Package 'CircSeqAlignTk'

January 29, 2023

Type Package

Title A toolkit for end-to-end analysis of RNA-seq data for circular genomes

Version 1.0.0

Description CircSeqAlignTk is designed for end-to-end RNA-Seq data analysis of circular genome sequences, from alignment to visualization. It mainly targets viroids which are composed of 246-401 nt circular RNAs. In addition, CircSeqAlignTk implements a tidy interface to generate synthetic sequencing data that mimic real RNA-Seq data, allowing developers to evaluate the performance of alignment tools and workflows.

Depends R (>= 4.2)

Imports stats, tools, utils, methods, S4Vectors, rlang, magrittr, dplyr, tidyr, ggplot2, BiocGenerics, Biostrings, IRanges, ShortRead, Rsamtools, Rbowtie2, Rhisat2

Suggests knitr, rmarkdown, testthat, R.utils, BiocStyle

VignetteBuilder knitr

biocViews Sequencing, SmallRNA, Alignment, Software

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.1

URL https://github.com/jsun/CircSeqAlignTk

BugReports https://github.com/jsun/CircSeqAlignTk/issues

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

git_branch RELEASE_3_16

git_last_commit 0166720

git_last_commit_date 2022-11-01

Date/Publication 2023-01-29

Author Jianqiang Sun [aut, cre] (<https://orcid.org/0000-0002-3438-3199>),
Xi Fu [aut],
Wei Cao [aut]

Maintainer Jianqiang Sun <sun@biunit.dev>

R topics documented:

`CircSeqAlignTk-package` *CircSeqAlignTk: A toolkit for end-to-end analysis of RNA-seq data for circular genomes*

Description

CircSeqAlignTk is designed for end-to-end RNA-Seq data analysis of circular genome sequences, from alignment to visualization. It mainly targets viroids which are composed of 246-401 nt circular RNAs. In addition, CircSeqAlignTk implements a tidy interface to generate synthetic sequencing data that mimic real RNA-Seq data, allowing developers to evaluate the performance of alignment tools and workflows.

Details

Refer to the vignette for an overview of the package, quick start, and detailed usages.

Author(s)

Maintainer: Jianqiang Sun <sun@biunit.dev> (ORCID)

Authors:

- Xi Fu
- Wei Cao

See Also

Useful links:

- https://github.com/jsun/CircSeqAlignTk
- Report bugs at https://github.com/jsun/CircSeqAlignTk/issues

Examples

```
browseVignettes("CircSeqAlignTk")
```

`align_reads` *Align sequence reads to a genome sequence*

Description

This function aligns sequence reads in a FASTQ file to the reference sequences of a genome.

Usage

```
align_reads(
  input,
  index,
  output,
  n_threads = 1,
  n_mismatch = 1,
  overwrite = TRUE,
  aligner = c("bowtie2", "hisat2"),
  add_args = NULL
)
```

Arguments

`input`	A path to a FASTQ format file for alignment.
`index`	A `CircSeqAlignTkRefIndex-class` object generated by the `build_index` function.
`output`	A path to a directory for saving the intermediate and final results of alignment.
`n_threads`	Number of threads to use for aligning reads.
`n_mismatch`	Number of allowed mismatches in alignment.
`overwrite`	Overwrite the existing files if `TRUE`.
`aligner`	A string to specify the alignment is for alignment.
`add_args`	A string of additional arguments to be passed on to the alignment tool directly. For example, `-N 0 -L 22`, `--no-spliced-alignment -k 10`, etc.

Details

This function aligns sequence reads in a FASTQ format file in two stages: (i) aligning reads to the type 1 reference sequence (i.e., `refseq.t1.fa`) and (ii) collecting the unaligned reads and aligning them with the type 2 reference (i.e., `refseq.t2.fa`). The alignment results are saved as BAM format files in the specified directory with the suffixes `*.t1.bam` and `*.t2.bam`. The original alignment results may contain mismatches. Hence, filtering is performed to remove the alignment with mismatches over the specified value from the BAM format file. The filtered results of the `*.t1.bam` and `*.t2.bam` are saved as `*.clean.t1.bam` and `*.clean.t2.bam`, respectively.

