Package ‘EpiTxDb’

May 2, 2024

Type Package

Title Storing and accessing epitranscriptomic information using the AnnotationDbi interface

Version 1.16.0

Date 2024-03-24

Description EpiTxDb facilitates the storage of epitranscriptomic information. More specifically, it can keep track of modification identity, position, the enzyme for introducing it on the RNA, a specifier which determines the position on the RNA to be modified and the literature references each modification is associated with.

License Artistic-2.0

Encoding UTF-8

LazyData false

biocViews Software, Epitranscriptomics

Depends R (>= 4.0), AnnotationDbi, Modstrings

Imports methods, utils, htt, xml2, curl, rex, GenomicFeatures, txdmker, GenomicRanges, GenomeInfoDb, BiocGenerics, BiocFileCache, S4Vectors, IRanges, RSQLite, DBI, Biostrings, tRNAdbImport

Suggests BiocStyle, knitr, rmarkdown, testthat, httpstest, AnnotationHub, ensemblb, ggplot2, EpiTxDb.Hs.hg38, BSgenome.Hsapiens.UCSC.hg38, BSgenome.Scerevisiae.UCSC.sacCer3, TxDb.Hsapiens.UCSC.hg38.knownGene


RoxygenNote 7.3.1

BugReports https://github.com/FelixErnst/EpiTxDb/issues
EpiTxDb-package

**EpiTxDb-package**

EpiTxDb: Storing and accessing epitranscriptomic information using the AnnotationDbi interface

**Description**

EpiTxDb facilitates the storage of epitranscriptomic information. More specifically, it can keep track of modification identity, position, the enzyme for introducing it on the RNA, a specifier which determines the position on the RNA to be modified and the literature references each modification is associated with.

**Author(s)**

Maintainer: Felix G.M. Ernst <felix.gm.ernst@outlook.com> (ORCID)
See Also

Useful links:

- [https://github.com/FelixErnst/EpiTxDb](https://github.com/FelixErnst/EpiTxDb)
- Report bugs at [https://github.com/FelixErnst/EpiTxDb/issues](https://github.com/FelixErnst/EpiTxDb/issues)

---

### Description

The EpiTxDb class is a `AnnotationDb` type container for storing Epitranscriptomic information.

The information are typically stored on a per transcript and not as genomic coordinates, but the EpiTxDb class is agnostic to this. In case of genomic coordinates `transcriptsBy` will return modifications per chromosome.

### Usage

```r
## S4 method for signature 'EpiTxDb'
organism(object)

## S4 method for signature 'EpiTxDb'
seqinfo(x)

## S4 method for signature 'EpiTxDb'
seqlevels(x)

## S4 method for signature 'EpiTxDb'
as.list(x)
```

### Arguments

- `x, object` a EpiTxDb object

### Value

For

- `organism()` and `seqlevels()` a character vector
- `seqinfo()` a `Seqinfo` object
- `as.list()` a list
See Also

- `makeEpiTxDbFromGRanges` for creating a `EpiTxDb` object from a `GRanges` object and its metadata columns
- `makeEpiTxDbFromRMBase` for creating a `EpiTxDb` object from RMBase online resources
- `makeEpiTxDbFromtRNAdb` for creating a `EpiTxDb` object from tRNAdb online resources
- `makeEpiTxDb` for creating a `EpiTxDb` object from data.frames
- `modifications`, `modificationsBy` for getting epitranscriptomic modification locations
- `select` for using the default interface of `AnnotationDb` objects.
- `shiftGenomicToTranscript` and `shiftTranscriptToGenomic` for transferring genomic to transcript coordinates and back again.

Examples

ebdb_file <- system.file("extdata", "EpiTxDb.Hs.hg38.snoRNAdb.sqlite", package="EpiTxDb")
epdb <- loadDb(ebdb_file)
epdb

# general methods
seqinfo(epdb) #
seqlevels(epdb) # easy access to all transcript names

---

**EpiTxDb-data**

**EpiTxDb internal data**

Description

EpiTxDb internal data

Usage

data(rmbase_data)

Format

data.frame
EpiTxDb-package

EpiTxDb - Storing and accessing epitranscriptomic information using the AnnotationDbi interface

Description

title

Author(s)

Felix G M Ernst [aut]

References


makeEpiTxDb

Creating a EpiTxDb from user supplied annotations as data.frames

Description

makeEpiTxDb is a low-level constructor for creating a EpiTxDb object from user supplied annotations.

