

# Package ‘diffUTR’

May 18, 2026

**Type** Package

**Title** diffUTR: Streamlining differential exon and 3' UTR usage

**Version** 1.21.0

**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, Rsubread, 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** devel

**git\_last\_commit** 412a54c

**git\_last\_commit\_date** 2026-04-28

**Repository** Bioconductor 3.24

**Date/Publication** 2026-05-18

**Author** Pierre-Luc Germain [cre, aut] (ORCID:  
<<https://orcid.org/0000-0003-3418-4218>>),  
Stefan Gerber [aut]

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

## Contents

addNormalizedAssays . . . . .	2
countFeatures . . . . .	3
deuBinPlot . . . . .	4
diffSplice2 . . . . .	5
diffSpliceDGEWrapper . . . . .	6
example_bin_se . . . . .	7
example_gene_annotation . . . . .	7
geneBinHeatmap . . . . .	8
geneLevelStats . . . . .	9
plotTopGenes . . . . .	10
prepareBins . . . . .	11
rn6_PAS . . . . .	12
simesAggregation . . . . .	12

<b>Index</b>	<b>13</b>
--------------	-----------

---

addNormalizedAssays	<i>addNormalizedAssays</i>
---------------------	----------------------------

---

### Description

addNormalizedAssays

### Usage

```
addNormalizedAssays(se, readLength = 50L)
```

### Arguments

se	A bin-wise ‘SummarizedExperiment’ as produced by <code>countFeatures</code>
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	<i>countFeatures</i>
---------------	----------------------

---

## 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

*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 <a href="#">countFeatures</a> and including bin-level tests (i.e. having been passed through one of the DEU wrappers such as <a href="#">diffSpliceWrapper</a> or <a href="#">DEXSeqWrapper</a> )
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 $\leq 3$ , 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.
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")
```

---

diffSplice2

*diffSplice2*


---

**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 <a href="#">MArrayLM-class</a> fitted model object produced by <a href="#">lmFit</a> or 'contrasts.fit', with rows corresponding to exons.
geneid	gene identifiers (as in <a href="#">diffSplice</a> )
exonid	exon identifiers (as in <a href="#">diffSplice</a> )
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**

```
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 *DEUwrappers*


---

## Description

Wrappers around commonly-used DEU methods ([diffSpliceDGE](#), [DEXSeq](#) and an improved version of [diffSplice](#))

## Usage

```
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 <a href="#">countFeatures</a>
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 <a href="#">geneLevelStats</a> for more options).
improved	Logical; whether to use <a href="#">diffSplice2</a> instead of the original <a href="#">diffSplice</a> (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	<i>Example bin-level 'RangedSummarizedExperiment'</i>
----------------	---

---

**Description**

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

**Value**

a 'RangedSummarizedExperiment'

**References**

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

---

example_gene_annotation	<i>Example gene annotation</i>
-------------------------	--------------------------------

---

**Description**

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

**Value**

a 'GRanges' object

---

geneBinHeatmap                      *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,
  ...
)
```

## Arguments

se	A bin-wise SummarizedExperiment as produced by <a href="#">countFeatures</a>
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 ‘ <a href="#">SingleAnnotation</a> ’ for details. Ignored if ‘left_annotation’ and/or ‘top_annotation’ are given directly.
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 <a href="#">Heatmap</a> )
minDensityRatio	Minimum ratio of read density (with respect to the gene’s average) for a bin to be plotted.
left_annotation	Passed to <a href="#">Heatmap</a> , overrides ‘anno_rows’.
top_annotation	Passed to <a href="#">Heatmap</a> , overrides ‘anno_columns’.
...	Passed to ‘ComplexHeatmap’ (see <a href="#">Heatmap</a> )

**Value**

A [Heatmap](#)

**Examples**

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

---

<code>geneLevelStats</code>	<i>geneLevelStats</i>
-----------------------------	-----------------------

---

**Description**

Aggregates bin-level statistics to the gene-level

**Usage**

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

**Arguments**

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

*plotTopGenes*


---

**Description**

plotTopGenes

**Usage**

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

**Arguments**

se	A bin-wise SummarizedExperiment as produced by <a href="#">countFeatures</a> and including bin-level tests (i.e. having been passed through one of the DEU wrappers such as <a href="#">diffSpliceWrapper</a> or <a href="#">DEXSeqWrapper</a> )
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 <a href="#">geom_label_repel</a> ; this can for instance be used to increase ‘max.overlaps’ when not all desired gene labels are displayed

**Value**

A ggplot

**Examples**

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

---

```
prepareBins      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**

<code>g</code>	A GRanges (or path to RDS file containing a GRanges) or path to a gtf file or EnsDb object containing the gene annotation.
<code>APA</code>	A GRanges (or path to a GRanges in RDS format) or bed file containing the alternative poly-A site database
<code>onlyMainChr</code>	Logical; whether to keep only main chromosomes
<code>removeAntisense</code>	Logical; whether to remove antisense APA sites
<code>chrStyle</code>	Chromosome notation to convert to (default no conversion)
<code>maxUTRbinSize</code>	Max width of new alternative UTR bins
<code>codingOnly</code>	Logical, whether to keep only coding transcripts
<code>genewise</code>	Logical, whether annotation should be flattened genewise
<code>stranded</code>	Logical, whether to perform disjoint in a stranded fashion.
<code>verbose</code>	Logical, whether to print run information

**Details**

See the vignette for more details.

**Value**

A ‘GRanges’ object.

**Author(s)**

Stefan Greber

**Examples**

```
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

*simesAggregation*


---

**Description**

Simes p-value correction and aggregation, adapted from `link[limma]{diffSplice}`

**Usage**

```
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**

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

# Index

`addNormalizedAssays`, 2

`countFeatures`, 2, 3, 4, 6–8, 10

`deuBinPlot`, 4

`DEUwrappers (diffSpliceDGEWrapper)`, 6

`DEXSeq`, 6

`DEXSeqWrapper`, 4, 10

`DEXSeqWrapper (diffSpliceDGEWrapper)`, 6

`diffSplice`, 5–7

`diffSplice2`, 5, 7

`diffSpliceDGE`, 6

`diffSpliceDGEWrapper`, 6

`diffSpliceWrapper`, 4, 10

`diffSpliceWrapper (diffSpliceDGEWrapper)`, 6

`example_bin_se`, 7

`example_gene_annotation`, 7

`geneBinHeatmap`, 8

`geneLevelStats`, 7, 9

`geom_label_repel`, 10

`Heatmap`, 8, 9

`lmFit`, 5

`plotTopGenes`, 10

`prepareBins`, 11

`RangedSummarizedExperiment-class`, 3

`rn6_PAS`, 12

`simesAggregation`, 12

`SingleAnnotation`, 8