Package ‘diffUTR’

April 10, 2024

Type Package

Title diffUTR: Streamlining differential exon and 3’ UTR usage

Version 1.10.0

Depends R (>= 4.0)

Description The diffUTR package provides a uniform interface and plotting functions for
limma/edgeR/DEXSeq -powered differential bin/exon usage. It includes in
addition an improved version of the limma::diffSplice method. Most
importantly, diffUTR further extends the application of these frameworks to
differential UTR usage analysis using poly-A site databases.

Imports S4Vectors, SummarizedExperiment, limma, edgeR, DEXSeq,
GenomicRanges, Rsубread, ggplot2, rtracklayer, ComplexHeatmap,
ggrepel, stringi, methods, stats, GenomeInfoDb, dplyr,
matrixStats, IRanges, ensembldb, viridisLite

Suggests BiocStyle, knitr, rmarkdown

biocViews GeneExpression

BugReports https://github.com/ETHZ-INS/diffUTR

VignetteBuilder knitr

License GPL-3

Encoding UTF-8

RoxygenNote 7.1.2

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

git_branch RELEASE_3_18

git_last_commit 90e9a90

git_last_commit_date 2023-10-24

Repository Bioconductor 3.18

Date/Publication 2024-04-10

Author Pierre-Luc Germain [cre, aut] (http://orcid.org/0000-0003-3418-4218),
Stefan Gerber [aut]

Maintainer Pierre-Luc Germain <pierre-luc.germain@hest.ethz.ch>
addNormalizedAssays

Description

addNormalizedAssays

Usage

addNormalizedAssays(se, readLength = 50L)

Arguments

se A bin-wise ‘SummarizedExperiment’ as produced by countFeatures
readLength Used as a minimum width to estimate read density (default 50).

Value

The ‘se’ object with populated ‘logcpm’ and ‘logNormDensity’ assays.

Examples

data(example_bin_se)
example_bin_se <- addNormalizedAssays(example_bin_se)
**countFeatures**

**Description**

countFeatures

**Usage**

countFeatures(
  bamfiles,
  bins,
  strandSpecific = 0,
  readLength = 50L,
  allowMultiOverlap = TRUE,
  inclNormalized = TRUE,
  tmpDir = tempdir(),
  ...
)

**Arguments**

- bamfiles: A vector of paths to bam files
- bins: A GRanges of bins in which to count reads (or path to a rds file containing such an object)
- strandSpecific: Passed to ‘Rsubread::featureCounts’
- readLength: Used as a minimum width to estimate read density.
- allowMultiOverlap: Passed to ‘Rsubread::featureCounts’
- inclNormalized: Logical; whether to include normalized assays (needed for plotting)
- tmpDir: Passed to ‘Rsubread::featureCounts’
- ...: Passed to ‘Rsubread::featureCounts’

**Value**

A RangedSummarizedExperiment-class

**Examples**

data("example_gene_annotation", package="diffUTR")
bins <- prepareBins(example_gene_annotation)
bam_files <- list.files(system.file("extdata", package="diffUTR"),
  pattern="bam$", full=TRUE)

# not run
# se <- countFeatures(bam_files, bins, verbose=FALSE)
deuBinPlot

Description

deuBinPlot

Usage

deuBinPlot(
  se,
  gene,
  type = c("summary", "condition", "sample"),
  intronSize = 2,
  exonSize = c("sqrt", "linear", "log"),
  y = NULL,
  condition = NULL,
  size = "type",
  lineSize = 1,
  colour = NULL,
  alpha = NULL,
  removeAmbiguous = TRUE,
  minDensityRatio = 0.1
)

Arguments

  se            A bin-wise SummarizedExperiment as produced by countFeatures and including bin-level tests (i.e. having been passed through one of the DEU wrappers such as diffSpliceWrapper or DEXSeqWrapper)
  gene          The gene of interest
  type          Either 'summary' (plot DEU summary), 'sample' (plot sample-wise data), or 'condition' (plot data aggregate by condition)
  intronSize    Intron plot size. If <=3, intron size will be this fraction of the mean exon size. If >3, each intron will have the given size.
  exonSize      Scaling for exon sizes, either 'sqrt', 'log', or 'linear'.
  y             Value to plot on the y-axis. If 'type="summary"', this should be a column of rowData(se), otherwise should be an assay name of 'se'.
  condition     The colData column containing the samples’ condition.
  size          rowData variable to use to determine the thickness of the bins.
  lineSize      Size of the line connecting the bins. Use ‘lineSize=0’ to omit the line.
  colour        rowData variable to use to determine the colour of the bins. If 'type="condition"', can also be "condition"; if 'type="sample"' can be any colData column.
  alpha         Alpha level, passed to ggplot.
diffSplice2

removeAmbiguous
Logical; whether to remove bins that are gene-ambiguous (i.e. overlap multiple genes).

minDensityRatio
Minimum ratio of read density (with respect to the gene’s average) for a bin to be plotted.