This function typically will not be used by regular users.

Usage

makeEpiTxDb(  
  modifications,  
  reactions = NULL,  
  specifiers = NULL,  
  references = NULL,  
  metadata = NULL,  
  reassign.ids = FALSE)

)
Arguments

**modifications** A data.frame containing the following columns:

- **mod_id**: a unique integer value per modification.
- **mod_type**: the modification type as a character or factor value. Must be a value from `shortName(ModRNAString())`.
- **mod_name**: a character or factor name for the specific modification
- **mod_start**: the start position for the modification as integer value. Usually `mod_start = mod_end`
- **mod_end**: the end position for the modification as integer value. Usually `mod_start = mod_end`
- **mod_strand**: the strand information for the modification as a character or factor.
- **sn_id**: an integer value per unique sequence
- **sn_name**: a character or factor as sequence name, e.g. a chromosome or a transcript identifier like chr1.

The first six are mandatory, whereas one of the last two has to be set. `sn_id` will be generated from `sn_name`, if `sn_id` is not set.

**reactions** An optional data.frame containing the following columns:

- **mod_id**: a integer value per modification and the link to the modification data.frame.
- **rx_genename**: a character or factor referencing a genename for the enzyme incorporating the modification.
- **rx_rank**: a integer for sorting enzyme reactions, if multiple enzymes are involved in the modification's incorporation/maintenance.
- **rx_ensembl**: a character or factor with an ensembl identifier for the genename of the enzyme.
- **rx_ensembltrans**: a character or factor with an ensembl identifier for the transcript being translated into the enzyme.
- **rx_entrezid**: a character or factor with an entrezid for the genename of the enzyme.

*(default: reactions = NULL)*

**specifiers** An optional data.frame containing the following columns:

- **mod_id**: a integer value per modification and the link to the modification data.frame.
- **spec_type**: a character or factor referencing a type of specifier, e.g. snotRNA. Not checked for validity.
- **spec_genename**: a character or factor referencing a genename for the specifier directing an enzyme to the specific location for the modification to be incorporated.
- **spec_ensembl**: a character or factor with an ensembl identifier for the genename of the specifer.
- **spec_ensembltrans**: a character or factor with an ensembl identifier for the transcript being translated into the specifier.
• `spec_entrezid`: a character or factor with an entrezid for the gene-name of the specifier.
  (default: `specifiers = NULL`)

**references**
An optional data frame containing the following columns:
• `mod_id`: an integer value per modification and the link to the modification data frame.
• `ref_type`: a character or factor with a reference type, e.g. PMID. Is not checked for validity.
• `ref`: a character or factor with a reference value, e.g. a specific pubmed id or an journal article. Is not checked for validity.
  (default: `references = NULL`)

**metadata**
An optional data frame containing the following columns:
• `name`: a character value used as name
• `value`: a character value
This dataframe will be returned by `metadata()` (default: `metadata = NULL`)

**reassign.ids**
TRUE or FALSE Controls how internal `mod_ids` should be assigned. If `reassign.ids` is FALSE (the default) and if the ids are supplied, then they are used as the internal ids, otherwise the internal ids are assigned in a way that is compatible with the order defined by ordering the modifications first by chromosome, then by strand, then by start, and finally by end.

**Value**
a `EpiTxDb` object.

**See Also**
• `makeEpiTxDbFromGRanges` for creating a `EpiTxDb` object from a `GRanges` object and its metadata columns
• `makeEpiTxDbFromRMBase` for creating a `EpiTxDb` object from RMBase online resources
• `makeEpiTxDbFromtRNAdb` for creating a `EpiTxDb` object from tRNAdb online resources
• `shortName` and `ModRNAString` for information on `ModRNAString` objects.

