Package 'EpiTxDb'

December 18, 2024

Type Package

Title Storing and accessing epitranscriptomic information using the AnnotationDbi interface

Version 1.19.0

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, httr, xml2, curl, rex, GenomicFeatures, txdbmaker, GenomicRanges, GenomeInfoDb, BiocGenerics, BiocFileCache, S4Vectors, IRanges, RSQLite, DBI, Biostrings, tRNAdbImport

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

Collate 'AllGenerics.R' 'EpiTxDb-SELECT-helpers.R' 'EpiTxDb-schema.R' 'EpiTxDb.R' 'EpiTxDb-class.R' 'makeEpiTxDb.R' 'makeEpiTxDbFromGRanges.R' 'shiftGenomicToTranscript.R' 'makeEpiTxDbFromRMBase.R' 'makeEpiTxDbFromtRNAdb.R' 'modifications.R' 'modificationsBy.R' 'ranges-helpers.R' 'select-methods.R'

RoxygenNote 7.3.1

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

URL https://github.com/FelixErnst/EpiTxDb

VignetteBuilder knitr

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

git_branch devel

git_last_commit 49a7f33

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-12-18

Author Felix G.M. Ernst [aut, cre] (ORCID:
<https://orcid.org/0000-0001-5064-0928>)

Maintainer Felix G.M. Ernst <felix.gm.ernst@outlook.com>

Contents

`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
- Report bugs at https://github.com/FelixErnst/EpiTxDb/issues

---

EpiTxDb-class *EpiTxDb objects*

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

```
## 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 it's 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 transfering genomic to transcript coordinates and back again.

Examples

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

# general methods
seqinfo(etdb) #
seqlevels(etdb) # 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

Jia-Jia Xuan, Wen-Ju Sun, Ke-Ren Zhou, Shun Liu, Peng-Hui Lin, Ling-Ling Zheng, Liang-Hu Qu, Jian-Hua Yang (2017): "RMBase v2.0: Deciphering the Map of RNA Modifications from Epitranscriptome Sequencing Data." Nucleic Acids Research, Volume 46, Issue D1, 4 January 2018, Pages D327–D334. doi: 10.1093/nar/gkx934

Jühling, Frank; Mörl, Mario; Hartmann, Roland K.; Sprinzl, Mathias; Stadler, Peter F.; Pütz, Joern (2009): "TRNAdb 2009: Compilation of tRNA Sequences and tRNA Genes." Nucleic Acids Research 37 (suppl_1): D159–D162. doi: 10.1093/nar/gkn772

Sprinzl, Mathias; Vassilenko, Konstantin S. (2005): "Compilation of tRNA Sequences and Sequences of tRNA Genes." Nucleic Acids Research 33 (suppl_1): D139–D140. doi: 10.1093/nar/gki012

`makeEpiTxDb` *Creating a* `EpiTxDb` *from user supplied annotations as* `data.frame`*s*

Description

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

This functions 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` containg 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 modificaion as a `character` or `factor`.
- `sn_id`: a `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` containg 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` containg 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. `snoRNA`. 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 specifier.
- `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 containg the following columns:

- mod_id: a 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 containg 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 it's 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

```
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_ensembl = "test",
                  rx_ensembltrans = "test",
                  rx_entrezid = "test")
spec <- data.frame(mod_id = 1L,
                   spec_type = "test",
                   spec_genename = "test",
                   spec_ensembl = "test",
                   spec_ensembltrans = "test",
                   spec_entrezid = "test")
ref <- data.frame(mod_id = 1L,
                  ref_type = "test",
                  ref = "test")
etdb <- makeEpiTxDb(mod,rx,spec,ref)
```

---

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.

Examples

```
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)
```

---

makeEpiTxDbFromRMBase *Create a* EpiTxDb *object from RMBase v2.0 online resources*

Description

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

Usage

```
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()
```

```
listAvailableGenomesFromRMBase(organism)

listAvailableModFromRMBase(organism, genome)
```

Arguments

organism	A character value, which must match an organism descriptor on the RMBase download website.
genome	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 removeIncompatibleModifica function from the Modstrings package.
metadata, reassign.ids	See makeEpiTxDb
verbose	TRUE or FALSE: Should verbose message be prined?
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

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

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

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

Juehling F, Moerl M, Hartmann RK, Sprinzl M, Stadler PF, Puetz J. 2009. "tRNAdb 2009: compilation of tRNA sequences and tRNA genes." Nucleic Acids Research, Volume 37 (suppl_1): D159–162. doi:10.1093/nar/gkn772.

Examples

```
## 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)
```

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)
```

---

rescale *Rescaling* Ranges *object*

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 a 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 vecor 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

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

```
                    strand = c("+","-"))
subject3 <- GRanges("chr2", IRanges(c(51, 54), c(53, 59)),
                    strand = "-")
subject <- GRangesList(a=subject1, b=subject2, c=subject3)
tx1 <- GRanges("chr1", IRanges(1, 40),
               strand="+")
tx2 <- GRanges("chr1", IRanges(10, 30),
               strand="+")
tx3 <- GRanges("chr2", IRanges(50, 60),
               strand="-")
tx <- GRangesList(a=tx1, b=tx2, c=tx3)

# 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