

Usage

```
XBSeqTest(XB, condA, condB, pvals_only = FALSE, method = c("NP", "MLE"), big_count = 900)
```

Arguments

XB	A XBSeqDataSet object
condA	Factor level specified for condition A
condB	Factor level specified for condition B
pvals_only	Logical;whether or not only extract p values
method	method to use for estimation of distribution parameters, 'NP' or 'MLE'. See details section for details
big_count	An integer specify a count number above which where be considerred as 'big' and beta approximation will be used instead for testing differential expression

Details

Differential expression analysis based on statistical methods proposed for DESeq. Details about the method can be found in DESeq manual page. Two methods can be choosen from for method. 'NP' stands for non-parametric method. 'MLE' stands for maximum liklihood estimation method. 'NP' is generally recommended for experiments with replicates smaller than 5.

Value

A data.frame with following columns:

id	rownames of XBSeqDataSet
baseMean	The basemean for all genes
baseMeanA	The basemean for condition 'A'
baseMeanB	The basemean for condition 'B'
foldChange	The fold change compare condition 'B' to 'A'
log2FoldChange	The log2 fold change
pval	The p value for all genes
padj	The adjusted p value for all genes

Author(s)

Yuanhang Liu

References

H. I. Chen, Y. Liu, Y. Zou, Z. Lai, D. Sarkar, Y. Huang, et al., "Differential expression analysis of RNA sequencing data by incorporating non-exonic mapped reads," BMC Genomics, vol. 16 Suppl 7, p. S14, Jun 11 2015.

See Also

XBSeq, estimateSCV

Examples

```
data(ExampleData)
conditions <- factor(c(rep('C1', 3), rep('C2', 3)))
XB <- XBSeqDataSet(Observed, Background, conditions)
XB <- estimateRealCount(XB)
XB <- estimateSizeFactors(XB)
XB <- estimateSCV(XB)
Teststas <- XBSeqTest(XB, levels(conditions)[1L], levels(conditions)[2L])
str(Teststas)
```

Index