**Examples**

```r
mod <- data.frame("mod_id" = 1L,
                   "mod_type" = "m1A",
                   "mod_name" = "m1A_1",
                   "mod_start" = 1L,
                   "mod_end" = 1L,
                   "mod_strand" = "+",
                   "sn_id" = 1L,
                   "sn_name" = "test")
rx <- data.frame(mod_id = 1L,
                  rx_genename = "test",
                  rx_rank = 1L,
                  rx_mod_id = 1L)
```
makeEpiTxDbFromGRanges

Create a EpiTxDb object from a GRanges object

Description

makeEpiTxDbFromGRanges extracts informations from a GRanges object. The following metadata columns can be used:

- mod_id, mod_type, mod_name and tx_ensembl. The first three are mandatory, whereas tx_ensembl is optional.
- rx_genename, rx_rank, rx_ensembl, rx_ensembltrans and rx_entrezid
- spec_type, spec_genename, spec_ensembl, spec_ensembltrans and spec_entrezid
- ref_type and ref

... and passed on the makeEpiTxDb.

Usage

makeEpiTxDbFromGRanges(gr, metadata = NULL, reassign.ids = FALSE)

Arguments

gr A GRanges object, which contains at least the mandatory columns.
metadata A 2-column data.frame containing meta information to be included in the EpiTxDb object. This data.frame is just passed to makeEpiTxDb. See makeEpiTxDb for more information about the format of metadata. (default: metadata = NULL)
reassign.ids = FALSE

Value

a EpiTxDb object.
**makeEpiTxDbFromRMBase**

Create a EpiTxDb object from RMBase v2.0 online resources

**Description**

`makeEpiTxDbFromRMBase` will make use of the RMBase v2.0 online resources.

**Usage**

```r
EPITXDB_RMBASE_URL
downloadRMBaseFiles(organism, genome, modtype)

makeEpiTxDbFromRMBase(
  organism,
  genome,
  modtype,
  tx = NULL,
  sequences = NULL,
  metadata = NULL,
  reassign.ids = FALSE,
  verbose = FALSE
)

getRMBaseDataAsGRanges(files, verbose = FALSE)

makeEpiTxDbFromRMBaseFiles(
  files,
  tx = NULL,
  sequences = NULL,
  metadata = NULL,
  reassign.ids = FALSE,
  verbose = FALSE
)

listAvailableOrganismsFromRMBase()
```

**Examples**

```r
library(GenomicRanges)
gr <- GRanges(seqnames = "test",
ranges = IRanges::IRanges(1,1),
strand = "+",
DataFrame(mod_id = 1L,
  mod_type = "Am",
  mod_name = "Am_1"))
etdb <- makeEpiTxDbFromGRanges(gr)
```
listAvailableGenomesFromRMBase(organism)

listAvailableModFromRMBase(organism, genome)

Arguments

organism A character value, which must match an organism descriptor on the RMBase download website.

geno A character value, which must match a genome descriptor on the RMBase download website.

modtype A character value, which must match one or more modification descriptors on the RMBase download website.

tx A GRangesList object which will be used to shift the genomic coordinates to transcript coordinates. This is optional, but highly recommended. (default: tx = NULL).

sequences A named DNAStringSet or RNAStringSet, which will be used to check whether the defined modifications are compatible with the original base. This uses removeIncompatibleModifications() function from the Modstrings package.

metadata, reassign.ids

See makeEpiTxDb

verbose TRUE or FALSE: Should verbose message be printed?

files From organism, genome and modtype the available files will be downloaded using the BiocFileCache interface and passed on to makeEpiTxDbFromRMBaseFiles. However, individual files can be provided as well.

Format

An object of class character of length 1.

Value

a EpiTxDb object.

makeEpiTxDbFromtRNAdb Create a EpiTxDb object from tRNAdb resources

Description

makeEpiTxDbFromtRNAdb will make use of the tRNAdb online resources and extract the modification information from the RNA database.

If a named DNAStringSet is provided as sequences, the result from the tRNAdb will be matched against the sequences. Valid matches will be used as transcript identifiers and returned after a check of modification compatibility with the provided sequence. By this process multiple copies of transcripts can be associated with a single modification.

makeEpiTxDbFromtRNAdb uses the functions provided by the tRNAdbImport package. import.tRNAdb will be used with database = "RNA" and the three different values for origin.
**Usage**

```r
gettRNAdbDataAsGRanges(
  organism,
  sequences = NULL,
  dbURL = tRNAdbImport::TRNA_DB_URL
)
```

```r
makeEpiTxDbFromtRNAdb(
  organism,
  sequences = NULL,
  metadata = NULL,
  dbURL = tRNAdbImport::TRNA_DB_URL
)
```

```r
listAvailableOrganismsFromtRNAdb()
```

**Arguments**

- `organism` A character value for an organism available on the tRNAdb website.
- `sequences` A named DNAStringSet or RNAStringSet, which will be used to associate a tRNAdb result with a specific transcript.
- `dbURL` The URL to the tRNA db website.
- `metadata` See `makeEpiTxDb`

**Value**

a EpiTxDb object.

