Package ‘doseR’

May 28, 2024

Type Package
Title doseR
Version 1.20.0
Author AJ Vaestermark, JR Walters.
Maintainer ake.vastermark <ake.vastermark@gmail.com>

Description doseR package is a next generation sequencing package for sex chromosome dosage compensation which can be applied broadly to detect shifts in gene expression among an arbitrary number of pre-defined groups of loci. doseR is a differential gene expression package for count data, that detects directional shifts in expression for multiple, specific subsets of genes, broad utility in systems biology research. doseR has been prepared to manage the nature of the data and the desired set of inferences. doseR uses S4 classes to store count data from sequencing experiment. It contains functions to normalize and filter count data, as well as to plot and calculate statistics of count data. It contains a framework for linear modeling of count data. The package has been tested using real and simulated data.

License GPL
Encoding UTF-8
RoxygenNote 6.1.1
Imports edgeR, methods, stats, graphics, matrixStats, mclust, lme4, RUnit, SummarizedExperiment, digest, S4Vectors
Suggests BiocStyle, knitr, rmarkdown
VignetteBuilder knitr
biocViews Infrastructure, Software, DataRepresentation, Sequencing, GeneExpression, SystemsBiology, DifferentialExpression

Depends R (>= 3.6)
git_url https://git.bioconductor.org/packages/doseR
git_branch RELEASE_3_19
git_last_commit 9fded83
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-28
Description

This function filters the expression using DAFS (see ref). This is a core function invoked by the DAFS wrapper.

Usage

dafs(VEC1, PL0T)

Arguments

 brains Vector 1, not intended for user interaction.
 PL0T Boolean, toggles plotting.

Details

This function filters the expression, using Data Adaptive Flag method.

Value

Returns vx[which.min(vv)] to wrapper function.
Author(s)
AJ Vaestermark, JR Walters.

References
BMC Bioinformatics, 2014, 15:92

Examples

```r
library(mclust)
data(hmel.se)
f_se <- dafsFilter(se)
```

---

**dafsFilter**

*dafsFilter Function to filter expression data within an object.*

### Description
This function filters the expression of the supplied object, by invoking the `dafsFilter` (dafs) function. `dafsFilter` is a filtering function used to remove rows (genes) of various expression data.

### Usage

```r
dafsFilter(se, PLOT=TRUE)
```

### Arguments

- **se**
  A SummarizedExperiment object.

- **PLOT**
  Boolean, toggles plotting.

### Details
This function filters the expression of the supplied object using a Data Adaptive Flag filter. The internal function uses a vector to store Kolmogorov Smirnov distance statistics, loops through cuts of the data to determine targeted K-S statistic, selects data greater than a quantile and runs Mclust on that data to determine theoretical distribution. The wrapper uses `simpleFilter` to determine first left-most local minima (using the Earth library).

### Value
Returns an invisible, filtered SummExp object.

### Author(s)
AJ Vaestermark, JR Walters.
References

BMC Bioinformatics, 2014, 15:92

Examples

```r
library(mclust)
data(hmel.se)
f_se <- dafsFilter(se)
```

```r
generateStats(se, groupings="annotation.ZA")
```

---

**generateStats**

Statistics function to summarize expression data of an s.e. object.

**Description**

`generateStats` is a summary function used on various expression data.

**Usage**

```r
generateStats(se, groupings=NULL, mode_mean=TRUE, LOG2=TRUE)
```

**Arguments**

- `se`: A SummarizedExperiment object.
- `groupings`: A grouping (annotation column), e.g. `groupings="something"`.
- `mode_mean`: Boolean, Calculate RowMeans or RowMedians.
- `LOG2`: Boolean, Calculate LOG2.

**Details**

This function completes summary statistics of the expression of the supplied s.e. object.

**Value**

Returns an invisible list of summary statistics, kruskal test and raw data of an s.e. object.

**Author(s)**

AJ Vaestermark, JR Walters.

**References**

The "doseR" package, 2018 (in press).

**Examples**

```r
data(hmel.se)
generateStats(se, groupings="annotation.ZA")
```
getLibsizes3  Get Lib Sizes.

Description

getLibsizes3 method for SummarizedExp class object, derived from getLibsizes2

Usage

getLibsizes3(se, subset = NULL,
estimationType = c("quantile", "total",
"edgeR"), quantile = 0.75, ...)

Arguments

se A SummarizedExperiment object.
subset Value
estimationType e.g. quantile, total, edgeR.
quantile A quantile, expressed as e.g. 0.75.
... Passthrough arguments.

