Package 'RNAModR'

June 6, 2016

Type Package

Title Functional Analysis of RNA Modifications

Version 0.1.0

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, beanplot, Biostrings, GenomeInfoDb, gplots, graphics, grDevices, methods, RSQLite, rtracklayer, stats, utils

Depends GenomicRanges, GenomicFeatures, IRanges

Suggests knitr, testthat

License GPL-3 | file LICENSE

LazyData false

RoxygenNote 5.0.1

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)
```

`GenerateNull` *Generate a list of null sites.*

Description

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

Usage

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

Arguments

`locus`	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 generate null distribution; if `method == "ntAbund"` the position of all nucleotides specified by `nt` will be used as null sites; if `method == "perm"` then null sites will be generated by uniform-randomly shuffling candidate positions 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 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.

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 nucleotide position distribution 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 better null distribution would be to consider all cytosines in transcript sections that also contain a site in `locus`. This can be achieved with `method = "ntAbund"`. A still better list of null sites would be based on all *expressed* and non-modified cytosines based on the same sequencing data that was used to identify modified cytosines.

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 junctions.*

Description

Get exon-exon junctions from transcriptome. See 'Details'.

Usage

```
GetEEJunct(refGenome = "hg38", filter = "CDS")
```

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 `"CDS"`.

Details

The function extracts exon-exon junction positions from a reference transcriptome specified by `refGenome`, and returns a `txLoc` object of exon-exon junctions. The position of an exon-exon junction is defined as the position of the 3'-last nucleotide of an exon followed by an intron.

Value

A txLoc object. See 'Details'.

Author(s)

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

GetMFE *Fold sequences.*

Description

Fold sequences. See 'Details'.

Usage

```
GetMFE(data, colSeq, colId = NULL)
```

Arguments

data	A dataframe object. See 'Details'.
colSeq	An integer scalar; specifies the column in data containing the sequences.
colId	An integer scalar; specifies the column in data containing sequence IDs; if NULL IDs are generated automatically; default is NULL.

Details

The function takes a dataframe, extracts sequences from a column specified by colSeq, and predicts secondary structures using RNAfold http://rna.tbi.univie.ac.at/. An optional column containing sequence IDs may be specified by colId. The function returns a dataframe with three columns:

1. Column 1: Sequence ID
2. Column 2: Length of the sequence (in nt)
3. Column 3: Mean free energy (MFE)

Value

A dataframe 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, refGenome = "hg38", filter = c("5UTR", "CDS", "3UTR"),
  maxMM = 0, showPb = TRUE)
```

Arguments

motif	A character vector; specifies the motif(s) that will be matched against the transcriptome.
refGenome	A character string; specifies the reference genome version; default is "hg38".
filter	A character vector; only consider transcript sections specified in filter; default is c("5UTR", "CDS", "3UTR").
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

The function searches for one or multiple motifs within sequences of a reference transcriptome specified by refGenome, and returns a txLoc object of the motif loci within different transcript sections. The maximum number of mismatches allowed in the motif search can be adjusted through maxMM. By default a text progressbar is shown showPb = TRUE. Note that the motif search may take a few minutes, depending on the size of the transcriptome and number of motifs.

Author(s)

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

GetNumberOfLoci *Method "GetNumberOfLoci" for S4 object txLoc.*

Description

Get the number of loci of a txLoc object in every transcript section.

Usage

```
GetNumberOfLoci(x)

## S4 method for signature txLoc
GetNumberOfLoci(x)
```

Arguments

x	A txLoc object.

Value

A named integer vector with the number of sites per transcript section.

Author(s)

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

Info *Method "Info" for S4 object txLoc.*

Description

Print summary information about txLoc object.

Usage

```
Info(x)

## S4 method for signature txLoc
Info(x)
```

Arguments

x	A txLoc object.

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");
Info(posSites);

## End(Not run)
```

PlotGC *Plot GC content.*

Description

Plot and assess GC content distributions from two site lists. See 'Details'.

Usage

```
PlotGC(loc1, loc2, flank = 10, filter = NULL, geneNorm = FALSE,
  subsample = TRUE)
```

Arguments

loc1	A txLoc object.
loc2	A txLoc object.
flank	An integer scalar; see 'Details'.
filter	A logical scalar; only consider loci in transcript regions specified in filter; Default is NULL.
geneNorm	A logical scalar; if TRUE normalise GC content in window to GC content of transcript section; default is FALSE.
subsample	A logical scalar; if TRUE a subsample of loc2 is used instead of the full set; the subsample size is dynamically determined based on the total number of sites in loc1; default is TRUE.

Details

The function calculates the GC content within a region around every site from two txLoc objects. The window is defined by extending the position of every transcript locus upstream and downstream by flank nucleotides (if possible). The means of the resulting GC content distributions are assessed using a two-tailed t-test. If geneNorm = TRUE, the site GC content is normalised to the GC content of the entire transcript section. If subsample = TRUE, only a subsample of entries from the *second* txLoc object will be used. This is useful (and therefore the default), as loc2 usually refers to the much larger list of null sites.

Author(s)

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

PlotOverlap *Plot overlap of sites.*

Description

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

Usage

```
PlotOverlap(loc1, loc2)
```

Arguments

loc1	A txLoc object.
loc2	A txLoc object.

Details

The function plots one or multiple Venn diagrams denoting 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.

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(loc1, loc2, flank = 1000, binWidth = 20,
  doBootstrap = TRUE)
```

Arguments

loc1	A txLoc object.
loc2	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 the minimum distance between entries from two txLoc objects located within the same transcript section. Relative distances are shown within a window (-flank, flank), where negative distances correspond to an upstream feature from loc1 relative to loc2, and positive distances to a downstream feature from loc1 relative to loc2. Relative distances are binned in bins of binWidth nt, and shown as an abundance histogram. If doBootstrap = TRUE, 95 calculated and shown, based on an empirical bootstrap of relative distances.