Two alignment tools (Bowtie2 and HISAT2) can be specified for building indexes through the aligner argument. This function first attempts to call the specified alignment tool installed on the operation system directly; however, if the tool is not installed, then the function attempts to call `bowtie2_build` or `hisat2_build` functions implemented in the Rbowtie2 or Rhisat2 packages for alignment.

Value

A `CircSeqAlignTkAlign-class` object.

See Also

`CircSeqAlignTkAlign-class`

Examples

```
output_dpath <- tempdir()

genome_seq <- system.file(package="CircSeqAlignTk", "extdata", "FR851463.fa")
fq <- system.file(package="CircSeqAlignTk", "extdata", "srna.fq.gz")

ref_index <- build_index(input = genome_seq,
                         output = file.path(output_dpath, 'index'))
aln <- align_reads(input = fq, index = ref_index,
                   output = file.path(output_dpath, 'align_results'))

slot(aln, 'stats')
```

`build_index` *Build indexes of reference sequences for alignment*

Description

This function internally calls Bowtie2 or HISAT2 to build indexes of reference sequences for alignment preparation.

Usage

```
build_index(
  input,
  output = NULL,
  n_threads = 1,
  overwrite = TRUE,
  aligner = c("bowtie2", "hisat2"),
  add_args = NULL
)
```

Arguments

input	A path to a FASTA format file containing a reference sequence of a genome for indexing.
output	A path to a directory for saving the reference sequences and indexes.
n_threads	Number of threads to use for aligning reads.
overwrite	Overwrite the existing files if TRUE.
aligner	A string to specify the alignment for indexing.
add_args	A string of additional arguments to be passed on to the alignment tool directly (e.g., --quiet).

Details

This function generates two types of reference sequences from a genome and indexes them in preparation for alignment. The type 1 reference sequence is identical to the sequence provided by the input argument. The type 2 reference sequence is generated by restoring the type 1 reference sequence to a circular RNA and opening the circle at the position opposite to that of type 1. The type 1 and 2 reference sequences are then saved as FASTA format files, refseq.t1.fa and refseq.t2.fa, respectively, under the directory specified by the output argument. Next, the function builds indexes for refseq.t1.fa and refseq.t2.fa.

Two alignment tools (Bowtie2 and HISAT2) can be specified for building indexes through the aligner argument. This function first attempts to call the specified alignment tool installed on the operation system directly; however, if the tool is not installed, then the function attempts to call bowtie2_build or hisat2_build functions implemented in the Rbowtie2 or Rhisat2 packages for indexing.

Value

A CircSeqAlignTkRefIndex-class object.

See Also

CircSeqAlignTkRefIndex-class

Examples

```
output_dpath <- tempdir()

genome_seq <- system.file(package="CircSeqAlignTk", "extdata", "FR851463.fa")
ref_index <- build_index(input = genome_seq,
                         output = file.path(output_dpath, "index"))
```

`calc_coverage` *Calculate alignment coverage*

Description

This function calculates alignment coverage according to the read strand and length from alignment results.

Usage

```
calc_coverage(x)
```

Arguments

x	A CircSeqAlignTkAlign-class object generated by the align_reads function.

Details

This function calculates alignment coverage from the two BAM files, `*.clean.t1.bam` and `*.clean.t2.bam`, generated by the align_reads function. The coverage is then sorted by the strand and length of the aligned reads and summarized into data frames.

Value

A CircSeqAlignTkCoverage-class object.