Value
A ggplot object

Examples

data(example_bin_se)
se <- diffSpliceWrapper(example_bin_se, ~condition)
deuBinPlot(se, "Jund")

Description
This is a small improvement to the diffSplice function written by Gordon Smyth and Charity Law.

Usage
diffSplice2(fit, geneid, exonid = NULL, robust = FALSE, verbose = TRUE)

Arguments
fit an MArrayLM-class fitted model object produced by lmFit or 'contrasts.fit', with rows corresponding to exons.
geneid gene identifiers (as in diffSplice)
exonid exon identifiers (as in diffSplice)
robust logical, should the estimation of the empirical Bayes prior parameters be robustified against outlier sample variances?
verbose logical, if TRUE will output some diagnostic information

Value
An MArrayLM-class object containing both exon level and gene level tests. Results are sorted by geneid and by exonid within gene.
Examples

```r
library(SummarizedExperiment)
library(edgeR)
data(example_bin_se)
se <- example_bin_se
design <- model.matrix(~condition, data=as.data.frame(colData(se)))
dds <- calcNormFactors(DGEList(assays(se)$counts))
dds <- voom(dds, design)
dds <- lmFit(dds, design)
res <- diffSplice2(dds, geneid=rowData(se)$gene, exonid=row.names(se))
topSplice(res)
```

---

diffSpliceDGEWrapper  \textit{DEUwrappers}

Description

Wrappers around commonly-used DEU methods (\texttt{diffSpliceDGE}, \texttt{DEXSeq} and an improved version of \texttt{diffSplice})

