Package 'RiboCrypt'

March 17, 2023

Type Package

Title Interactive visualization in genomics

Version 1.5.0

License MIT + file LICENSE

Description R Package for interactive visualization and browsing NGS data.
It contains a browser for both transcript and genomic coordinate view.
In addition a QC and general metaplots are included, among others differential translation plots and gene expression plots. The package is still under development.

biocViews Software, Sequencing, RiboSeq, RNASeq,

Encoding UTF-8

LazyData true

BugReports https://github.com/m-swirski/RiboCrypt/issues

URL https://github.com/m-swirski/RiboCrypt

Depends R (>= 3.6.0), ORFik (>= 1.13.12)

Imports BiocGenerics, BiocParallel, Biostrings, data.table, dplyr, GenomeInfoDb, GenomicFeatures, GenomicRanges, ggplot2, IRanges, plotly, rlang

Suggests testthat, rmarkdown, knitr, BiocStyle, BSgenome, BSgenome.Hsapiens.UCSC.hg19

RoxygenNote 7.1.2

VignetteBuilder knitr

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

git_branch master

git_last_commit 4766276

git_last_commit_date 2022-11-01

Date/Publication 2023-03-17

Author Michal Swirski [aut, cre], Haakon Tjeldnes [ctb]

Maintainer Michal Swirski <michal.swirski@uw.edu.pl>

R topics documented:

`antisense` *Get antisense*

Description

Get antisense

Usage

```
antisense(grl)
```

Value

a GRangesList

`createSeqPanel` *Create sequence panel for RiboCrypt*

Description

Create sequence panel for RiboCrypt

Usage

```
createSeqPanel(
  sequence,
  start_codons = "ATG",
  stop_codons = c("TAA", "TAG", "TGA"),
  frame = 1,
  custom_motif = NULL
)
```

Arguments

start_codons	character vector, default "ATG"
stop_codons	character vector, default c("TAA", "TAG", "TGA")
custom_motif	character vector, default NULL.

Value

a ggplot object

---

getCoverageProfile *Get coverage profile*

Description

Get coverage profile

Usage

```
getCoverageProfile(grl, reads, kmers = 1, kmers_type = "mean")
```

Arguments

grl	a GRangesList
reads	GRanges
kmers	1
kmers_type	"mean"

Value

data.table of coverage

---

getIndexes *Get index*

Description

Get index

Usage

```
getIndexes(ref_granges)
```

Arguments

ref_granges	a GRanges object

Value

integer vector, indices

---

ggplotlyHover *Call ggplotly with hoveron defined*

Description

Call ggplotly with hoveron defined

Usage

```
ggplotlyHover(x, ...)
```

Arguments

x	a a ggplot argument
...	additional arguments for ggplotly

Value

a ggplotly object

---

matchMultiplePatterns *Match multiple patterns*

Description

Match multiple patterns

Usage

```
matchMultiplePatterns(patterns, Seq)
```

Arguments

patterns	character
Seq	a DNAStringSet

Value

integer vector, indices (named with pattern hit)

`matchToGRanges` *Match to GRanges*

Description

Match to GRanges

Usage

```
matchToGRanges(matches, ref_granges)
```

Arguments

`matches`	integer vector, indices
`ref_granges`	GRanges

Value

GRanges object

`multiOmicsPlot_animate` *Multi-omics animation using list input*

Description

The animation will move with a play butten, there is 1 transition per library given.

Usage

```
multiOmicsPlot_animate(
  target_range,
  annotation = target_range,
  reference_sequence,
  reads,
  withFrames = NULL,
  colors = NULL,
  kmers = NULL,
  kmers_type = c("mean", "sum")[1],
  ylabels = NULL,
  proportions = NULL,
  width = NULL,
  height = NULL,
  plot_name = "default",
  plot_title = NULL,
```

```
  display_sequence = FALSE,
  annotation_names = NULL,
  start_codons = "ATG",
  stop_codons = c("TAA", "TAG", "TGA"),
  custom_motif = NULL
)
```