**References**


**Examples**

```r
## Not run:
# getting just the annotation data
etdb <- makeEpiTxDbFromtRNAdb("Saccharomyces cerevisiae")

# For associating the result with transcripts, provide and additional
# named DNAStringSet object. Matching will be done against each sequence
# allowing 5 mismatches and indels. The final result will be checked for
# validity regarding the identity of the modifications
etdb <- makeEpiTxDbFromtRNAdb("Saccharomyces cerevisiae",
                               some_transcript_sequences)
```

## End(Not run)
modifications

Getting modification data from a EpiTxDb-object

Description

modifications and modificationsBy are functions to extract modification annotation from a EpiTxDb object.

modifiedSeqsByTranscript returns a ModRNAStringSet from a EpiTxDb object and compatible RNAStringSet object. This used the combineIntoModstrings() function from the Modstrings package.

Usage

modifications(
  x,
  columns = c("mod_id", "mod_type", "mod_name"),
  filter = NULL,
  use.names = FALSE,
  ...
)

modificationsBy(
  x,
  by = c("seqnames", "mod_type", "reaction", "specifier", "specifier_type"),
  ...
)

modifiedSeqsByTranscript(x, sequences, ...)

## S4 method for signature 'EpiTxDb'
modifications(
  x,
  columns = c("mod_id", "mod_type", "mod_name"),
  filter = NULL,
  use.names = FALSE
)

## S4 method for signature 'EpiTxDb'
modificationsBy(
  x,
  by = c("seqnames", "modtype", "reaction", "specifier", "specifiertype")
)

## S4 method for signature 'EpiTxDb,DNAStringSet'
modifiedSeqsByTranscript(x, sequences)
positionSequence

Arguments

x  
a EpiTxDb

columns  
Columns to include in the result. If the vector is named, those names are used for the corresponding column in the element metadata of the returned object. (default: columns = c("mod_id","mod_type","mod_name"))

filter  
Either NULL or a named list of vectors to be used to restrict the output. Valid names for this list are: "mod_id", "mod_type", "mod_name", "sn_id", "sn_name", "rx_genename", "rx_ensembl", "rx_ensembltrans", "rx_entrezid", "spec_genename", "spec_type", "spec_ensembl", "spec_ensembltrans", "spec_entrezid", "ref_type" and "ref". (default: filter = NULL)

use.names  
TRUE or FALSE. If TRUE, the modification names are set as the names of the returned object. (default: use.names = FALSE)

...  
Not used.

by  
By which information type should the result be split into? A character value from one of the following values:

• seqnames
• mod_type
• reaction
• specifier
• specifier_type

sequences  
A RNAStringSet, which can be used as input for combineIntoModstrings(). See ?combineIntoModstrings for additional details.

Value

a GRanges object for modifications and a GRangesList for modificationsBy.

Examples

etdb_file <- system.file("extdata", "EpiTxDb.Hs.hg38.snoRNAdb.sqlite", package="EpiTxDb")
etdb <- loadDb(etdb_file)
etdb

positionSequence  
Generate integer sequences from position information of Ranges

Description

positionSequence generates sequences of integer values along the range information of x. This can be used for navigating specific positions on a range information.
Usage

positionSequence(x, order = FALSE, decreasing = FALSE)

## S4 method for signature 'Ranges'
positionSequence(x, order = FALSE, decreasing = FALSE)

## S4 method for signature 'RangesList'
positionSequence(x, order = FALSE, decreasing = FALSE)

## S4 method for signature 'Ranges'
as.integer(x)

Arguments

x
a Ranges object, like a GRanges or IRanges, or a RangesList object, like a GRangesList or IRangesList

order
TRUE or FALSE: Should the position be ordered? (default: order = FALSE)

decreasing
TRUE or FALSE: If order = TRUE Should the position be ordered in a decreasing order? (default: order = FALSE)

Value

a integer vector if x is a GRanges object and a IntegerList if x is a GRangesList

