# Package 'tximeta'

April 15, 2020

**Version** 1.4.5

**Title** Transcript Quantification Import with Automatic Metadata

**Description** Transcript quantification import from Salmon with automatic population of metadata and transcript ranges. Filtered, combined, or de novo transcriptomes can be linked to the appropriate sources with linkedTxomes and shared for reproducible analyses.

**Maintainer** Michael Love <michaelisaiahlove@gmail.com>

**License** GPL-2

**VignetteBuilder** knitr

**Imports** SummarizedExperiment, tximport, jsonlite, S4Vectors, GenomicRanges, AnnotationDbi, GenomicFeatures, ensembldb, Biostrings, BiocFileCache, tibble, GenomeInfoDb, rappdirs, utils, methods

**Suggests** knitr, rmarkdown, testthat, tximportData, org.Dm.eg.db, DESeq2, edgeR, limma, devtools

**URL** https://github.com/mikelove/tximeta

**biocViews** Annotation, DataImport, Preprocessing, RNASeq, Transcriptomics, Transcription, GeneExpression, ImmunoOncology

**RoxygenNote** 7.0.2

**Encoding** UTF-8

**git_url** https://git.bioconductor.org/packages/tximeta

**git_branch** RELEASE_3_10

**git_last_commit** 6d7e03f

**git_last_commit_date** 2020-03-10

**Date/Publication** 2020-04-14

**Author** Michael Love [aut, cre],
Rob Patro [aut, ctb],
Peter Hickey [aut, ctb],
Charlotte Soneson [aut, ctb]

# R topics documented:



---

`addExons` *Add exons to rowRanges of a transcript-level SummarizedExperiment*

---

## Description

After running `tximeta`, the `SummarizedExperiment` output will have `GRanges` representing the transcript locations attached as `rowRanges` to the object. These provide the start and end of the transcript in the genomic coordiantes, and strand information. However, the exonic locations are not provided. This function, `addExons`, swaps out the `GRanges` with a `GRangesList`, essentially a list along the rows of the `SummarizedExperiment`, where each element of the list is a `GRanges` providing the locations of the exons for that transcript.

## Usage

```
addExons(se)
```

## Arguments

| | |
|---|---|
| `se` | the SummarizedExperiment |

## Details

This function is designed only for transcript-level objects. This "lack of a feature" reflects a belief on the part of the package author that it makes more sense to think about exons belonging to transcripts than to genes. For users desiring exonic information alongside gene-level objects, for example, which exons are associated with a particular gene, it is recommended to pull out the relevant `GRangesList` for the transcripts of this gene, while the object represents transcript-level data, such that the exons are still associated with transcripts.

For an example of `addExons`, please see the tximeta vignette.

## Value

a SummarizedExperiment

## addIds — *Add IDs to rowRanges of a SummarizedExperiment*

### Description

For now this just works with SummarizedExperiments with Ensembl gene or transcript IDs. See example of usage in tximeta vignette. For obtaining multiple matching IDs for each row of the SummarizedExperiment set `multiVals="list"`. See `select` for documentation on use of `multiVals`.

### Usage

```
addIds(se, column, gene = FALSE, ...)
```

### Arguments

| | |
|---|---|
| `se` | the SummarizedExperiment |
| `column` | the name of the new ID to add (a `column` of the org database) |
| `gene` | logical, whether to map by genes or transcripts (default is FALSE). if rows are genes, and easily detected as such (ENSG or ENSMUSG), it will automatically switch to TRUE. if rows are transcripts and `gene=TRUE`, then it will try to use a `gene_id` column to map IDs to `column` |
| `...` | arguments passed to `mapIds` |

### Value

a SummarizedExperiment

### Examples

```
example(tximeta)
library(org.Dm.eg.db)
se <- addIds(se, "REFSEQ", gene=FALSE)
```

## getTximetaBFC — *Get or set the directory of the BiocFileCache used by tximeta*

### Description

Running `getTximetaBFC` will report the saved directory, if it has been determined, or will return NULL. Running `setTximetaBFC` will ask the user to specify a BiocFileCache directory for accessing and saving TxDb sqlite files.

