

# Package ‘txcutr’

December 14, 2024

**Title** Transcriptome CUTteR

**Description** Various mRNA sequencing library preparation methods generate sequencing reads specifically from the transcript ends. Analyses that focus on quantification of isoform usage from such data can be aided by using truncated versions of transcriptome annotations, both at the alignment or pseudo-alignment stage, as well as in downstream analysis. This package implements some convenience methods for readily generating such truncated annotations and their corresponding sequences.

**Version** 1.13.0

**Date** 2024-04-18

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 4.1.0)

**Imports** AnnotationDbi, GenomicFeatures, txdbmaker, IRanges, GenomicRanges, BiocGenerics, Biostrings, S4Vectors, rtracklayer, BiocParallel, stats, methods, utils

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**biocViews** Alignment, Annotation, RNASeq, Sequencing, Transcriptomics

**Suggests** RefManageR, BiocStyle, knitr, sessioninfo, rmarkdown, testthat (>= 3.0.0), TxDb.Scerevisiae.UCSC.sacCer3.sgdGene, BSgenome.Scerevisiae.UCSC.sacCer3

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/txcutr>

**git\_branch** devel

**git\_last\_commit** 68a3c9e

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2024-12-13

**Author** Mervin Fansler [aut, cre] (ORCID:  
<https://orcid.org/0000-0002-4108-4218>>)

**Maintainer** Mervin Fansler <fanslerm@mskcc.org>

## Contents

<code>.clipTranscript</code> . . . . .	2
<code>.fillReduce</code> . . . . .	3
<code>.mutateEach,CompressedGRangesList-method</code> . . . . .	3
<code>.propagateMap</code> . . . . .	4
<code>exportFASTA</code> . . . . .	4
<code>exportGTF</code> . . . . .	5
<code>exportMergeTable</code> . . . . .	6
<code>generateMergeTable</code> . . . . .	7
<code>truncateTxome</code> . . . . .	8
<code>txdbToGRangesList</code> . . . . .	9
<b>Index</b>	<b>11</b>

---

<code>.clipTranscript</code>	<i>Clip Transcript to Given Length</i>
------------------------------	--

---

### Description

Internal function for operating on individual GRanges, where ranges represent exons in a transcript. This is designed to be used in an `*apply` function over a GRangesList object.

### Usage

```
.clipTranscript(gr, maxTxLength)
```

### Arguments

<code>gr</code>	a GRanges object
<code>maxTxLength</code>	a positive integer

### Value

the clipped GRanges object

---

.fillReduce                      *Convert GRanges to Single Range*

---

**Description**

Convert GRanges to Single Range

**Usage**

```
.fillReduce(gr, validate = TRUE)
```

**Arguments**

gr	a GRanges with ranges to be merged.
validate	logical determining whether entries should be checked for compatible seqnames and strands.

**Details**

The validation assumes seqnames and strand are Rle objects.

**Value**

GRanges with single interval

---

.mutateEach, CompressedGRangesList-method  
*Efficient Metadata Columns Mutation*

---

**Description**

Efficient Metadata Columns Mutation

**Usage**

```
## S4 method for signature 'CompressedGRangesList'  
.mutateEach(grl, ...)
```

**Arguments**

grl	a CompressedGRangesList
...	named list of vectors to insert as metadata columns on each element GRanges. Each vector length must match the length of the GRangesList.

**Value**

a CompressedGRangesList with all element GRanges updated with supplied metadata columns

---

<code>.propagateMap</code>	<i>Propagate Transcript Merge Map</i>
----------------------------	---------------------------------------

---

**Description**

Propagate Transcript Merge Map

**Usage**

```
.propagateMap(df, MAXITERS = 1000)
```

**Arguments**

<code>df</code>	a <code>data.frame</code> with columns <code>tx_in</code> and <code>tx_out</code>
<code>MAXITERS</code>	a numeric controlling the maximum number of iterations

