Package 'RNAModR'

October 28, 2019

Type Package

Title Functional Analysis of RNA Modifications

Version 0.2.3

Author Maurits Evers <maurits.evers@anu.edu.au>

Maintainer Maurits Evers <maurits.evers@anu.edu.au>

Description Transcriptome-wide analysis of RNA modifications.

Imports AnnotationDbi,
Biostrings,
beanplot,
GenomeInfoDb,
gplots,
graphics,
grDevices,
magrittr,
methods,
RSQLite,
S4Vectors,
stats,
utils

Depends GenomicRanges,
GenomicFeatures,
IRanges,
R (>= 3.5.0)

Suggests knitr,
testthat,
rmarkdown

License GPL-3 | file LICENSE

LazyData false

RoxygenNote 6.1.1

Encoding UTF-8

VignetteBuilder knitr

R topics documented:

`BuildTx` *Build a custom transcriptome.*

Description

Build a custom, organism-specific transcriptome. See 'Details'.

Usage

```
BuildTx(genomeVersion = c("hg38", "hg19", "hg18", "mm10", "mm9", "mm8",
  "dm6", "dm3", "dm2", "sacCer3", "sacCer2", "sacCer1"), force = FALSE)
```

Arguments

`genomeVersion`	A character string; refers to a specific reference genome assembly version; default is `"hg38"`.
`force`	A logical scalar; if `TRUE` force rebuild of transcriptome; this will overwrite existing data.

Details

The function builds an organism-specific transcriptome containing one transcript per unique Entrez ID; the transcript is selected from all UCSC RefSeq annotation-based isoforms as the transcript with the longest CDS, and longest upstream/downstream adjoining UTRs. Transcript segments are stored per transcript section, and written into a `.RData` file. For most operations, the user will run this function once, and continue with further downstream analyses. Various RNAModR routines will automatically load the transcriptome data to e.g. map sites to and from the transcriptome. Currently, RNAModR supports analyses of human, mouse, fruitfly and yeast data, based on different reference genome versions:

- Homo sapiens: hg38, hg19, hg18
- Mus musculus: mm10, mm9, mm8
- Drosophila melanogaster: dm6, dm3, dm2
- Cerevisiae saccharomyces: sacCer3, sacCer2, sacCer1

Reconstruction of existing transcriptome data can be achieved by running `BuildTx` with `force = TRUE`. Note that this will overwrite the existing RData file. Running `BuildTx` with `sanityCheck = TRUE` performs additional checks of the various transcriptome components, and is intended for debugging purposes only. It is usually safe to run with the default `sanityCheck = FALSE`.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
# Build the human hg38-based reference transcriptome
BuildTx("hg38");

## End(Not run)
```

Datasets *Internal raw datasets.*

Description

Internal raw datasets in BED6 format. All genome positions where converted to the most recent genome assembly version; i.e. for human data, all positions are given in GRCh38/hg38 coordinates. For mouse data, positions are given in mm10 coordinates.

Details

`bsRNAseq_m5C_Squires2012_hg38.bed` bsRNA-seq 5-methylcytosine (m5C) sites. Reference: https://www.ncbi.nlm.nih.gov/pubmed/22344696

`HITSCLIP_eIF4A3_Sauliere2012_hg38.bed` HITSCLIP eIF4AIII binding sites. Reference: https://www.ncbi.nlm.nih.gov/pubmed/23085716

`MeRIPseq_m1A_Dominissini2016_hg38.bed` MeRIP-seq N(1)-methyladenosine (m1A) sites. Reference: https://www.ncbi.nlm.nih.gov/pubmed/26863196

`miCLIP_m6A_Linder2015_hg38.bed` miCLIP N(6)-methyladenosine (m6A) sites. Reference: https://www.ncbi.nlm.nih.gov/pubmed/26121403

`PARCLIP_AGO1234_Hafner2010_hg38.bed` PAR-CLIP AGO1-4 binding sites. Reference: https://www.ncbi.nlm.nih.gov/pubmed/20371350

`PARCLIP_eIF3_Meyer2015_hg38.bed` PAR-CLIP eIF3 binding sites. Reference: https://www.ncbi.nlm.nih.gov/pubmed/26593424

`PARCLIP_eIF3_Lee2015_hg38.bed` PAR-CLIP eIF3 binding sites. Reference: https://www.ncbi.nlm.nih.gov/pubmed/25849773

`PARCLIP_YTHDF2_Wang2014_hg38.bed` PAR-CLIP YTHDF2 binding sites. Reference: https://www.ncbi.nlm.nih.gov/pubmed/24284625

`PseudoU_Schwartz2014_Li2015_hg38.bed` Merged Psi-seq and CeU-seq pseudouridine (Psi) sites. References: https://www.ncbi.nlm.nih.gov/pubmed/25219674, https://www.ncbi.nlm.nih.gov/pubmed/26075521

`TargetScan_miRNAtargets_hg38.bed` TargetScan-based miRNA target sites. Reference: http://www.targetscan.org/vert_72/

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
bedFile <- ReadBED(system.file(
    "extdata",
    "miCLIP_m6A_Linder2015_hg38.bed",
    package = "RNAModR")

## End(Not run)
```

