Package 'Ularcirc'

March 30, 2021

Type Package

Title Shiny app for canonical and back splicing analysis (i.e. circular and mRNA analysis)

Version 1.8.0

Description Ularcirc reads in STAR aligned splice junction files and provides visualisation and analysis tools for splicing analysis. Users can assess backsplice junctions and forward canonical junctions.

biocViews DataRepresentation,Visualization, Genetics, Sequencing, Annotation, Coverage, AlternativeSplicing, DifferentialSplicing

License file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.4.0)

Imports AnnotationHub, AnnotationDbi, BiocGenerics, Biostrings, BSgenome, data.table (>= 1.9.4), DT, GenomicFeatures, GenomeInfoDb, GenomeInfoDbData, GenomicAlignments, GenomicRanges, ggplot2, ggrepel, gsubfn, mirbase.db, moments, Organism.dplyr, S4Vectors, shiny, shinydashboard, shinyFiles, shinyjs, Sushi, yaml

RoxygenNote 6.0.1

Suggests BSgenome.Hsapiens.UCSC.hg38, BiocStyle, httpuv, knitr, org.Hs.eg.db, rmarkdown, TxDb.Hsapiens.UCSC.hg38.knownGene

VignetteBuilder knitr

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

git_branch RELEASE_3_12

git_last_commit c22748f

git_last_commit_date 2020-10-27

Date/Publication 2021-03-29

Author David Humphreys [aut, cre]

Maintainer David Humphreys <d.humphreys@victorchang.edu.au>

R topics documented:

bsj_fastq_generate 2
bsj_to_circRNA_sequence 3
Compatible_Annotation_DBs 4
Ularcirc 5

Index **6**

`bsj_fastq_generate` *bsj_fastq_generate*

Description

Takes a circRNA predicted sequence and generates synthetic short sequence reads

Usage

```
bsj_fastq_generate(circRNA_Sequence, fragmentLength = 300,
  readLength = 100, variations = 4, headerID = "")
```

Arguments

`circRNA_Sequence`	: Linear sequence of a circRNA. i.e. the backspice junction is the first and last base of this sequence
`fragmentLength`	: Is the length the library fragment
`readLength`	: The sequence read length
`variations`	: Number of sequences returned for each read type. Note each sequence variation will start at a unique location (where possible)
`headerID`	: Character identifier that will be incorporated into sequence header

Value

Returns a list of two DNAstring sets labelled "read1" and "read2" which correspond to forward and reverse read pairs.

Examples

```
library('Ularcirc')


# Generate a 500nt sequence containing A" and which is flanked with GG and CC.
circRNA_Sequence <- paste(rep('A',500),collapse='')
circRNA_Sequence <- paste('GG',circRNA_Sequence, 'CC', sep='')
# The GG and CC ends of sequence represent ends of linear exons that are circularised.
# Therefore the backsplice junction (BSJ) is GGCC.
# Generate reads that alternate over this BSJ

fastqReads <- bsj_fastq_generate(circRNA_Sequence, fragmentLength=300, readLength=100,
              variations = 4,   # Four type I , II, III, and IV reads generated
          headerID='circRNA_example') # Identifier incorporated in name of each sequence
```

```
# The following will indicate 12 sequences are present in each list entry
length(fastqReads$read1)
length(fastqReads$read2)

# Can create fastq file as follows
Biostrings::writeXStringSet( fastqReads$read1,"circRNA_Sample_R1.fastq.gz",
                             compress = TRUE, format="fastq")
Biostrings::writeXStringSet( fastqReads$read2,"circRNA_Sample_R2.fastq.gz",
                             compress = TRUE, format="fastq")
```

---

bsj_to_circRNA_sequence

bsj_to_circRNA_sequence

Description

Takes one BSJ coordinate and generates a predicted circular RNA sequence.

Usage

```
bsj_to_circRNA_sequence(BSJ, geneID = NULL, genome, TxDb,
  annotationLibrary)
```

Arguments

BSJ	: BSJ coordinate in the format of chr_coordinate_chr_coorindate OR chr:coordinate-coorindate:strand.
geneID	: The gene ID that the BSJ aligns to. Not essential as this can be identified from the BSJ coordinate, however time performance of function improved if this information can be provided.
genome	: Is the length f the library fragment
TxDb	: The sequence read length
annotationLibrary	: annotation database. See details for example.

Value

Returns a DNAstring object.

Examples

```
library('Ularcirc')
library('BSgenome.Hsapiens.UCSC.hg38')
library('TxDb.Hsapiens.UCSC.hg38.knownGene')
TxDb <- TxDb.Hsapiens.UCSC.hg38.knownGene::TxDb.Hsapiens.UCSC.hg38.knownGene
genome <- BSgenome.Hsapiens.UCSC.hg38::BSgenome.Hsapiens.UCSC.hg38
annotationLibrary <- org.Hs.eg.db::org.Hs.eg.db

# Define BSJ. Following two formats are accepted
BSJ <- 'chr2:40430305-40428472:-'       # SLC8A1
BSJ  <- 'chr2_40430305_chr2_40428472'   # SLC8A1
```

```
circRNA_sequence <- bsj_to_circRNA_sequence(BSJ, "SLC8A1", genome,TxDb, annotationLibrary)

# You can also retrieve sequence without passing gene annotation - but this is slower
# circRNA_sequence <- bsj_to_circRNA_sequence(BSJ, NULL, genome,TxDb, annotationLibrary)
```

Compatible_Annotation_DBs *Compatible_Annotation_DBs*

Description

Interogates Bioconductor databases and identifies those that are compatible with Ularcirc. Builds a list of commands that the user can copy to install the required database on their local computer. Once installed the databases are immediately available to Ularcirc upon re-starting the shiny app. This function requires connection to the internet.

Usage

```
Compatible_Annotation_DBs(search_term = "")
```

Arguments

search_term	: character string of a full or part name of a database. Will return only those entries that contain this search term. Not case sensitive.

Value

Returns a list of compatible annotation databases

Examples

```
# Get all Bioconductor annotation databases that are compatible with Ularcirc
library('BSgenome')
library('httpuv')
library('AnnotationHub')
# Prepare a dataframe of all compatible annotation databases
## Not run:  compatible_DBs_human <- Compatible_Annotation_DBs("Hsapiens")

# Example of how to find a relevant database and load the relevant databases:
# This example find hg38 databases
   idx <- grep(pattern="hg38", x= compatible_DBs_human[,"genome"])

   if (!requireNamespace("BiocManager", quietly=TRUE))
       install.packages("BiocManager")
  BiocManager::install(c(compatible_DBs[idx,]))

## End(Not run)
```

Ularcirc *Ularcirc*

Description

When the function is invoked the Ularcirc shiny app is started. The starting screen has quickstart instructions on how to use the software. Please refer to the Ularcirc vignette for a more detailed workflow.

Usage

```
Ularcirc()
```

Value

Does not return anything

Examples

```
# The following commands will load the shiny app either through an RStudio session or
# through your internet browser

library("Ularcirc")
## Not run:  Ularcirc()
```

Index

bsj_fastq_generate, 2
bsj_to_circRNA_sequence, 3

Compatible_Annotation_DBs, 4

Ularcirc, 5