**Value**

a converged `data.frame`, such that, `tx_out` is not present in any `tx_in`

---

<code>exportFASTA</code>	<i>Export Transcriptome as FASTA</i>
--------------------------	--------------------------------------

---

**Description**

Export Transcriptome as FASTA

**Usage**

```
exportFASTA(txdb, genome, file, ...)
```

**Arguments**

<code>txdb</code>	a <code>TxDb</code> object representing a transcriptome annotation
<code>genome</code>	a <code>BSgenome</code> object from which to extract sequences
<code>file</code>	a string for output FASTA file. File names ending in <code>".gz"</code> will automatically use gzip compression.
<code>...</code>	additional arguments to pass through to <a href="#">writeXStringSet</a>

**Value**

The `txdb` argument is invisibly returned.

## Examples

```
library(TxDb.Scerevisiae.UCSC.sacCer3.sgdGene)
library(BSgenome.Scerevisiae.UCSC.sacCer3)

## load annotation and genome
txdb <- TxDb.Scerevisiae.UCSC.sacCer3.sgdGene
sacCer3 <- BSgenome.Scerevisiae.UCSC.sacCer3

## restrict to 'chrI' transcripts (makes for briefer example runtime)
seqlevels(txdb) <- c("chrI")

## last 500 nts per tx
txdb_w500 <- truncateTxome(txdb)

## export uncompressed
outfile <- tempfile("sacCer3.sgdGene.w500", fileext=".fa")
exportFASTA(txdb_w500, sacCer3, outfile)

## export compressed
outfile <- tempfile("sacCer3.sgdGene.w500", fileext=".fa.gz")
exportFASTA(txdb_w500, sacCer3, outfile)
```

---

exportGTF

*Export GTF*

---

## Description

Exports a TxDb annotation to a GTF file

## Usage

```
exportGTF(txdb, file, source = "txcutr")
```

## Arguments

txdb	transcriptome to be output
file	a string or <a href="#">connection</a> to output GTF file. Automatically recognizes strings ending with ".gz" for zipped output.
source	a string to go in the source column

## Value

The txdb argument is invisibly returned.

## Examples

```
library(Txdb.Scerevisiae.UCSC.sacCer3.sgdGene)

## load annotation
txdb <- Txdb.Scerevisiae.UCSC.sacCer3.sgdGene

## restrict to 'chrI' transcripts
seqlevels(txdb) <- c("chrI")

## last 500 nts per tx
txdb_w500 <- truncateTxome(txdb)

## export uncompressed
outfile <- tempfile("sacCer3.sgdGene.w500", fileext=".gtf")
exportGTF(txdb_w500, outfile)

## export compressed
outfile <- tempfile("sacCer3.sgdGene.w500", fileext=".gtf.gz")
exportGTF(txdb_w500, outfile)
```

---

exportMergeTable	<i>Export Merge Table for Transcriptome</i>
------------------	---

---

## Description

Export Merge Table for Transcriptome

## Usage

```
exportMergeTable(txdb, file, minDistance = 200L)
```

## Arguments

txdb	a TxDb object representing a transcriptome annotation
file	a string or <a href="#">connection</a> to output TSV file. Automatically recognizes strings ending with <b>".gz"</b> for zipped output.
minDistance	the minimum separation to regard overlapping transcripts as unique.

## Value

The txdb argument is invisibly returned.

## Examples

```
library(TxDB.Scerevisiae.UCSC.sacCer3.sgdGene)

## load annotation
txdb <- TxDb.Scerevisiae.UCSC.sacCer3.sgdGene

## restrict to 'chrI' transcripts (makes for briefer example runtime)
seqlevels(txdb) <- c("chrI")

## last 500 nts per tx
txdb_w500 <- truncateTxome(txdb)

## export plain format
outfile <- tempfile("sacCer3.sgdGene.w500", fileext=".tsv")
exportMergeTable(txdb_w500, outfile)

## export compressed format
outfile <- tempfile("sacCer3.sgdGene.w500", fileext=".tsv.gz")
exportMergeTable(txdb_w500, outfile)
```