DownsampleTxLoc *Downsample a* `txLoc` *object.*

Description

Downsample a `txLoc1` object based on the number of sites per region from a `txLoc2` object.

Usage

```
DownsampleTxLoc(txLoc1, txLoc2, seed = NULL)
```

Arguments

`txLoc1`	A `txLoc` object; this is the `txLoc` object that will be downsampled.
`txLoc2`	A `txLoc` object; this is the `txLoc` object that will be used as a target for the downsampling.
`seed`	A single value, interpreted as an `integer`, or `NULL`; this is to ensure reproducibility when subsampling `txLoc2` sites; default is `NULL`.

Value

A txLoc object.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

FilterTxLoc *Filter sections of a* txLoc *object.*

Description

Filter sections of a txLoc object.

Usage

```
FilterTxLoc(txLoc, filter = NULL)
```

Arguments

txLoc	A txLoc object.
filter	A character vector; only keep transcript regions that match entries from filter; if NULL, return the original txLoc object.

Value

A txLoc object.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

GenerateNull *Generate a list of null sites.*

Description

Generate a list of null sites. See 'Details'.

Usage

```
GenerateNull(txLoc, id = NULL, method = c("ntAbund", "perm"),
  nt = "C", showPb = TRUE)
```

Arguments

txLoc	A txLoc object.
id	A character string; identifier for null sites; if NULL then id = "null"; default is NULL.
method	A character string; specifies the method used to the generate null distribution; if method == "ntAbund" the position of all nucleotides specified by nt will be used as null sites; if method == "perm" (DEPRECATED, see 'Details') then null sites will be generated by uniform-randomly shuffling candidatepositions from locus within the corresponding transcript region; default is "ntAbund".
nt	A single character; if method == "ntAbund", use nt to derive distribution of null sites.
showPb	A logical scalar; if TRUE show a progress bar; default is TRUE.

Details

The function generates a null distribution of single-nucleotide sites across different transcript sections, and returns a txLoc object. Two different methods can be employed:

1. method = "ntAbund": Null sites are generated based on the position of all non-modified nucleotides of type nt in those transcript sections that also contain a modified site of the same type and as specified in locus. For example, if locus contains a list of m^{6}A sites, the list of null sites consists of all non-methylated adenosines in those transcripts that contain at least one m^{6}A site.
2. method = "perm": Null sites are generated by randomly permuting the position of sites from locus uniformly within the corresponding transcript section. Note that this will generate a list of null sites with the same abundance ratios across transcript sections as the list of sites from locus. It is therefore not useful for assessing an enrichment of sites within a particular transcript section. In fact, this method should not be used and is included purely for paedagogical purposes (to demonstrate the importance of a sensible null distribution). It is likely that this method will be removed from future RNAModR versions.

It is import to emphasise that any downstream enrichment analysis may depend critically on the choice of the null distribution. For example, a position permution-based null distribution may not be a valid null distribution, if the distribution of nucleotides is highly non-uniform across a transcript section. This is the case e.g. for the spatial distribution of cytosines within and across the 5'UTR, CDS and/or 3'UTR. In this case, a permutation- based distribution of cytosines will not give a sensible null distribution. Instead, a sensible null distribution can be derived from the position of all cytosines in the relevant transcript region containing the methylated cytosine site in locus. method = "ntAbund" generates a list of null sites using this approach.