Examples

library(GenomicRanges)
# Returns an integer vector
gr <- GRanges("chr1:1-5:+")
positionSequence(gr)
gr2 <- GRanges("chr1:1-5:-")
positionSequence(gr)
# returns an IntegerList
grl <- GRangesList("1" = gr,"2" = gr,"3" = gr2) # must be named
positionSequence(grl)

<table>
<thead>
<tr>
<th>rescale</th>
<th>Rescaling Ranges object</th>
</tr>
</thead>
</table>

Description

rescale() rescales IRanges, GRanges, IRangesList and GRangesList by using minima and maxima derived from to and from.
Usage

rescale(x, to = 1L, from = 1L)

## S4 method for signature 'IRanges'
rescale(x, to = 1L, from = 1L)

## S4 method for signature 'IRangesList'
rescale(x, to = 1L, from = 1L)

## S4 method for signature 'GRanges'
rescale(x, to = 1L, from = 1L)

## S4 method for signature 'GRangesList'
rescale(x, to = 1L, from = 1L)

Arguments

x         a IRanges, GRanges, IRangesList and GRangesList object
to, from   an IRanges object, a character vector coercible to IRanges or a integer vector parallel to x or with length = 1L.

Value

an object of the same type and dimensions as x

Author(s)

H. Pagès, F. Ernst

See Also

IRanges for details on character vectors coercible to IRanges.

Examples

x <- IRanges("5-10")
# widen the ranges
rescale(x, 100, 10)
# widen and shift
rescale(x, "31-60", "5-14")
select

Using the "select" interface on EpiTxDb objects

Description

As expected for an AnnotationDb object, the general accessors select, keys, columns and keytypes can be used to get information from a EpiTxDb object.

Usage

## S4 method for signature 'EpiTxDb'
select(x, keys, columns, keytype, ...)

## S4 method for signature 'EpiTxDb'
columns(x)

## S4 method for signature 'EpiTxDb'
keys(x, keytype, ...)

## S4 method for signature 'EpiTxDb'
keytypes(x)

Arguments

x a EpiTxDb object
keys, columns, keytype, ...

See AnnotationDb for more comprehensive description of the methods select, keys, columns and keytypes and their arguments.

Value

A data.frame object for select() and a character vector for keytypes(), keys() and columns().

Examples

etdb_file <- system.file("extdata", "EpiTxDb.Hs.hg38.snoRNAdb.sqlite",
                          package="EpiTxDb")
etdb <- loadDb(etdb_file)
etdb
**shiftTranscriptToGenomic**

*Shift GRanges coordinates based on another GRanges object*

**Description**

shiftGenomicToTranscript shifts positions of a GRanges object based on coordinates of another GRanges object. The most common application is to shift genomic coordinates to transcript coordinates, which is reflected in the name. shiftTranscriptToGenomic implements the reverse operation.

Matches are determined by findOverlaps for shiftGenomicToTranscript and by findMatches for shiftTranscriptToGenomic using the seqnames of the subject and the names of tx.

**Usage**

shiftTranscriptToGenomic(subject, tx)

shiftGenomicToTranscript(subject, tx)

### S4 method for signature 'GRanges,GRangesList'

shiftTranscriptToGenomic(subject, tx)

### S4 method for signature 'GRangesList,GRangesList'

shiftTranscriptToGenomic(subject, tx)

### S4 method for signature 'GRanges,GRangesList'

shiftGenomicToTranscript(subject, tx)

### S4 method for signature 'GRangesList,GRangesList'

shiftGenomicToTranscript(subject, tx)

**Arguments**

- `subject` a GRanges or GRangesList object
- `tx` a named GRangesList object.

**Value**

a GRanges or GRangesList object depending on the type of subject

**Examples**

```r
library(GenomicRanges)
# Construct some example data
subject1 <- GRanges("chr1", IRanges(3, 6),
                     strand = "+")
subject2 <- GRanges("chr1", IRanges(c(17,23), width=3),
                     strand = "+")
```
subject3 <- GRanges("chr2", IRanges(c(51, 54), c(53, 59)),
                      strand = "-")

subject <- GRangesList(a=subject1, b=subject2, c=subject3)

# shift to transcript coordinates. Since the third subject does not have
# a match in tx it is dropped with a warning
shifted_grl <- shiftGenomicToTranscript(subject, tx)