---

generateMergeTable	<i>Generate Merge Table</i>
--------------------	-----------------------------

---

## Description

Generate Merge Table

## Usage

```
generateMergeTable(txdb, minDistance = 200)

## S4 method for signature 'TxDb'
generateMergeTable(txdb, minDistance = 200L)
```

## Arguments

txdb	an object representing a transcriptome
minDistance	the minimum separation to regard overlapping transcripts as unique

## Value

a data.frame with three columns - tx\_in the input transcript - tx\_out the transcript merged into - gene\_out the gene merged into

a data.frame with three columns - tx\_in the input transcript - tx\_out the transcript merged into - gene\_out the gene merged into

**Examples**

```

library(TxDb.Scerevisiae.UCSC.sacCer3.sgdGene)

## load annotation
txdb <- TxDb.Scerevisiae.UCSC.sacCer3.sgdGene

## restrict to 'chrI' transcripts
seqlevels(txdb) <- c("chrI")

## last 500 nts per tx
txdb_w500 <- truncateTxome(txdb)
txdb_w500

## last 100 nts per tx
txdb_w100 <- truncateTxome(txdb, maxTxLength=100)
txdb_w100

```

---

truncateTxome

*Truncate Transcriptome*


---

**Description**

Truncate Transcriptome

**Usage**

```

truncateTxome(txdb, maxTxLength = 500, ...)

## S4 method for signature 'TxDb'
truncateTxome(txdb, maxTxLength = 500, BPPARAM = bpparam())

```

**Arguments**

txdb	a TxDb object
maxTxLength	the maximum length of transcripts
...	additional arguments
BPPARAM	A <a href="#">BiocParallelParam</a> object specifying whether and how the method should be parallelized.

**Value**

a TxDb object  
a TxDb object



**Examples**

```
library(TxDb.Scerevisiae.UCSC.sacCer3.sgdGene)

## load annotation
txdb <- TxDb.Scerevisiae.UCSC.sacCer3.sgdGene

## restrict to 'chrI' transcripts
seqlevels(txdb) <- c("chrI")

## last 500 nts per tx
txdb_w500 <- truncateTxome(txdb)
txdb_w500

## last 100 nts per tx
txdb_w100 <- truncateTxome(txdb, maxTxLength=100)
txdb_w100
```

---

txdbToGRangesList      *Convert TxDb object to GRangesList*

---

**Description**

Convert TxDb object to GRangesList

**Usage**

```
txdbToGRangesList(
  txdb,
  geneCols = c("gene_id"),
  transcriptCols = c("gene_id", "tx_name"),
  exonCols = c("gene_id", "tx_name", "exon_id", "exon_rank")
)
```

**Arguments**

txdb                    a TxDb object

geneCols                names of columns to include in the genes ranges

transcriptCols        names of columns to include in the transcripts ranges

exonCols                names of columns to include in the exons ranges

**Value**

a GRangesList object with entries c(genes, transcripts, exons)

**Examples**

```
library(TxDb.Scerevisiae.UCSC.sacCer3.sgdGene)

## load annotation
txdb <- TxDb.Scerevisiae.UCSC.sacCer3.sgdGene

grl <- txdbToGRangesList(txdb)
grl
```

# Index

.clipTranscript, [2](#)  
.fillReduce, [3](#)  
.mutateEach, CompressedGRangesList-method,  
[3](#)  
.propagateMap, [4](#)  
BiocParallelParam, [8](#)  
connection, [5](#), [6](#)  
exportFASTA, [4](#)  
exportGTF, [5](#)  
exportMergeTable, [6](#)  
generateMergeTable, [7](#)  
generateMergeTable, TxDb-method  
(generateMergeTable), [7](#)  
truncateTxome, [8](#)  
truncateTxome, TxDb-method  
(truncateTxome), [8](#)  
txdbToGRangesList, [9](#)  
writeXStringSet, [4](#)