Arguments

target_range	the whole region to visualize, a GRangesList or GRanges object
annotation	the whole annotation which your target region is a subset, a GRangesList or GRanges object
reference_sequence	the genome reference, a FaFile or FaFile convertible object
reads	the NGS libraries, as a list of GRanges with or without score column for replicates.
withFrames	a logical vector, default NULL. Alternative: a length 1 or same length as list length of "reads" argument.
colors	character, default NULL (automatic colouring). If "withFrames" argument is TRUE, colors are set to to c("red", "green", "blue") for the 3 frames. Alternative: Character vector of length 1 or length of "reads" list argument.
kmers	numeric (integer), bin positions into kmers.
kmers_type	character, function used for kmers sliding window. default: "mean", alternative: "sum"
ylabels	character, default NULL. Name of libraries in "reads" list arugment.
proportions	numeric, default NULL. Width of plot.
width	numeric, default NULL. Width of plot.
height	numeric, default NULL. Height of plot.
plot_name	= character, default "default" (will create name from target_range name). Alternative: custom name for region.
plot_title	character, default NULL. A title for plot.
display_sequence	logical, default FALSE. If TRUE, display nucleotide sequence in plot.
annotation_names	character, default NULL. Alternative naming for annotation.
start_codons	character vector, default "ATG"
stop_codons	character vector, default c("TAA", "TAG", "TGA")
custom_motif	character vector, default NULL.

Value

the plot object

Examples

```
library(ORFik)
df <- ORFik.template.experiment()[3,] #Use third library in experiment only
if (requireNamespace("BSgenome.Hsapiens.UCSC.hg19")) {
  cds <- loadRegion(df, "cds")
  multiOmicsPlot_ORFikExp(extendLeaders(extendTrailers(cds[1], 30), 30), df = df,
                          reference_sequence = BSgenome.Hsapiens.UCSC.hg19::Hsapiens,
                          frames_type = "columns")
}
```

`multiOmicsPlot_list` *Multi-omics plot using list input*

Description

Customizable html plots for visualizing genomic data.

Usage

```
multiOmicsPlot_list(
  target_range,
  annotation = target_range,
  reference_sequence,
  reads,
  withFrames = NULL,
  frames_type = "lines",
  colors = NULL,
  kmers = NULL,
  kmers_type = c("mean", "sum")[1],
  ylabels = NULL,
  proportions = NULL,
  width = NULL,
  height = NULL,
  plot_name = "default",
  plot_title = NULL,
  display_sequence = FALSE,
  annotation_names = NULL,
  start_codons = "ATG",
  stop_codons = c("TAA", "TAG", "TGA"),
  custom_motif = NULL,
  BPPARAM = bpparam()
)
```

Arguments

`target_range` the whole region to visualize, a `GRangesList` or `GRanges` object

annotation — the whole annotation which your target region is a subset, a GRangesList or GRanges object

reference_sequence — the genome reference, a FaFile or FaFile convertible object

reads — the NGS libraries, as a list of GRanges with or without score column for replicates.

withFrames — a logical vector, default NULL. Alternative: a length 1 or same length as list length of "reads" argument.

frames_type — character, default "lines". Alternative:
- columns
- stacks
- area

colors — character, default NULL (automatic colouring). If "withFrames" argument is TRUE, colors are set to to c("red", "green", "blue") for the 3 frames. Alternative: Character vector of length 1 or length of "reads" list argument.

kmers — numeric (integer), bin positions into kmers.

kmers_type — character, function used for kmers sliding window. default: "mean", alternative: "sum"

ylabels — character, default NULL. Name of libraries in "reads" list arugment.

proportions — numeric, default NULL. Width of plot.

width — numeric, default NULL. Width of plot.

height — numeric, default NULL. Height of plot.

plot_name — = character, default "default" (will create name from target_range name). Alternative: custom name for region.

plot_title — character, default NULL. A title for plot.

display_sequence — logical, default FALSE. If TRUE, display nucleotide sequence in plot.

annotation_names — character, default NULL. Alternative naming for annotation.

start_codons — character vector, default "ATG"

stop_codons — character vector, default c("TAA", "TAG", "TGA")

custom_motif — character vector, default NULL.

BPPARAM — how many cores/threads to use? default: `BiocParallel::bpparam()`. To see number of threads used, do `BiocParallel::bpparam()$workers`. You can also add a time remaining bar, for a more detailed pipeline.

Value

the plot object

Examples

```
library(ORFik)
df <- ORFik.template.experiment()[3,] #Use third library in experiment only
if (requireNamespace("BSgenome.Hsapiens.UCSC.hg19")) {
  cds <- loadRegion(df, "cds")
  multiOmicsPlot_ORFikExp(extendLeaders(extendTrailers(cds[1], 30), 30), df = df,
                          reference_sequence = BSgenome.Hsapiens.UCSC.hg19::Hsapiens,
                          frames_type = "columns")
}
```

`multiOmicsPlot_ORFikExp` *Multi-omics plot using ORFik experiment input*

Description

Customizable html plots for visualizing genomic data.

