Package 'tximeta'

October 16, 2019

Version 1.2.2

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, devtools

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

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

RoxygenNote 6.1.1

Encoding UTF-8

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

git_branch RELEASE_3_9

git_last_commit d25e8ce

git_last_commit_date 2019-07-06

Date/Publication 2019-10-15

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

R topics documented:

addIds 2
getTximetaBFC 3
linkedTxome 3
summarizeToGene,SummarizedExperiment-method 5
tximeta 5

Index **8**

`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 path to the Salmon or Sailfish index
source	the source of transcriptome (e.g. "Gencode" or "Ensembl")
organism	organism (e.g. "Homo sapiens")
release	release number (e.g. "27")
genome	genome (e.g. "GRCh38")
fasta	FTP location for the FASTA sequence (of which the index is a subset)
gtf	FTP location for the GTF file (of which the index is a subset)
write	should a JSON file be written out which documents the transcriptome signature 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 which combined two Ensembl FASTA files:
dir <- system.file("extdata/salmon_dm/SRR1197474", package="tximportData")
file <- file.path(dir, "quant.sf.gz")
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
dir <- system.file("extdata", package="tximeta")
indexDir <- file.path(dir, "Drosophila_melanogaster.BDGP6.v92_salmon_0.10.2")

# point to the source FASTA and GTF:
fastaFTP <- c("ftp://ftp.ensembl.org/pub/release-92/fasta/drosophila_melanogaster/cdna/Drosophila_melanogas
        "ftp://ftp.ensembl.org/pub/release-92/fasta/drosophila_melanogaster/ncrna/Drosophila_melanogaster.

# we comment this out for the example, and instead point to a local version
# usually one would point to the FTP source for the GTF file here
# gtfFTP <- "ftp://ftp.ensembl.org/pub/release-92/gtf/drosophila_melanogaster/Drosophila_melanogaster.BDGP6.

dir2 <- system.file("extdata/salmon_dm", package="tximportData")
gtfPath <- file.path(dir2,"Drosophila_melanogaster.BDGP6.92.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="92", genome="BDGP6", 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 hash signature of the Salmon or Sailfish 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 this package is in "beta" / under development.

Usage

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

Arguments

coldata	a data.frame with at least two columns (others will propogate to object): • files - character, paths of quantification files • names - character, sample names 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 or Sailfish, so default is TRUE, and it's recommended to use summarizeToGene following tximeta for gene-level summarization.
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.
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 or Sailfish. However, tximeta can be used with any quantification type that is supported by tximport, where it will return an un-ranged SummarizedExperiment.

tximeta checks the hash signature of the index against a database of known transcriptomes (this database under construction) or a locally stored linkedTxome (see link{makeLinkedTxome}), and then 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 or Sailfish 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 hash signature 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_cdna", "quant.sf.gz")
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:
dir <- system.file("extdata", package="tximeta")
indexDir <- file.path(dir, "Drosophila_melanogaster.BDGP6.cdna.v92_salmon_0.10.2")
fastaFTP <- "ftp://ftp.ensembl.org/pub/release-92/fasta/drosophila_melanogaster/cdna/Drosophila_melanogaste
dir2 <- system.file("extdata/salmon_dm", package="tximportData")
gtfPath <- file.path(dir2,"Drosophila_melanogaster.BDGP6.92.gtf.gz")
makeLinkedTxome(indexDir=indexDir, source="Ensembl", organism="Drosophila melanogaster",
                release="92", genome="BDGP6", 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

addIds, 2

getTximetaBFC, 3

linkedTxome, 3
loadLinkedTxome *(*linkedTxome*)*, 3

makeLinkedTxome *(*linkedTxome*)*, 3

setTximetaBFC, *6*
setTximetaBFC *(*getTximetaBFC*)*, 3
summarizeToGene, *6*
summarizeToGene,SummarizedExperiment-method, 5

tximeta, 5
tximport, *6*