See Also

CircSeqAlignTkAlign-class, CircSeqAlignTkCoverage-class, align_reads

Examples

```
output_dpath <- tempdir()
genome_seq <- system.file(package="CircSeqAlignTk", "extdata", "FR851463.fa")
fq <- system.file(package="CircSeqAlignTk", "extdata", "srna.fq.gz")
ref_index <- build_index(input = genome_seq,
                         output = file.path(output_dpath, 'index'))
aln <- align_reads(input = fq, index = ref_index,
                   output = file.path(output_dpath, 'align_results'))

alncov <- calc_coverage(aln)
```

CircSeqAlignTkAlign-class
Class to store alignment results

Description

A class to store alignment results, including the paths to FASTQ and BAM format files and the alignment summary. The object belongs to this class is generated by `align_reads` function.

Slots

`input_fastq` A path to the query FASTQ format file.

`fastq` A vector containing the paths to the two FASTQ format files used for alignment to the type 1 and type 2 references, respectively. See `align_reads` for how the FASTQ format files are generated.

`bam` A vector containing the paths to the two BAM format files corresponding to the alignment results of the two FASTQ files shown in the `fastq` slot, respectively.

`clean_bam` A vector containing the paths to the two BAM format files after filtering by number of mismatch from BAM format files shown in `bam` slot.

`stats` A data frame containing alignment summary, e.g., number of query reads, aligned reads, and unaligned reads.

`reference` A `CircSeqAlignTkRefIndex-class` storing the information of reference for alignment.

See Also

`CircSeqAlignTkRefIndex-class`, `align_reads`

CircSeqAlignTkCoverage-class
Class to save alignment coverage

Description

A class to store the alignment coverage generated by `calc_coverage` function.

Slots

`forward` A matrix containing the alignment coverage of the forward strand reads.

`reversed` A matrix containing the alignment coverage of the reversed strand reads.

`.figdata` A string of adapter sequence.

See Also

`calc_coverage`

CircSeqAlignTkRefIndex-class

Class to store reference information

Description

A class to store reference information for alignment. The object belongs to this class is generated by `build_index` function.

Slots

`name` Reference name. The sequence name written the header of FASTA format file.

`seq` Reference sequence.

`length` Length of the reference sequence.

`fasta` A vector containing the paths to the two FASTA format files of the type 1 and type 2 reference sequences, respectively. See `build_index` for how FASTA format files are generated.

`index` A vector containing the paths to the two reference indexes corresponding to the two FASTA format files stored in the `fasta` slot, respectively.

`cut_loc` The position on the user-given sequence (i.e., the type 1 sequence) to cut for generating the type 2 reference sequence.

See Also

`build_index`

CircSeqAlignTkSim-class

Class to save information of synthetic reads

Description

A class to store parameters for generating synthetic sequence reads. The object belongs to this class is generated by `generate_reads` function.

Slots

`seq` A string of a genome sequence, which is used for sampling synthetic sequence reads.

`adapter` A string of adapter sequence.

`read_info` A data frame storing the summary information of read generation. It contains the start and end positions of sampling, strand, and nucleotide sequence of each synthetic read.

`peak` A data frame storing the peaks information of alignment coverage.

`coverage` A `CircSeqAlignTkCoverage-class` storing the information of alignment coverage.

`fastq` A path to FASTQ format file saving the synthetic reads.

See Also

generate_reads, CircSeqAlignTkCoverage-class

---

filter_reads *Filter sequence reads in a FASTQ file by length*

Description

This function removes sequence reads with lengths outside the specified range from the FASTQ file.

Usage

```
filter_reads(input, output, read_lengths = seq(21, 24), overwrite = TRUE)
```

Arguments

input	A path to a FASTQ file targeted for filtering.
output	A path to save the filtered reads in FASTQ format.
read_lengths	A series of integers to specify read length. Reads other than the length specified will be excluded during alignment.
overwrite	Overwrite the existing files if TRUE.