Usage

```
multiOmicsPlot_ORFikExp(
  target_range,
  annotation = target_range,
  df,
  reference_sequence = findFa(df),
 reads = outputLibs(df, type = "pshifted", output.mode = "envirlist", naming = "full"),
 withFrames = libraryTypes(df, uniqueTypes = FALSE) %in% c("RFP", "RPF", "LSU"),
  frames_type = "lines",
  colors = NULL,
  kmers = NULL,
  kmers_type = c("mean", "sum")[1],
  ylabels = bamVarName(df),
  proportions = NULL,
  width = NULL,
  height = NULL,
  plot_name = "default",
  plot_title = NULL,
  display_sequence = FALSE,
  annotation_names = NULL,
  start_codons = "ATG",
  stop_codons = c("TAA", "TAG", "TGA"),
  custom_motif = NULL,
  BPPARAM = bpparam()
)
```

Arguments

target_range	the whole region to visualize, a GRangesList or GRanges object
annotation	the whole annotation which your target region is a subset, a GRangesList or GRanges object
df	an ORFik experiment or a list containing ORFik experiments. Usually a list when you have split Ribo-seq and RNA-seq etc.
reference_sequence	the genome reference, default ORFik::findFa(df)
reads	the NGS libraries, as a list of GRanges with or without score column for replicates. Default: outputLibs(df, type = "pshifted", output.mode = "envirlist", naming = "full")
withFrames	a logical vector, default libraryTypes(df, uniqueTypes = FALSE) %in% c("RFP", "RPF", "LSU") Alternative: a length 1 or same length as list length of "reads" argument.
frames_type	character, default "lines". Alternative: - columns - stacks - area
colors	character, default NULL (automatic colouring). If "withFrames" argument is TRUE, colors are set to to c("red", "green", "blue") for the 3 frames. Alternative: Character vector of length 1 or length of "reads" list argument.
kmers	numeric (integer), bin positions into kmers.
kmers_type	character, function used for kmers sliding window. default: "mean", alternative: "sum"
ylabels	character, default bamVarName(df). Name of libraries in "reads" list argument.
proportions	numeric, default NULL. Width of plot.
width	numeric, default NULL. Width of plot.
height	numeric, default NULL. Height of plot.
plot_name	character, default "default" (will create name from target_range name). Alternative: custom name for region.
plot_title	character, default NULL. A title for plot.
display_sequence	logical, default FALSE. If TRUE, display nucleotide sequence in plot.
annotation_names	character, default NULL. Alternative naming for annotation.
start_codons	character vector, default "ATG"
stop_codons	character vector, default c("TAA", "TAG", "TGA")
custom_motif	character vector, default NULL.
BPPARAM	how many cores/threads to use? default: BiocParallel::bpparam(). To see number of threads used, do BiocParallel::bpparam()$workers. You can also add a time remaining bar, for a more detailed pipeline.

Value

the plot object

Examples

```
library(ORFik)
df <- ORFik.template.experiment()[3,] #Use third library in experiment only
if (requireNamespace("BSgenome.Hsapiens.UCSC.hg19")) {
  cds <- loadRegion(df, "cds")
  multiOmicsPlot_ORFikExp(extendLeaders(extendTrailers(cds[1], 30), 30), df = df,
                          reference_sequence = BSgenome.Hsapiens.UCSC.hg19::Hsapiens,
                          frames_type = "columns")
}
```

RiboCrypt.template.experiment

An ORFik experiment to see how it looks

Description

Toy-data created to resemble human genes:
Number of genes: 6
Ribo-seq: 2 libraries RNA-seq: 2 libraries CAGE: 1 library PAS (poly-A): 1 library

Usage

```
RiboCrypt.template.experiment(as.temp = FALSE)
```

Arguments

as.temp	logical, default FALSE, load as ORFik experiment. If TRUE, loads as data.frame template of the experiment.

Value

an ORFik experiment

Examples

```
ORFik.template.experiment()
```

`trimOverlaps` *Trim overlaps*

Description

Trim overlaps

Usage

```
trimOverlaps(overlaps, target_range)
```

Arguments

`overlaps`	GRanges
`target_range`	GRanges

Value

GRanges

Index