Value

A txLoc object. See 'Details'.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
bedFile <- system.file("extdata",
                       "miCLIP_m6A_Linder2015_hg38.bed",
                       package = "RNAModR")
sites <- ReadBED(bedFile)
posSites <- SmartMap(sites, id = "m6A", refGenome = "hg38")
negSites <- GenerateNull(posSites, method = "ntAbund", nt = "A")

## End(Not run)
```

`GetEEJunct` *Get exon-exon boundary sites from reference transcriptome*

Description

Get exon-exon boundary sites from transcriptome. See 'Details'.

Usage

```
GetEEJunct(refGenome = "hg38", filter = c("CDS", "5'UTR"))
```

Arguments

`refGenome`	A character string; specifies a specific reference genome assembly version based on which the matching transcriptome is loaded; default is `"hg38"`.
`filter`	A character vector; only consider transcript sections specified in `filter`; default is `c("CDS","5'UTR")`.

Details

The function extracts exon-exon boundary sites (EEBS) from a reference trancriptome specified by `refGenome`, and returns a `GRanges` object. Boundary sites are defined as the location of those exonic single nucleotides that are closest upstream to the intronic donor site.

Value

A `GRanges` object. See 'Details'.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

GetMotifLoc *Get loci of motif(s).*

Description

Get loci of motif(s) from transcriptome. See 'Details'.

Usage

```
GetMotifLoc(motif = NULL, refGenome = "hg38", id = NULL, maxMM = 0,
  showPb = TRUE)
```

Arguments

motif	A character vector; specifies the query sequences that will be matched against the reference transcriptome.
refGenome	A character string; specifies the reference genome version; default is "hg38".
id	A character string; specifies an identifier for motif; default is NULL
maxMM	An integer scalar; specifies the maximum number of mismatches that are allowed during the motif matching; default is 0.
showPb	A logical scalar; if TRUE show a progress bar; default is TRUE.

Details

Based on a set of query sequences motif, extract the loci of matching sequences within different regions of a reference transcriptome refGenome. The function returns a txLoc object including transcriptomic and genomic loci of the query motif(s). The maximum number of mismatches allowed in the motif search can be adjusted through maxMM. Note that the motif search may take a few minutes, depending on the number of motif(s), hardware, and size of the transcriptome.

Value

A txLoc object.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
PAS <- GetMotifLoc(id = "PAS")
PAS <- FilterTxLoc(PAS, c("3'UTR", "CDS", "5'UTR"))
PlotSpatialDistribution(PAS, absolute = FALSE)

## End(Not run)
```

GetSplicingSites *Get splicing sites from reference transcriptome*

Description

Get splicing sites from transcriptome. See 'Details'.

Usage

```
GetSplicingSites(refGenome = "hg38", filter = c("CDS", "5'UTR"))
```

Arguments

refGenome	A character string; specifies a specific reference genome assembly version based on which the matching transcriptome is loaded; default is "hg38".
filter	A character vector; only consider transcript sections specified in filter; default is c("CDS","5'UTR").

Details

The function extracts splicing sites from a reference trancriptome specified by refGenome, and returns a GRanges object. Splicing sites are identified as either donor (splicing site at the 5' end of the intron) or acceptor (splicing site at the 3' end of the intron) site.

Value

A GRanges object. See 'Details'.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

PlotGC *Plot GC content.*

Description

Plot and assess the difference in the distributions of GC content within a window around sites from two txLoc objects. See 'Details'.

Usage

```
PlotGC(txLoc1, txLoc2, flank = 10, norm_region = FALSE,
  downsample = TRUE, seed = NULL)
```

Arguments

txLoc1	A txLoc object.
txLoc2	A txLoc object.
flank	An integer scalar; see 'Details'.
norm_region	A logical scalar; if TRUE normalise the GC content in the window to the GC content of the corresponding transcript region; default is FALSE.
downsample	A logical scalar; if TRUE, subsample sites from txLoc2 to match the number of sites per transcript region from txLoc1; default is TRUE.
seed	A single value, interpreted as an integer, or NULL; this is to ensure reproducibility when subsampling txLoc2 sites; ignored when downsample == FALSE; default is NULL.