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(locPos, locNeg, locRef, flank = 1000, binWidth = 20)
```

Arguments

locPos	A txLoc object.
locNeg	A txLoc object.
locRef	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 between entries from locPos relative to locRef, and locNeg relative to locRef. 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 correspond to an enrichment/depletion of sites from locPos relative to sites from locNeg upstream of the closest site from locRef. Positive distances correspond to a downstream enrichment/depletion relative to sites from locRef. The bin width and window size can be adjusted with flank and binWidth.

Author(s)

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

PlotSectionDistribution
Plot piechart of the number of loci in every transcript section.

Description

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

Usage

```
PlotSectionDistribution(locus, filter = NULL)
```

Arguments

locus	A txLoc object.
filter	A character vector; only consider transcript sections specified in filter; if NULL consider all sections.

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 transcript section enrichment analysis and plot results.*

Description

Perform transcript section enrichment analysis 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(locPos, locNeg, filter = NULL,
  withExtendedAxisLabel = 2)
```

Arguments

locPos	A txLoc object. These should be the positive control sites.
locNeg	A txLoc object. These should be the negative control sites.
filter	Only plot loci in transcript regions specified in filter. Default is NULL.
withExtendedAxisLabel	Plot extended axis labels. Default is 2. See ??? for 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 = "nuclAbundance");
PlotSectionEnrichment(posSites, negSites,
                      filter = c("5UTR", "CDS", "3UTR"));

## End(Not run)
```

`PlotSeqLogo` *Plot sequence logo.*

Description

Plot sequence logo.

Usage

```
PlotSeqLogo(locus, flank = 5, filter = NULL, ylim = c(0, 2))
```

Arguments

`locus`	A `txLoc` object.
`flank`	An integer scalar; see 'Details'.
`filter`	A character vector; only plot sequence logos of sections specified in `filter`; if `NULL` plot all sections; default is NULL.
`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 `locus` upstream and downstream by `flank` nucleotides. By default logos are shown for every transcript section from `locus`. Use `filter` to specify specific transcript sections.

Author(s)

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

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

Description

Plot spatial distribution of loci from txLoc object within every transcript section.

Usage

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

Arguments

locus	A txLoc object.
filter	Only plot loci in transcript regions specified in filter.
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.

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("5UTR", "CDS", "3UTR"),
                        ylim = c(0, 200));

## End(Not run)
```

PlotSpatialEnrichment *Perform spatial enrichment analysis and plot results.*

Description

Perform spatial enrichment analysis 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(locPos, locNeg, filter = NULL, binWidth = 20,
  posMax = 1000)
```

Arguments

locPos	A txLoc object. These should be the positive control sites.
locNeg	A txLoc object. These should be the negative control sites.
filter	Only plot loci in transcript regions specified in filter.
binWidth	Spatial bin width. Default is 20 nt.
posMax	Evaluate enrichment within a window given by posMax. Default is 1000 nt.

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 = "permutation");
PlotSpatialEnrichment(posSites, negSites,
                      filter = c("5UTR", "CDS", "3UTR"));

## 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; if TRUE loci spanning more than one nucleotide are 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.

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);
```

ReadDBN *Read a DBN file.*

Description

Read a DBN file. See 'Details'.

Usage

```
ReadDBN(file)
```

Arguments

file	A character string; specifies the input DBN file.

Details

The function reads in a DBN (dot-bracket) structure file, and returns a dataframe with the following data columns:

1. Column 1: Sequence ID
2. Column 2: Length of the sequence (in nt)
3. Column 3: Mean free energy (MFE)

Value

A dataframe object. See 'Details'.

Author(s)

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

SmartMap *Map genome coordinates to transcript coordinates.*

Description

Map genome coordinates to transcript coordinates.

Usage

```
SmartMap(locus, id = NULL, refGenome = "hg38", ignore.strand = FALSE)
```

Arguments

locus	A GRanges object; specifies the list of of genomic features to be mapped.
id	A character string; specifies a name for loci from locus; 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.

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 thrown 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)
```

---

txLoc-class *txLoc object. The S4 object stores information about mapped loci per transcript section. Meta-data such as an identifier and the reference genome are stored in separate slots.*

Description

txLoc object. The S4 object stores information about mapped loci per transcript section. Meta-data such as an identifier and the reference genome are stored in separate slots.

Slots

`loci` A `list` of `dataframe` objects; specifies the list of loci per transcript section.

`id` A character string; identifier for loci in `txLoc` object.

`refGenome` A character string; gives the reference genome upon which transcriptome-derived positions are based.

`version` A character string; can be used to flag specific version, e.g. using the current system time & date.

Author(s)

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

WriteTxLocToBED *Write* `txLoc` *object to a BED file.*

Description

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

Usage

```
WriteTxLocToBED(locus, file = NULL, noChrName = FALSE)
```

Arguments

locus	A txLoc object.
file	A character string; specifies the filename of the output BED file. If NULL, then file = "sites.bed"; default is NULL.
noChrName	A logical scalar; if TRUE, chromosome names will be written without "chr"; default is FALSE.

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. If noChrName = TRUE, chromosome names in column 1 of the BED file will be written without "chr".

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");
WriteTxLocToBED(txSites);

## End(Not run)
```

Index