Usage

```r
diffSpliceDGEWrapper(
  se,
  design,
  coef = NULL,
  QLF = TRUE,
  robust = TRUE,
  countFilter = TRUE,
  excludeTypes = NULL
)

diffSpliceWrapper(
  se,
  design,
  coef = NULL,
  robust = TRUE,
  improved = TRUE,
  countFilter = TRUE,
  excludeTypes = NULL
)

DEXSeqWrapper(
  se,
  design = ~sample + exon + condition:exon,
  reducedModel = ~sample + exon,
  excludeTypes = NULL,
)```
Arguments

se  A bin-wise SummarizedExperiment as produced by countFeatures

design  A formula (using columns of ‘colData(se)’) or (for ‘diffSpliceWrapper’ or ‘diffSpliceDGEWrapper’ only) a model.matrix.

coef  The coefficient to be tested (ignored for ‘DEXSeqWrapper’).

QLF  Logical; whether to use edgeR’s quasi-likelihood negative binomial (applicable only to ‘diffSpliceDGEWrapper’).

robust  Logical; whether to use robust fitting for the dispersion trend (ignored for ‘DEXSeqWrapper’).

countFilter  Logical; whether to filter out low-count bins (ignored for ‘DEXSeqWrapper’).

excludeTypes  A vector of bin types to ignore for testing. To test for any kind of differential usage, leave empty. To test for differential UTR usage, use ‘excludeTypes=c("CDS","non-coding")’ (or see geneLevelStats for more options).

improved  Logical; whether to use diffSplice2 instead of the original diffSplice (default TRUE).

reducedModel  A reduced formula (applicable only to ‘DEXSeqWrapper’).

...  Further arguments (passed to ‘testForDEU’ and ‘estimateExonFoldChanges’) of ‘DEXSeq’. Can for instance be used to enable multithreading, by passing ‘BPPARAM=BiocParallel::MulticoreParam(ncores)’.

Value

The ‘se’ object with additional rowData columns contain bin (i.e. exon) -level statistics, and a metadata slot containing gene level p-values.

Examples

library(SummarizedExperiment)
data(example_bin_se)
se <- diffSpliceWrapper(example_bin_se, ~condition)
head(rowData(se))

example_bin_se  Example bin-level ‘RangedSummarizedExperiment’

Description

An object produced by countFeatures containing small subset of genes from mouse hippocampal slices undergoing Forskolin-induced long-term potentiation (GSE84643).
geneBinHeatmap

Value

A `RangedSummarizedExperiment`

References

https://www.nature.com/articles/s41598-017-17407-w

eexample_gene_annotation

Example gene annotation

Description

An example gene annotation containing only a small subset of mouse genes.

Value

A `GRanges` object

geneBinHeatmap

description

A wrapper around `ComplexHeatmap`.

Usage

geneBinHeatmap(
  se, gene, what = NULL, anno_rows = c("type", "logWidth", "meanLogDensity", "log10PValue", "geneAmbiguous"), anno_columns = c(), anno_colors = list(), removeAmbiguous = FALSE, merge_legends = TRUE, cluster_columns = FALSE, minDensityRatio = 0.1, left_annotation = NULL, top_annotation = NULL, ...
)
geneLevelStats

Arguments

se            A bin-wise SummarizedExperiment as produced by countFeatures
gene          The gene of interest
what          Type of values (i.e. assay) to plot
anno_rows     Row annotation columns (i.e. columns of ‘rowData(se)’) to plot
anno_columns  Column annotation columns (i.e. columns of ‘colData(se)’) to plot
anno_colors   Annotation colors, as a list named with the row/column annotations, see 'SingleAnnotation'
               for details. Ignored if ‘left_annotation’ and/or ‘top_annotation’ are given di-
               rectly.
removeAmbiguous Logical; whether to remove bins that are gene-ambiguous (i.e. overlap multiple
genes).
merge_legends  Logical; whether to merge legends. This effectively calls ‘draw(..., merge_legends=TRUE)’
               around the heatmap.
cluster_columns Logical; whether to cluster columns (passed to Heatmap)
minDensityRatio Minimum ratio of read density (with respect to the gene’s average) for a bin to
                  be plotted.
left_annotation Passed to Heatmap, overrides ‘anno_rows’.
top_annotation  Passed to Heatmap, overrides ‘anno_columns’.
...            Passed to ‘ComplexHeatmap’ (see Heatmap)

Value

A Heatmap

Examples

    data(example_bin_se)
    se <- diffSpliceWrapper(example_bin_se, ~condition)
    geneBinHeatmap(se, "Jund")

---

geneLevelStats     geneLevelStats

Description

Aggregates bin-level statistics to the gene-level
geneLevelStats

Usage

geneLevelStats(
  se,
  coef = NULL,
  excludeTypes = NULL,
  includeTypes = NULL,
  returnSE = TRUE,
  minDensityRatio = 0.1,
  minWidth = 20,
  excludeGeneAmbiguous = TRUE
)

Arguments

se  A ‘RangedSummarizedExperiment’ containing the results of one of the DEU wrappers.
coef  The coefficients tested (if the model included more than one term).
excludeTypes  Vector of bin types to exclude.
includeTypes  Vector of bin types to include (overrides ‘excludeTypes’)
returnSE  Logical; whether to return the updated ‘se’ object (default), or the gene-level table.
minDensityRatio  Minimum ratio of read density (with respect to the gene’s average) for a bin to be included.
minWidth  Minimum bin width to include
excludeGeneAmbiguous  Logical; whether to exclude bins which are ambiguous (i.e. can be from different genes)

Value

If ‘returnSE=TRUE’ (default), returns the ‘se’ object with an updated ‘metadata(se)$geneLevel’ slot, otherwise returns the gene-level data.frame.

Examples

library(SummarizedExperiment)
data(example_bin_se)
se <- diffSpliceWrapper(example_bin_se, ~condition)
se <- geneLevelStats(se, includeTypes="3UTR")
head(metadata(se)$geneLevel)
plotTopGenes

**Description**

plotTopGenes

**Usage**

```r
plotTopGenes(se, n = 10, FDR = 0.05, diffUTR = FALSE, alpha = 1, ...)
```

**Arguments**

- `se` A bin-wise SummarizedExperiment as produced by `countFeatures` and including bin-level tests (i.e. having been passed through one of the DEU wrappers such as `diffSpliceWrapper` or `DEXSeqWrapper`)
- `n` The maximum number of genes for which to plot labels
- `FDR` The FDR threshold above which to plot labels
- `diffUTR` Logical; if FALSE, uses absolute coefficients (appropriate for normal differential exon usage); if TRUE, uses non-absolute (ie changes should be in the same direction across significant bins) and width-weighted scores (i.e. larger bins have more weight) – this is relevant only when testing UTR usage.
- `alpha` Points transparency
- `...` Passed to `geom_label_repel`; this can for instance be used to increase ‘max.overlaps’ when not all desired gene labels are displayed)

**Value**

A ggplot

**Examples**