Value

Libsize value

Author(s)

AJ Vaestermark, JR Walters.

References

The "doseR" package, 2018 (in press).

Examples

data(hmel.se)
getLibsizes3(se)
Description

This function is an LME4 wrapper for dosage analysis.

Usage

\texttt{glSeq(dm, model, \ldots)}

Arguments

- \texttt{dm}: The dm data. Generally a reformatted object from \texttt{se.DM}.
- \texttt{model}: The model. Expressed using standard LME4 syntax, see vignette.
- \texttt{\ldots}: passthrough arguments.

Details

This function is an lme4 wrapper.

Value

Returns LME4 output.

Author(s)

AJ Vaestermark, JR Walters.

References

The "doseR" package, 2018 (in press).

Examples

```
data(hmel.se)
f_se <- quantFilter(se, lo.bound = 0.4, hi.bound = 0.5)
dm <- se.DM(f_se)
glSeq(dm, \"-1 + replicate\")
```
**Hmel data set**

**Description**

Hmel data set

**Usage**

```r
data("hmel.data.doser")
```

**Format**

An object of class list of length 3.

**References**

To demonstrate a typical workflow using doseR, we will use data from a study in Heliconius butterflies. *H. melpomene* has 20 autosomes with chromosome 21 being the "Z" sex chromosome. Like all Lepidoptera, this is a female-heterogametic species, so males are diploid for the Z while females have a single Z and W chromosomes. The relevant data are included in the doseR package, which must be installed. These test data are unpublished.

---

**iqrxFilter**

_iqrxFilter Function to filter expression data within an s.e. object._

**Description**

This function filters the expression of the supplied se object. iqrxFilter is a filtering function used to remove rows (genes) of various expression data.

**Usage**

```r
iqrxFilter(se, iqr_multi = 1.5, MEDIAN = FALSE, na.rm = TRUE)
```

**Arguments**

- **se**
  - A SummarizedExperiment object.
- **iqr_multi**
  - Numeric multiplier; removes any outliers that are iqr_multi times the mid-50 percentile distance greater or less than the 25th and 75th percentiles, by default
- **MEDIAN**
  - Boolean, Calculate RowMeans or RowMedians.
- **na.rm**
  - Boolean, NA removal.
Details
This function filters the expression of the supplied object, based on a selected percentage cutoff and selected interquartile range multiplier. The function iqrxFilter will: 1) log-base two transform all RPKM values (obligatory); (2) remove any outliers that were 1.5 times the mid-50 percentile distance greater or less than the 75th and 25th percentiles (by default), respectively; and (3) uses mean values and instead of median values (by default).

Value
Returns a filtered SummarizedExperiment object.

Author(s)
AJ Vaestermark, JR Walters.

References
Jue et al. BMC Genomics 2013 14:150

Examples
```r
data(hmel.se)
f_se <- iqrxFilter(se)
```

---

make_RPKM

**make_RPKM** Make RPKM.

Description
make_RPKM populates RPKM slot of SummarizedExperiment S4 object.

Usage
```r
make_RPKM(se)
```

Arguments
```
se
```
A SummarizedExperiment object.

Value

RPKM populated object

Author(s)
AJ Vaestermark, JR Walters.
plotExpr

References

The "doseR" package, 2018 (in press).

Examples

data(hmel.se)
SummarizedExperiment::assays(se)$rpkm <- make_RPKM(se)

plotExpr Function to generate a boxplot (expression) for a Summa-
izedExperiment object based on the replicate data.

Description

This function generates a boxplot of FPKM expression values from the supplied object. FPKM
values are averaged across replicates and partitioned among groups of loci as specified in a selected
column from the annotation slot of the provided object.

Usage

plotExpr(se, groupings= NULL, mode_mean=TRUE, 
treatment=levels(colData(se)$Treatment), 
LOG2=TRUE, clusterby_grouping=TRUE, ...)

Arguments

se A SummarizedExp object containing FPKM values and at least one annotation
column.

logings Specifies which column in the dataframe of the annotation slot that will be used
to group loci in the boxplot. Can provide either a character value matching
the column name, or a single numerical value used as an index of dataframe
columns.

mode_mean Logical. If TRUE then FPKM values are averaged by mean across replicates
within treatment. If FALSE, values are averaged by median.

treatment A character vector indicating which treatments (i.e. levels in the replicates slot
vector) will be plotted. Order matters, and controls the ordering of treatments
represented in the boxplot.

LOG2 Logical. If TRUE then average FPKM values are Log2 transformed.

clusterby_grouping Logical. If TRUE then boxplots are arranged by locus annotation grouping.
If FALSE they are arranged by treatment levels, as indicated in the treatment
argument.