Details

The function calculates the GC content within a window around every site from two txLoc objects. The window is defined by extending the position of every txLoc site upstream and downstream by flank nucleotides (if possible). The means of the resulting GC content distributions are assessed using a two-sample two-tailed t-test. If norm_region = TRUE, the GC content in the window is normalised to the GC content of the entire transcript region. If downsample = TRUE, the number of windows/sites from the *second* txLoc2 object is downsampled to match the number of windows/sites from the txLoc1 object. This is useful (and therefore the default setting) when comparing the GC distribution around positive and null sites, as the list of null sites is often significantly larger than that of the positive sites. Note that this function calls GetGC, which performs the sequence extraction and GC calculation. See ?GetGC for details.

Value

NULL.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

PlotOverlap *Plot overlap of sites.*

Description

Plot overlap of sites from two txLoc objects. See 'Details'.

Usage

```
PlotOverlap(txLoc1, txLoc2)
```

Arguments

txLoc1	A txLoc object.
txLoc2	A txLoc object.

Details

The function plots one or multiple Venn diagrams showing the spatial overlap between entries from two txLoc objects. Two features are defined as overlapping, if they overlap by at least one nucleotide. Overlaps are determined using the function GenomicRanges::countOverlaps.

Value

NULL.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

PlotRelDistDistribution

Plot distribution of relative distances.

Description

Plot distribution of relative distances between sites from two txLoc objects. See 'Details'.

Usage

```
PlotRelDistDistribution(txLoc, txLocRef, flank = 1000, binWidth = 20,
  doBootstrap = TRUE)
```

Arguments

txLoc	A txLoc object.
txLocRef	A txLoc object.
flank	An integer scalar; specifies the absolute maximum relative distance used as a cutoff; default is 1000.
binWidth	An integer scalar; specifies the spatial width by which distances will be binned; default is 20.
doBootstrap	A logical scalar; if YES calculate 95 transcript region; default is TRUE.

Details

The function calculates minimum distances per transcript region, between entries from txLoc relative to txLocRef. Relative distances are shown within a window (-flank, flank), where negative distances correspond to a feature from txLoc that is upstream of a site from txLocRef, and positive distances indicate a feature from txLoc that is downstream of a site from txLocRef. Relative distances are binned in bins of binWidth nt, and shown as an abundance histogram. If doBootstrap = TRUE, 95 are calculated and shown, based on an empirical bootstrap of relative distances.

Value

NULL.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

PlotRelDistEnrichment *Perform enrichment analysis of relative distances.*

Description

Perform enrichment analysis and plot results of two relative distance distributions. See 'Details'.

Usage

```
PlotRelDistEnrichment(txLoc1, txLoc2, txLocRef, flank = 1000,
  binWidth = 20)
```

Arguments

txLoc1	A txLoc object.
txLoc2	A txLoc object.
txLocRef	A txLoc object.
flank	An integer scalar; specifies the absolute maximum relative distance used as a cutoff; default is 1000.
binWidth	An integer scalar; specifies the spatial width by which distances will be binned; default is 20.

Details

The function calculates minimum distances per transcript region, between entries from txLoc1 relative to txLocRef, and txLoc2 relative to txLocRef. Enrichment/depletion is assessed using multiple Fisher's exact tests on the counts per distance bin relative to the counts in all other bins within the window defined by (-flank, flank). Resulting enrichment plots show odds-ratios (including 95% confidence intervals) and associated p-values as a function of relative distance bins. Negative distances indicate sites from txLoc1 and txLoc2 that are *upstream* of sites from txLocRef; positive distances correspond to sites from txLoc1 and txLoc2 that are *downstream* of txLocRef. The bin width and window size can be adjusted with flank and binWidth.

Value

NULL.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

PlotSectionDistribution *Plot piechart of the number of loci in every transcript region*

Description

Plot piechart of the number of loci in every transcript region.