Details

Studies on small RNA-seq data from viroid-infected plants have mostly focused on reads with lengths ranging from 21 nt to 24 nt. This function is intended to be used to remove sequence reads with lengths outside the specified range. The default range is 21-24 nt, which can be changed through the read_lengths argument.

Note that, if filtering by read length has already been performed during the quality control process, there is no need to use this function.

Value

A path to the filtered FASTQ file.

Examples

```
output_dpath <- tempdir()

fq <- system.file(package="CircSeqAlignTk", "extdata", "srna.fq.gz")
output_fq <- file.path(output_dpath, "sran.filtered.fq.gz")
filter_reads(fq, output_fq, seq(21, 24))
```

`generate_reads` *Generate synthetic sequence reads*

Description

This function generates synthetic sequence reads to mimic RNA-seq reads sequenced from organelles or organisms with circular genome sequences in FASTQ format file.

Usage

```
generate_reads(
  n = 10000,
  seq = NULL,
  output = NULL,
  adapter = NULL,
  srna_length = NULL,
  read_length = 150,
  mismatch_prob = 0,
  peaks = NULL,
  read_name_prefix = NULL
)
```

Arguments

`n`	Number of reads should be generated.
`seq`	A file path to a genome sequence in FASTA format file or a string of genome sequence.
`output`	A file path to store the synthetic reads in FASTQ format file. The extension should be one of `.fq`, `.fastq`. Note that to compress the FASTQ format file, add `.gz` or `.gzip` to the extension (e.g., `.fq.gz`, `.fq.gzip`).
`adapter`	A path to a FASTA format file containing a string of adapter sequence. If `NULL` is specified, the sequence "AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC" is used as the adapter sequence. If `NA` is specified, the adapter sequence is not included in the synthetic reads.
`srna_length`	A data frame to specify the lengths of sequence reads sampled from the genome sequence. The data frame should contain two columns named as `length` and `prob`. The values in the `length` column is used to specify the lengths of sequence reads; the values in the `prob` column is used to specify the probability that reads with specified length among all reads. If the argument is not given (i.e., `srna_length = NULL`), a data frame is randomly generated before sampling the reads.
`read_length`	The length of synthetic reads. If `adapter` is specified, the reads are generated by concatenating sequence reads and adapter sequences until the specified length. If `adapter = None`, ignore this argument.

`mismatch_prob` A vector to specify probabilities of mismatches occurring in the reads. In order not to allow any mismatches in the reads, set the argument to `0`. To allow multiple mismatches in the reads, set multiple probabilities (e.g., `c(0.05, 0.01)`).

`peaks` A data frame to specify the peaks of the alignment coverage. The data frame should contain four columns named as `mean`, `std`, `strand`, and `prob`. The values in the `mean` and `std` columns are used to sample the start position of sequence reads from the genome sequence given by `seq`. The values in the `strand` column should be `+` or `-` to specify which read strand generates the peak. The values in the `prob` column should be probabilities to use the `mean`, `std`, and `strand` of the same row for read generation. If the argument is not given (i.e., `peaks = NULL`), a data frame is randomly generated before sampling the reads.

`read_name_prefix` The prefix of read name in FASTQ format file. If `NULL`, generate the prefix randomly.

Value

A `CircSeqAlignTkSim-class` object containing parameters for read generation.

See Also

`CircSeqAlignTkSim-class`

Examples

```
output_dpath <- tempdir()

sim <- generate_reads(output = file.path(output_dpath, 'sample1.fq.gz'))

srna_length <- data.frame(length = c(21, 22, 23, 24),
                          prob = c(0.5, 0.3, 0.1, 0.1))
sim <- generate_reads(output = file.path(output_dpath, 'sample2.fq.gz'),
                      srna_length = srna_length)


sim <- generate_reads(output = file.path(output_dpath, 'sample3.fq.gz'),
                      mismatch_prob = c(0.1, 0.1))


peaks <- data.frame(mean =   c( 50, 100, 150),
                    std =    c(  3,   5,   5),
                    strand = c('+', '-', '+'),
                    prob =   c(0.4, 0.4, 0.2))
sim <- generate_reads(output = file.path(output_dpath, 'sample4.fq.gz'),
                      peaks = peaks)
```

get_slot_contents *Get the slot contents from a formal class*

Description

This function returns the slot contents from a formal class. It is convenient to use `@` when accessing the contents of a slot, however, using `@` will generate warnings during the unit tests under software development. This function was created to avoid that warning. Users do not have to use this function.

