Package 'txcutr'

May 9, 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 1.11.0

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 4e59583

git_last_commit_date 2024-04-30

Repository Bioconductor 3.20

Date/Publication 2024-05-08

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

Maintainer Mervin Fansler <fanslerm@mskcc.org>

Contents

`.clipTranscript` *Clip Transcript to Given Length*

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

`gr`	a `GRanges` object
`maxTxLength`	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

.propagateMap *Propagate Transcript Merge Map*

Description

Propagate Transcript Merge Map

Usage

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

Arguments

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

Value

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

exportFASTA *Export Transcriptome as FASTA*

Description

Export Transcriptome as FASTA

Usage

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

Arguments

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

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 connection 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` *Export Merge Table for Transcriptome*

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 `connection` to output TSV file. Automatically recognizes strings ending with "**.gz**" 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 *Generate Merge Table*

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 BiocParallelParam 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