... Additional named arguments and graphical parameters passed to the boxplot
function.
Details

This function generates boxplots to visualize the distribution of FPKM expression values provided in an object, arranged by selected treatments and locus annotations. FPKM values are averaged (mean or median) within selected treatments, to provide a single expression value per locus per treatment. Loci are partitioned into groupings based on a specified column in the dataframe of annotations slot of the object. Thus a box is drawn for each grouping of loci for each treatment indicated. Desired treatments and their ordering are specified by the treatment argument. Groupings are arranged by sort order of the annotation column indicated, and can thus be controlled by providing a factor with a pre-specified level order. By default (clusterby_grouping = TRUE), boxes are arranged by annotation group first, and then by treatment, but setting this option to FALSE arranges boxes by treatment and then annotation group. This function uses the base graphics boxplot function to generate the plot, so can accept all relevant graphical arguments for customizing the figure; see boxplot for details.

Value

Returns an invisible data frame containing values and labels used to generate the figure.

Author(s)

AJ Vaestermark, JR Walters.

References

The 'doseR' package, 2018 (in press).

Examples

data(hmel.se)
plotExpr(se, groupings = "annotation.ZA", treatment = 'Male')

plotMA.se Function to make MA plot.

Description

This function generates MA plot.

Usage

plotMA.se(se, samplesA, samplesB, scale = NULL, xlab = 'A', ylab = 'M', ...)

Arguments

- `se` A SummarizedExperiment object.
- `samplesA` Either a character vector, identifying sample set A by either replicate name or sample name, or a numerical vector giving the columns of data in the object that forms sample set A.
- `samplesB` Either a character vector, identifying sample set B by either replicate name or sample name, or a numerical vector giving the columns of data in the object that forms sample set B.
- `scale` If given, defines the scale on which the log-ratios will be plotted.
- `xlab` Label for the X-axis. Defaults to 'A'.
- `ylab` Label for the Y-axis. Defaults to 'M'.
- `...` Any other parameters to be passed to the plot function.

Details

This function makes MA plot from SummExperiment object.

Value

Returns MA plot.

Author(s)

AJ Vaestermark, JR Walters.

References

The 'doseR' package, 2018 (in press).

Examples

data(hmel.se)
plotMA.se(se, samplesA = 'Male', samplesB = 'Female')

Description

This function plots the expression of the supplied object, representing ratios between a pair of selected treatments as a boxplot for each group in the selected annotation column.
Usage

```r
plotRatioBoxes(se, groupings= NULL, treatment1=NULL, treatment2=NULL,
mode_mean=TRUE, LOG2=TRUE, ...)
```

Arguments

- `se` A SummarizedExperiment object.
- `groupings` A grouping (annotation column), e.g. groupings="something".
- `treatment1` Symbol, treatment 1.
- `treatment2` Symbol, treatment 2.
- `mode_mean` Boolean, Calculate RowMeans or RowMedians.
- `LOG2` Boolean, Calculate LOG2.
- `...` Passthrough arguments to boxplot (additional arguments affecting the summary produced).

Details

This function boxplots expression of the supplied object using ratios of treatment1/treatment2.

Value

Returns an invisible data frame containing the values.

Author(s)

AJ Vaestermark, JR Walters.

References

The "doseR" package, 2018 (in press).

Examples

```r
data(hmel.se)
plotRatioBoxes(se, groupings='annotation.ZA', treatment1 = 'Male',
treatment2 = 'Female')
```
plotRatioDensity Function to plot density of ratios between two treatments (using groupings from an annotation column) within an object.

Description

This function plots the expression of the supplied object, using ratios between a pair of selected treatments.

Usage

plotRatioDensity(se, groupings=NULL, treatment1=NULL, treatment2=NULL, mode_mean=TRUE, LOG2=TRUE,...)

Arguments

se
A SummarizedExperiment object.

groupings
A grouping (annotation); groupings="annotation.ZA"

treatment1
Symbol, treatment 1.

treatment2
Symbol, treatment 2.

mode_mean
Boolean, Calculate RowMeans or RowMedians.

LOG2
Boolean, Calculate LOG2.

...
Passthrough arguments to boxplot (additional arguments affecting the summary produced).

Details

This function plots expression of the supplied object using ratios of treatment1/treatment2.

Value

Returns an invisible data frame containing the x-values and corresponding density for each applicable annotation column entry.

Author(s)

AJ Vaestermark, JR Walters.