### Usage

```
getTximetaBFC()

setTximetaBFC(dir)
```

### Arguments

| | |
|---|---|
| dir | the location for tximeta's BiocFileCache. can be missing in which case the function will call `file.choose` for choosing location interactively |

### Value

the directory of the BiocFileCache used by tximeta (or nothing, in the case of `setTximetaBFC`)

### Examples

```
# getting the BiocFileCache used by tximeta
# (may not be set, which uses BiocFileCache default or temp directory)
getTximetaBFC()

# don't want to actually change user settings so this is not run:
# setTximetaBFC()
```

---

## linkedTxome *Make and load linked transcriptomes ("linkedTxome")*

---

### Description

For now, for details please see the vignette `inst/script/linked.Rmd`

### Usage

```
makeLinkedTxome(
  indexDir,
  source,
  organism,
  release,
  genome,
  fasta,
  gtf,
  write = TRUE,
  jsonFile
)

loadLinkedTxome(jsonFile)
```

### Arguments

| | |
|---|---|
| indexDir | the local path to the Salmon index |
| source | the source of transcriptome (e.g. "GENCODE", "Ensembl", "de-novo") |
| organism | organism (e.g. "Homo sapiens") |
| release | release number (e.g. "27") |
| genome | genome (e.g. "GRCh38", or "none") |

| | |
|---|---|
| fasta | location(s) for the FASTA transcript sequences (of which the transcripts used to build the index is equal or a subset). This can be a local path, or an HTTP or FTP URL |
| gtf | location for the GTF/GFF file (of which the transcripts used to build the index is equal or a subset). This can be a local path, or an HTTP or FTP URL While the `fasta` argument can take a vector of length greater than one (more than one FASTA file containing transcripts used in indexing), the `gtf` argument has to be a single GTF/GFF file. If transcripts were added to a standard set of reference transcripts (e.g. fusion genes, or pathogen transcripts), it is recommended that the tximeta user would manually add these to the GTF/GFF file, and post the modified GTF/GFF publicly, such as on Zenodo. This enables consistent annotation and downstream annotation tasks, such as by `summarizeToGene`. |
| write | logical, should a JSON file be written out which documents the transcriptome checksum and metadata? (default is TRUE) |
| jsonFile | the path to the json file for the linkedTxome |

## Value

nothing, the function is run for its side effects

## Examples

```
# point to a Salmon quantification file with an additional artificial transcript
dir <- system.file("extdata/salmon_dm", package="tximportData")
file <- file.path(dir, "SRR1197474.plus", "quant.sf")
coldata <- data.frame(files=file, names="SRR1197474", sample="1",
                      stringsAsFactors=FALSE)

# now point to the Salmon index itself to create a linkedTxome
# as the index will not match a known txome
indexDir <- file.path(dir, "Dm.BDGP6.22.98.plus_salmon-0.14.1")

# point to the source FASTA and GTF:
fastaFTP <- c("ftp://ftp.ensembl.org/pub/release-98/fasta/drosophila_melanogaster/cdna/Drosophila_melanogas
         "ftp://ftp.ensembl.org/pub/release-98/fasta/drosophila_melanogaster/ncrna/Drosophila_melanogaster
              "extra_transcript.fa.gz")
gtfPath <- file.path(dir, "Drosophila_melanogaster.BDGP6.22.98.plus.gtf.gz")

# now create a linkedTxome, linking the Salmon index to its FASTA and GTF sources
makeLinkedTxome(indexDir=indexDir, source="Ensembl", organism="Drosophila melanogaster",
               release="98", genome="BDGP6.22", fasta=fastaFTP, gtf=gtfPath, write=FALSE)

# to clear the entire linkedTxome table
# (don't run unless you want to clear this table!)
# bfcloc <- getTximetaBFC()
# bfc <- BiocFileCache(bfcloc)
# bfcremove(bfc, bfcquery(bfc, "linkedTxomeTbl")$rid)
```

summarizeToGene,SummarizedExperiment-method
*Summarize estimated quantitites to gene-level*

## Description

Summarizes abundances, counts, lengths, (and inferential replicates or variance) from transcript- to gene-level. This function operates on SummarizedExperiment objects, and will automatically access the relevant TxDb (by either finding it in the BiocFileCache or by building it from an ftp location). #' This function uses the tximport package to perform summarization, where a method is defined that works on simple lists.

## Usage

```
## S4 method for signature 'SummarizedExperiment'
summarizeToGene(object, varReduce = FALSE, ...)
```

## Arguments

| | |
|---|---|
| object | a SummarizedExperiment produced by tximeta |
| varReduce | whether to reduce per-sample inferential replicates information into a matrix of sample variances variance (default FALSE) |
| ... | arguments passed to tximport |

## Value

a SummarizedExperiment with summarized quantifications

## Examples

```
example(tximeta)
gse <- summarizeToGene(se)
```

tximeta *tximeta: Transcript quantification import with automatic metadata*

## Description

tximeta leverages the hashed checksum of the Salmon index, in addition to a number of core Bioconductor packages (GenomicFeatures, ensembldb, GenomeInfoDb, BiocFileCache) to automatically populate metadata for the user, without additional effort from the user. Note that tximeta requires that the entire output directory of Salmon/Alevin is present and unmodified in order to identify the provenance of the reference transcripts.