Usage

```
PlotSectionDistribution(txLoc)
```

Arguments

txLoc	A txLoc object.

Value

NULL.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
bedFile <- system.file("extdata",
                       "miCLIP_m6A_Linder2015_hg38.bed",
                       package = "RNAModR")
sites <- ReadBED(bedFile)
posSites <- SmartMap(sites, id = "m6A", refGenome = "hg38")
PlotSectionDistribution(posSites)

## End(Not run)
```

PlotSectionEnrichment *Perform enrichment analysis of sites per transcript region and plot results.*

Description

Perform enrichment analysis of the number of positive sites in txLoc.pos relative to the number of null sites in txLoc.neg per transript region and plot results. Enrichment/depletion is evaluated using (multiple) Fisher's exact test(s). Multiple hypothesis testing correction is applied following the method of Bejamini and Hochberg.

Usage

```
PlotSectionEnrichment(txLoc.pos, txLoc.neg, xAxisLblFmt = 2)
```

Arguments

txLoc.pos	A txLoc object. These correspond to the positive sites.
txLoc.neg	A txLoc object. These correspond to the negative (null) sites.
xAxisLblFmt	Plot extended axis labels. Default is 2. See ??? for details.

Value

NULL.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
bedFile <- system.file("extdata",
                       "miCLIP_m6A_Linder2015_hg38.bed",
                       package = "RNAModR")
sites <- ReadBED(bedFile)
pos <- SmartMap(sites, id = "m6A", refGenome = "hg38")
neg <- GenerateNull(pos, method = "nuclAbundance")
PlotSectionEnrichment(
    FilterTxLoc(pos, c("5'UTR", "CDS", "3'UTR")),
    FilterTxLoc(neg, c("5'UTR", "CDS", "3'UTR")))

## End(Not run)
```

PlotSeqLogo *Plot sequence logo.*

Description

Plot sequence logo.

Usage

```
PlotSeqLogo(txLoc, flank = 5, ylim = c(0, 2))
```

Arguments

txLoc	A txLoc object.
flank	An integer scalar; see 'Details'.
ylim	An integer vector; specifies limits for the y-axis; automatically determined if ymin = NULL; default is c(0,2).

Details

The function determines the sequence logo within a window defined by extending sites from txLoc upstream and downstream by flank nucleotides.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

`PlotSpatialDistribution` *Plot spatial distribution of loci from a* `txLoc` *object.*

Description

Plot spatial distribution of loci from a `txLoc` object within every transcript region.

Usage

```
PlotSpatialDistribution(txLoc, nbreaks = 100, absolute = FALSE,
  binWidth = NULL, posMax = 1000, doBootstrap = TRUE, ...)
```

Arguments

`txLoc`	A `txLoc` object.
`nbreaks`	Number of spatial bins. Default is 100.
`absolute`	Plot spatial distribution in absolute coordinates. Default is `FALSE`.
`binWidth`	Spatial bin width. Overrides `nbreaks` if not `NULL`.
`posMax`	If `absolute == TRUE`, show spatial distribution within a window given by `posMax` from the 5'/3' position of the transcript feature. Default is 1000 nt.
`doBootstrap`	Calculate 95 sites within transcript region. Default is `TRUE`.
`...`	Additional parameters passed to plot.

Value

`NULL`.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
bedFile <- system.file("extdata",
                       "miCLIP_m6A_Linder2015_hg38.bed",
                       package = "RNAModR")
sites <- ReadBED(bedFile)
posSites <- SmartMap(sites, id = "m6A", refGenome = "hg38")
PlotSpatialDistribution(posSites)
PlotSpatialDistribution(posSites,
                        absolute = TRUE,
                        filter = c("5'UTR", "CDS", "3'UTR"),
                        ylim = c(0, 200))

## End(Not run)
```

PlotSpatialEnrichment *Perform spatial enrichment analysis and plot results.*

Description

Perform enrichment analysis of the spatial distribution of positive sites in txLoc.pos relative to the distribution of null sites in txLoc.neg per transcript region and plot results. Enrichment/depletion is evaluated using (multiple) Fisher's exact test(s). Multiple hypothesis testing correction is applied following the method of Bejamini and Hochberg.

Usage