# ... and back
shifted_grl2 <- shiftTranscriptToGenomic(shifted_grl, tx)

# comparison of ranges work. However the seqlevels differ
ranges(shifted_grl2) == ranges(subject[[list(1, c(1,1), c(1,2))]])
Index

* datasets
  EpiTxDb-data, 4
  makeEpiTxDbFromRMBase, 9
* internal
  EpiTxDb-package, 2
  EpiTxDb (EpiTxDb-class), 3
  ?combineIntoModstrings, 13
  AnnotationDb, 3, 4, 16
  as.integer, Ranges-method (positionSequence), 13
  as.list, EpiTxDb-method (EpiTxDb-class), 3
  BiocFileCache, 10
  columns (select), 16
  columns, EpiTxDb-method (select), 16
  combineIntoModstrings(), 12, 13
  DNAStringSet, 10
  downloadRMBaseFiles
    (makeEpiTxDbFromRMBase), 9
  EpiTxDb, 5, 12, 13, 16
  EpiTxDb (EpiTxDb-class), 3
  EpiTxDb-class, 3
  EpiTxDb-data, 4
  EpiTxDb-package, 2
  EpiTxDb-package#, 5
  EPIITXDB_RMBASE_URL
    (makeEpiTxDbFromRMBase), 9
  findMatches, 17
  findOverlaps, 17
  getRMBaseDataAsGRanges
    (makeEpiTxDbFromRMBase), 9
  gettRNAdbDataAsGRanges
    (makeEpiTxDbFromtRNAdb), 10
  GRanges, 4, 7, 8, 13, 14, 17
  import.tRNAdb, 10
  IRanges, 14, 15
  IRangesList, 14
  keys (select), 16
  keys, EpiTxDb-method (select), 16
  keytypes (select), 16
  keytypes, EpiTxDb-method (select), 16
  listAvailableGenomesFromRMBase
    (makeEpiTxDbFromRMBase), 9
  listAvailableModFromRMBase
    (makeEpiTxDbFromRMBase), 9
  listAvailableOrganismsFromRMBase
    (makeEpiTxDbFromRMBase), 9
  listAvailableOrganismsFromtRNAdb
    (makeEpiTxDbFromtRNAdb), 10
  makeEpiTxDb, 4, 5, 8, 10, 11
  makeEpiTxDbFromGRanges, 4, 7, 8
  makeEpiTxDbFromRMBase, 4, 7, 9
  makeEpiTxDbFromRMBaseFiles
    (makeEpiTxDbFromRMBase), 9
  makeEpiTxDbFromtRNAdb, 4, 7, 10
  modifications, 4, 12
  modifications, EpiTxDb-method
    (modifications), 12
  modificationsBy, 4
  modificationsBy (modifications), 12
  modificationsBy (modifications), 12
  modifiedSeqsByTranscript
    (modifications), 12
  modifiedSeqsByTranscript, EpiTxDb, DNAStringSet-method
    (modifications), 12
  ModRNAString, 7
  ModRNAStringSet, 12
organism,EpiTxDb-method
  (EpiTxDb-class), 3

positionSequence, 13
positionSequence,Ranges-method
  (positionSequence), 13
positionSequence,RangesList-method
  (positionSequence), 13

removeIncompatibleModifications(), 10
rescale, 14
rescale,GRanges-method(rescale), 14
rescale,GRangesList-method(rescale), 14
rescale,IRanges-method(rescale), 14
rescale,IRangesList-method(rescale), 14
rmbase_data(EpiTxDb-data), 4

select, 4, 16
select,EpiTxDb-method(select), 16
Seqinfo, 3
seqinfo,EpiTxDb-method(EpiTxDb-class), 4

seqlevels,EpiTxDb-method
  (EpiTxDb-class), 3
shiftGenomicToTranscript, 4
shiftGenomicToTranscript
  (shiftTranscriptToGenomic), 17
shiftGenomicToTranscript,GRanges,GRangesList-method
  (shiftTranscriptToGenomic), 17
shiftGenomicToTranscript,GRangesList,GRangesList-method
  (shiftTranscriptToGenomic), 17
shiftTranscriptToGenomic, 4, 17
shiftTranscriptToGenomic,GRanges,GRangesList-method
  (shiftTranscriptToGenomic), 17
shiftTranscriptToGenomic,GRangesList,GRangesList-method
  (shiftTranscriptToGenomic), 17
shortName, 7
tRNAdbImport, 10