## Usage

```
tximeta(
  coldata,
  type = "salmon",
  txOut = TRUE,
  skipMeta = FALSE,
  skipSeqinfo = FALSE,
  cleanDuplicateTxps = FALSE,
  ...
)
```

## Arguments

| | |
|---|---|
| coldata | a data.frame with at least two columns (others will propogate to object):<br>• files - character, paths of quantification files<br>• names - character, sample names<br>if coldata is a vector, it is assumed to be the paths of quantification files and unique sample names are created |
| type | what quantifier was used (see tximport) |
| txOut | whether to output transcript-level data. tximeta is designed to have transcript-level output with Salmon, so default is TRUE, and it's recommended to use summarizeToGene following tximeta for gene-level summarization. For an Alevin file, tximeta will import the gene level counts ignoring this argument (Alevin produces only gene-level quantification). |
| skipMeta | whether to skip metadata generation (e.g. to avoid errors if not connected to internet). This calls tximport directly and so either txOut=TRUE or tx2gene should be specified. |
| skipSeqinfo | whether to skip the addition of Seqinfo, which requires an internet connection to download the relevant chromosome information table from UCSC |
| cleanDuplicateTxps | whether to try to clean duplicate transcripts (exact sequence duplicates) by replacing the transcript names that do not appear in the GTF with those that do appear in the GTF |
| ... | arguments passed to tximport |

## Details

Most of the code in tximeta works to add metadata and transcript ranges when the quantification was performed with Salmon. However, tximeta can be used with any quantification type that is supported by tximport, where it will return an non-ranged SummarizedExperiment.

tximeta performs a lookup of the hashed checksum of the index (stored in an auxilary information directory of the Salmon output) against a database of known transcriptomes, which lives within the tximeta package and is continually updated on Bioconductor's release schedule. In addition, tximeta performs a lookup of the checksum against a locally stored table of linkedTxome's (see link{makeLinkedTxome}). If tximeta detects a match, it will automatically populate, e.g. the transcript locations, the transcriptome release, the genome with correct chromosome lengths, etc. It allows for automatic and correct summarization of transcript-level quantifications to the gene-level via summarizeToGene without the need to manually build a tx2gene table.

tximeta on the first run will ask where the BiocFileCache for this package should be kept, either using a default location or a temporary directory. At any point, the user can specify a location

using setTximetaBFC and this choice will be saved for future sessions. Multiple users can point to the same BiocFileCache, such that transcript databases (TxDb) associated with certain Salmon indices and linkedTxomes can be accessed by different users without additional effort or time spent downloading/building the relevant TxDb.

In order to allow that multiple users can read and write to the same location, one should set the BiocFileCache directory to have group write permissions (g+w).

## Value

a SummarizedExperiment with metadata on the rowRanges. (if the hashed checksum in the Salmon or Sailfish index does not match any known transcriptomes, or any locally saved linkedTxome, tximeta will just return a non-ranged SummarizedExperiment)

## Examples

```
# point to a Salmon quantification file:
dir <- system.file("extdata/salmon_dm", package="tximportData")
files <- file.path(dir, "SRR1197474", "quant.sf")
coldata <- data.frame(files, names="SRR1197474", condition="A", stringsAsFactors=FALSE)

# normally we would just run the following which would download the appropriate metadata
# se <- tximeta(coldata)

# for this example, we instead point to a local path where the GTF can be found
# by making a linkedTxome:
indexDir <- file.path(dir, "Dm.BDGP6.22.98_salmon-0.14.1")
fastaFTP <- c("ftp://ftp.ensembl.org/pub/release-98/fasta/drosophila_melanogaster/cdna/Drosophila_melanogas
         "ftp://ftp.ensembl.org/pub/release-98/fasta/drosophila_melanogaster/ncrna/Drosophila_melanogaster
gtfPath <- file.path(dir, "Drosophila_melanogaster.BDGP6.22.98.gtf.gz")
makeLinkedTxome(indexDir=indexDir, source="Ensembl", organism="Drosophila melanogaster",
               release="98", genome="BDGP6.22", fasta=fastaFTP, gtf=gtfPath, write=FALSE)
se <- tximeta(coldata)

# to clear the entire linkedTxome table
# (don't run unless you want to clear this table!)
# bfcloc <- getTximetaBFC()
# bfc <- BiocFileCache(bfcloc)
# bfcremove(bfc, bfcquery(bfc, "linkedTxomeTbl")$rid)
```

# Index