References

The "doseR" package, 2018 (in press).

Examples

data(hmel.se)
plotRatioDensity(se, groupings='annotation.ZA', treatment1 = 'Male', treatment2 = 'Female', lty=1, type="l")
quantFilter Function to filter expression data of a SummarizedExperiment object.

Description

This function filters the expression of the supplied object; quantFilter is a filtering function used to remove rows (genes) of various expression data.

Usage

quantFilter (se, lo.bound=.25, hi.bound=.75, MEDIAN = FALSE, na.rm = TRUE)

Arguments

se A SummarizedExperiment object.
lo.bound The lower cutoff, expressed as a percentage.
hi.bound The upper cutoff, expressed as a percentage.
MEDIAN Boolean, Calculate RowMeans or RowMedians.
a.rm Boolean, NA removal.

Details

This function filters the expression of the supplied object, based on a selected percentage cutoff.

Value

Returns a filtered SummarizedExperiment object.

Author(s)

AJ Vaestermark, JR Walters.

References

The "doseR" package, 2018 (in press).

Examples

data(hmel.se)
f_se <- quantfilter(se, lo.bound=0.5)
**se**  

*Hmel data set*

**Description**

Hmel data set

**Usage**

```r
data("hmel.se")
```

**Format**

An object of class `SummarizedExperiment` with 13619 rows and 6 columns.

**References**

To demonstrate a typical workflow using `doseR`, we will use data from a study in Heliconius butterflies. *H. melpomene* has 20 autosomes with chromosome 21 being the “Z” sex chromosome. Like all Lepidoptera, this is a female-heterogametic species, so males are diploid for the Z while females have a single Z and W chromosomes. The relevant data are included in the `doseR` package, which must be installed. These test data are unpublished. This is the `SummarizedExperiment` version of the sample data.

---

**se.DM**  

*se.DM Function to convert SummarizedExperiment object to LME4 input.*

**Description**

This function generates LME4 input.

**Usage**

```r
se.DM(se, weightByLL = TRUE)
```

**Arguments**

- `se`: A `SummarizedExperiment` object containing FPKM values and at least one annotation column.
- `weightByLL`: Logical, weigh by log likelihood score.

**Details**

This function converts `SummarizedExperiment` object.
simpleFilter

Description
	simpleFilter is a filtering function used to remove rows (genes) of various expression data.

Usage

simpleFilter(se, mean_cutoff=NULL, min_cutoff=NULL, median_cutoff=NULL, counts=TRUE)

Arguments

  se          A SummarizedExperiment object.
  mean_cutoff The lower cutoff, using mean.
  min_cutoff  The cutoff, expressed as a minimum acceptable value.
  median_cutoff The cutoff, using a median value.
  counts      Boolean, use raw counts data (default) or RPKM.

Details

  This function filters the expression of the supplied obj.

Value

  Returns a filtered SummarizedExperiment object.

Author(s)

  AJ Vaestermark, JR Walters.
test_diffs

References
The "doseR" package, 2018 (in press).

Examples

data(hmel.se) ; f_se <- simpleFilter(se, mean_cutoff=0.5)

test_diffs

Description
generateStats is a summary function used on various expression data, using ratios between selected treatments.

Usage
test_diffs(se, groupings= NULL, treatment1=NULL, treatment2=NULL,
mode_mean=TRUE, LOG2=TRUE)

Arguments
  se A SummarizedExperiment object.
  groupings A grouping (annotation column); groupings="annotation.ZA".
  treatment1 Symbol, treatment 1.
  treatment2 Symbol, treatment 2.
  mode_mean Boolean, Calculate RowMeans or RowMedians.
  LOG2 Boolean, Calculate LOG2.

Details
This function completes summary statistics of the expression of the supplied SummarizedExperiment object.

Value
Returns an invisible list of summary statistics, kruskal test and raw data of an object, using ratios between selected treatments.

Author(s)
AJ Vaestermark, JR Walters.

References
The "doseR" package, 2018 (in press).
Examples

data(hmel.se)
test_diffs(se, groupings='annotation.ZA', treatment1="Male",
treatment2="Female" )
Index

* datasets
  * hmel.dat, 7
  * se, 15

dafs, 2
dafsFilter, 3

generateStats, 4
getLibsizes3, 5
glSeq, 6

hmel.dat, 7
iqrxFilter, 7

make_RPKM, 8

plotExpr, 9
plotMA.se, 10
plotRatioBoxes, 11
plotRatioDensity, 13

quantFilter, 14

se, 15
se.DM, 15
simpleFilter, 16

test_diffs, 17