```
PlotSpatialEnrichment(txLoc.pos, txLoc.neg, binWidth = 20,
  posMax = 1000)
```

Arguments

txLoc.pos	A txLoc object. These correspond to the positive sites.
txLoc.neg	A txLoc object. These correspond to the negative (null) sites.
binWidth	Spatial bin width. Default is 20 nt.
posMax	Evaluate enrichment within a window given by posMax. Default is 1000 nt.

Value

NULL.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
bedFile <- system.file("extdata",
                       "miCLIP_m6A_Linder2015_hg38.bed",
                       package = "RNAModR")
sites <- ReadBED(bedFile)
pos <- SmartMap(sites, id = "m6A", refGenome = "hg38")
null <- GenerateNull(posSites, method = "permutation")
PlotSpatialEnrichment(
    FilterTxLoc(pos, c("5'UTR", "CDS", "3'UTR")),
    FilterTxLoc(neg, c("5'UTR", "CDS", "3'UTR")))

## End(Not run)
```

ReadBED *Read BED-formatted file.*

Description

Read BED-formatted file. See 'Details'.

Usage

```
ReadBED(file, collapseRange = TRUE)
```

Arguments

file	A character string; specifies the input BED file.
collapseRange	A logical scalar; should TRUE loci spanning more than one nucleotide be collapsed to a single nucleotide locus corresponding to the midpoint of the range; default is TRUE.

Details

The function opens and reads in a BED6-formatted file, and stores the annotation features in a GRanges object. Chromosome names must follow either UCSC (chr1, ..., chrM, chrX) or Ensembl naming conventions (1, ..., MT, X).

Value

A GRanges object. See 'Details'.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
bedFile <- system.file("extdata",
                       "miCLIP_m6A_Linder2015_hg38.bed",
                       package = "RNAModR")
sites <- ReadBED(bedFile)
```

SmartMap *Map genome coordinates to transcript coordinates.*

Description

Map genome coordinates to transcript coordinates.

Usage

```
SmartMap(gr, id = NULL, refGenome = "hg38", ignore.strand = FALSE,
  showPb = TRUE)
```

Arguments

gr	A GRanges object; specifies the list of of genomic features to be mapped.
id	A character string; specifies a name for loci from gr; if NULL then id = ""; default is NULL.
refGenome	A character string; specifies a specific reference genome assembly version based on which the matching transcriptome is loaded; default is "hg38".
ignore.strand	A logical scalar; if TRUE strand information is ignored when mapping genome coordinates to transcript coordinates; default is FALSE.
showPb	A logical scalar; if TRUE show a progress bar; default is TRUE.

Details

The function maps genomic coordinates from locus to transcript section coordinates. The function automatically loads a reference transcriptome based on refGenome. An error is produced if a reference transcriptome could not be found. This usually means that BuildTx was not yet run successfully. The function returns a txLoc object of mapped positions.

Value

A txLoc object. See 'Details'.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
bedFile <- system.file("extdata",
                       "miCLIP_m6A_Linder2015_hg38.bed",
                       package = "RNAModR")
sites <- ReadBED(bedFile)
txSites <- SmartMap(sites, id = "m6A", refGenome = "hg38")

## End(Not run)
```

SubsampleTxLoc *Subsample a* txLoc *object.*

Description

Subsample a txLoc1 object based on a vector of fractions for every transcript region.

Usage

```
SubsampleTxLoc(txLoc, fractions, seed = NULL)
```

Arguments

txLoc	A txLoc object; this is the txLoc object from which the subsampled txLoc will be created.
fractions	A numeric vector, specifying the fraction of sites that will be sampled from every transcript region. Note that the length of fractions has to match the length of GetRegions(txLoc).
seed	A single value, interpreted as an integer, or NULL; this is to ensure reproducibility when subsampling txLoc2 sites; default is NULL.

Value

A txLoc object.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

---

txLoc-accessors txLoc *accessors.*

Description

Various txLoc accessors.