Usage

```
get_slot_contents(object, name)
```

Arguments

`object`	An object from a formally defined class.
`name`	The name of the slot.

Value

The contents of the specified slot from the given object.

Examples

```
output_dpath <- tempdir()
sim <- generate_reads(output = file.path(output_dpath, 'sample1.fq.gz'))
head(get_slot_contents(sim, 'peak'))
```

merge.CircSeqAlignTkSim *Merge multiple synthetic datasets*

Description

Merge multiple synthetic datasets generated by `generate_reads`.

Usage

```
## S3 method for class 'CircSeqAlignTkSim'
merge(..., output = NULL, overwrite = TRUE)
```

Arguments

...	CircSeqAlignTkSim class objects.
output	A file path to store the synthetic reads in FASTQ format file. The extension should be one of .fq, .fastq. Note that to compress the FASTQ format file, add .gz or .gzip to the extension (e.g., .fq.gz, .fq.gzip).
overwrite	Overwrite the existing files if TRUE.

Details

Merge multiple synthetic datasets generated by generate_reads into one dataset.

Value

A CircSeqAlignTkSim-class object.

See Also

CircSeqAlignTkSim-class, generate_reads

Examples

```
output_dpath <- tempdir()

sim_params_1 <- data.frame(length = c(21, 22), prob = c(0.5, 0.4))
sim_1 <- generate_reads(n = 5e2,
                        output = file.path(output_dpath, 'sample1.fq.gz'),
                        srna_length = sim_params_1)

sim_params_2 <- data.frame(length = c(19, 20, 23), prob = c(0.2, 0.7, 0.1))
sim_2 <- generate_reads(n = 5e2,
                        output = file.path(output_dpath, 'sample2.fq.gz'),
                        srna_length = sim_params_2)

sim <- merge(sim_1, sim_2, output = file.path(output_dpath, 'sample.fq.gz'))
```

---

plot_coverage *Visualize alignment coverage*

Description

This function visualizes the alignment coverage using an area chart. By default, the upper and lower directions of the y-axis represent the alignment coverage of the reads aligned in the forward and reversed strands, respectively.

Usage

```
plot_coverage(x, read_lengths = NULL, fill = "read_length", scale_fun = NULL)

## S3 method for class 'CircSeqAlignTkCoverage'
plot(x, ...)
```

Arguments

x	A CircSeqAlignTkCoverage-class object generated by the calc_coverage function.
read_lengths	Numeric numbers to specify the lengths of reads targeted for visualization. If NULL (default), plot the alignment coverage of reads with all lengths.
fill	Specify NULL or read_length. If read_length is specified, then color the area chart according to the read length.
scale_fun	Set log10 or log to plot the alignment coverage in logarithmic scale.
...	Other graphical parameters.

Value

An object of ggplot2.

See Also

CircSeqAlignTkCoverage-class, calc_coverage

Examples

```
output_dpath <- tempdir()
genome_seq <- system.file(package="CircSeqAlignTk", "extdata", "FR851463.fa")
fq <- system.file(package="CircSeqAlignTk", "extdata", "srna.fq.gz")
ref_index <- build_index(input = genome_seq,
                         output = file.path(output_dpath, 'index'))
aln <- align_reads(input = fq, index = ref_index,
                   output = file.path(output_dpath, 'align_results'))

alncov <- calc_coverage(aln)
plot(alncov)
```

Index