```r
data(example_bin_se)
se <- diffSpliceWrapper(example_bin_se, ~condition)
plotTopGenes(se)
```
prepareBins

Description

prepareBins

Usage

prepareBins(
  g,
  APA = NULL,
  onlyMainChr = TRUE,
  removeAntisense = TRUE,
  chrStyle = NULL,
  maxUTRbinSize = 15000,
  codingOnly = FALSE,
  genewise = FALSE,
  stranded = FALSE,
  verbose = TRUE
)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>A GRanges (or path to RDS file containing a GRanges) or path to a gtf file or EnsDb object containing the gene annotation.</td>
</tr>
<tr>
<td>APA</td>
<td>A GRanges (or path to a GRanges in RDS format) or bed file containing the alternative poly-A site database</td>
</tr>
<tr>
<td>onlyMainChr</td>
<td>Logical; whether to keep only main chromosomes</td>
</tr>
<tr>
<td>removeAntisense</td>
<td>Logical; whether to remove antisense APA sites</td>
</tr>
<tr>
<td>chrStyle</td>
<td>Logical; whether to remove antisense APA sites</td>
</tr>
<tr>
<td>maxUTRbinSize</td>
<td>Max width of new alternative UTR bins</td>
</tr>
<tr>
<td>codingOnly</td>
<td>Logical, whether to keep only coding transcripts</td>
</tr>
<tr>
<td>genewise</td>
<td>Logical, whether annotation should be flattened genewise</td>
</tr>
<tr>
<td>stranded</td>
<td>Logical, whether to perform disjoin in a stranded fashion.</td>
</tr>
<tr>
<td>verbose</td>
<td>Logical, whether to print run information</td>
</tr>
</tbody>
</table>

Details

See the vignette for more details.

Value

A ‘GRanges‘ object.
**Author(s)**

Stefan Greber

**Examples**

```r
data(example_gene_annotation)
bins <- prepareBins(example_gene_annotation)
```

---

**rn6_PAS**

*Poly-A sites compendium for Rattus Norvegicus (Rno6)*

**Description**

These are the sites from polyA_DB release 3.2, downloaded from [https://exon.apps.wistar.org/PolyA_DB/v3/download/3.2/rat_pas.zip](https://exon.apps.wistar.org/PolyA_DB/v3/download/3.2/rat_pas.zip), and lifted over to Rno6.

**Value**

a `GRanges` object

---

**simesAggregation**

**Description**

Simes p-value correction and aggregation, adapted from [limma][diffSplice]

**Usage**

```r
simesAggregation(p.value, geneid)
```

**Arguments**

- `p.value`: A vector of p-values
- `geneid`: A vector of group labels such as gene identifiers

**Value**

A named vector of aggregated p-values

**Examples**

```r
p <- runif(50)
genes <- sample(LETTERS,50,replace=TRUE)
simesAggregation(p, genes)
```
Index

addNormalizedAssays, 2

countFeatures, 2, 3, 4, 7, 9, 11

deuBinPlot, 4
DEUwrappers (diffSpliceDGEWrapper), 6
DEXSeq, 6
DEXSeqWrapper, 4, 11
DEXSeqWrapper (diffSpliceDGEWrapper), 6
diffSplice, 5–7
diffSplice2, 5, 7
diffSpliceDGE, 6
diffSpliceDGEWrapper, 6
diffSpliceWrapper, 4, 11
diffSpliceWrapper (diffSpliceDGEWrapper), 6

example_bin_se, 7
example_gene_annotation, 8

geneBinHeatmap, 8
geneLevelStats, 7, 9
geneLevelStats, 7, 9
geneLevelStats, 7, 9
geneLevelStats, 7, 9
geom_label_repel, 11

Heatmap, 9

lmFit, 5

plotTopGenes, 11
prepareBins, 12

RangedSummarizedExperiment-class, 3
rn6_PAS, 13

simesAggregation, 13
SingleAnnotation, 9