Usage

```
## S4 method for signature 'txLoc'
show(object)

## S4 method for signature 'txLoc'
GetLoci(object)

## S4 method for signature 'txLoc'
GetId(object)

## S4 method for signature 'txLoc'
GetRef(object)

## S4 method for signature 'txLoc'
GetVersion(object)

## S4 method for signature 'txLoc'
GetNumberOfLoci(object)

## S4 method for signature 'txLoc'
GetRegions(object)
```

Arguments

object	A txLoc object.

Details

- `show(object)`: Print summary information about the `txLoc` object.
- `GetLoci(object)`: Get a `list` of `DataFrame` objects of the loci of the `txLoc` object.
- `GetId(object)`: Get the identifier string of the `txLoc` object.
- `GetRef(object)`: Get the reference genome string of the `txLoc` object.
- `GetVersion(object)`: Get the version string of a `txLoc` object.
- `GetNumberOfLoci(object)`: Get the number of loci of a `txLoc` object in every transcript section.
- `GetRegions(objects)`: Character vector of the transcript regions of a `txLoc` object.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

`txLoc-class` `txLoc` *object.*

Description

The S4 object stores information about mapped loci per transcript region.

Details

The `txLoc` S4 object contains the following four slots:

1. `loci`: A `list` of `DataFrame` objects, each with the following columns:
   - `locus_in_txx_region`: A `GRanges` object
   - `locus_in_genome`: A `GRanges` object
   - `score`: A `numeric` vector
   - `id`: A `character` vector
   - `tx_region`: A `character` vector
   - `tx_region_width`: An `integer` vector
   - `tx_region_sequence`: A `DNAStringSet` object
   - `tx_refseq`: A `character` vector
   - `gene_entrez`: A `character` vector
   - `gene_symbol`: A `character` vector
   - `gene_ensembl`: A `character` vector
   - `gene_name`: A `character` vector

   The `loci` slot can be accessed using `GetLoci(txLoc)`.
2. `id`: An identifier specified by the user. The `id` slot can be accessed using `GetId(txLoc)`.
3. `refGenome`: The reference genome version (e.g. `"hg38"`), which determines the mapping between genomic and transcriptomic coordinates. The `refGenome` slot can be accessed using `GetRef(txLoc)`.
4. ‘version‘: A version identifier; currently this slot is used to to store the current system time & date. The `version` slot can be accessed using `GetVersion(txLoc)`.

Slots

`loci` A `list` of `DataFrame` objects; see 'Details'.

`id` A `character` string; see 'Details.

`refGenome` A `character` string; see 'Details'.

`version` A `character` string; see 'Details'.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

WriteBED *Write* txLoc *object to a BED file.*

Description

Write `txLoc` object to a BED file. See 'Details'.

Usage

```
WriteBED(txLoc, file = NULL)
```

Arguments

`txLoc`	A `txLoc` object.
`file`	A character string; specifies the filename of the output BED file. If `NULL`, then `file = "sites.bed"`; default is `NULL`.

Details

The function writes entries from a `txLoc` object to a 6-column BED file (BED6). Note that this process is not "splice-aware", i.e. if an entry spans an intron the BED entry gives the left and right-most genomic coordinate of the feature. If `file = NULL`, entries will be written to `sites.bed`.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Examples

```
## Not run:
bedFile <- system.file("extdata",
                       "miCLIP_m6A_Linder2015_hg38.bed",
                       package = "RNAModR")
sites <- ReadBED(bedFile)
txSites <- SmartMap(sites, id = "m6A", refGenome = "hg38")
WriteBED(txSites)

## End(Not run)
```

WriteCSV *Write txLoc object to CSV file.*

Description

Write a txLoc object to a comma-separated-values file.

Usage

```
WriteCSV(txLoc, file = NULL, withSeq = FALSE, withGC = FALSE)
```

Arguments

txLoc	A txLoc object.
file	Filename of output CSV file. If NULL then file = "sites.csv".
withSeq	If TRUE then include full sequence. Default is FALSE.
withGC	If TRUE then include GC content. Default is FALSE. Unused.

Author(s)

Maurits Evers, <maurits.evers@anu.edu.au>

Index