Package 'NxtIRFdata'

December 10, 2024

Title Data for NxtIRF

Version 1.12.0

Date 2024-09-07

Description NxtIRFdata is a companion package for SpliceWiz, an interactive analysis and visualization tool for alternative splicing quantitation (including intron retention) for RNA-seq BAM files. NxtIRFdata contains Mappability files required for the generation of human and mouse references. NxtIRFdata also contains a synthetic genome reference and example BAM files used to demonstrate SpliceWiz's functionality.

BAM files are based on 6 samples from the Leucegene dataset provided by NCBI Gene Expression Omnibus under accession number GSE67039.

License MIT + file LICENSE

Imports ExperimentHub, BiocFileCache, rtracklayer, R.utils

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

biocViews ExperimentData, PackageTypeData, Genome, RNASeqData, ExpressionData, ExperimentHub

Encoding UTF-8

URL https://github.com/alexchwong/NxtIRFdata

BugReports https://support.bioconductor.org/

RoxygenNote 7.1.2

Config/testthat/edition 3

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

git_branch RELEASE_3_20

git_last_commit 01ae92e

git_last_commit_date 2024-10-29

Repository Bioconductor 3.20

Date/Publication 2024-12-10

Author Alex Chit Hei Wong [aut, cre, cph], Ulf Schmitz [ctb]

Maintainer Alex Chit Hei Wong <alexchwong.github@gmail.com>

Contents

`NxtIRFdata-package` *NxtIRFdata: Data Package for SpliceWiz*

Description

This package contains files that provides a workable example for the SpliceWiz package.

Usage

```
chrZ_genome()

chrZ_gtf()

example_bams(path = tempdir(), overwrite = FALSE, offline = FALSE)

get_mappability_exclusion(
  genome_type = c("hg38", "hg19", "mm10", "mm9"),
  as_type = c("GRanges", "bed", "bed.gz"),
  path = tempdir(),
  overwrite = FALSE,
  offline = FALSE
)
```

Arguments

`path`	(Default = tempdir()) The desired destination path in which to place a copy of the files. The directory does not need to exist but its parent directory does.
`overwrite`	(Default = `FALSE`) Whether or not to overwrite files if they already exist in the given path.
`offline`	(Default = `FALSE`) Whether or not to work in offline mode. This may be suitable if these functions have been previously run and the user wishes to run these functions without fetching online hub resources. Default = FALSE
`genome_type`	Either one of `hg38`, `hg19`, `mm10` or `mm9`
`as_type`	(Default "GRanges") Whether to return the Mappability exclusion data as a GRanges object `"GRanges"`, or a BED `"bed"` or gzipped BED `"bed.gz"` copied locally to the given directory `path`.

Details

(Update) Please note that NxtIRFcore is replaced by the SpliceWiz package which will be available from Bioconductor 3.16 onwards!

A synthetic reference, with genome sequence (FASTA) and gene annotation (GTF) files are provided, based on the genes SRSF1, SRSF2, SRSF3, TRA2A, TRA2B, TP53 and NSUN5. These

genes, with an additional 100 flanking nucleotides, were used to construct an artificial "chromosome Z" (chrZ). Gene annotations, based on release-94 of Ensembl GRCh38 (hg38), were modified with genome coordinates corresponding to this artificial chromosome.

Accompanying this, an example dataset was created based on 6 samples from the Leucegene dataset (GSE67039). Raw sequencing reads were downloaded from GSE67039, and were aligned to GRCh38 (Ensembl release-94) using STAR v2.7.3a. Then, alignments belonging to the 7 genes of the chrZ genome were filtered, and the nucleotide sequences of these alignments were realigned to the chrZ reference using STAR.

Additionally, NxtIRFdata contains Mappability exclusion regions generated using NxtIRF/SpliceWiz, suitable for use in generating references based on hg38, hg19, mm10 and mm9 genomes. These were generated empirically. Synthetic 70-nt reads, with start distances 10-nt apart, were systematically generated from the genome. These reads were aligned to the same genome using the STAR aligner. Then, the BAM file read coverage was assessed. Whereas mappable regions are expected to be covered with 7 reads, low mappability regions are defined as regions covered with 4 or fewer reads.

Value

For `chrZ_genome` and `chrZ_gtf`: returns the path to the example genome FASTA and gene annotation GTF files

For `example_bams`: returns a vector specifying the location of the 6 example BAM files, copied to the given `path` directory. Returns NULL if a connection to ExperimentHub could not be established, or if some BAM files could not be downloaded.

For `get_mappability_exclusion`: returns the mappability exclusion regions resource, with type as specified by the parameter `as_type`. Returns NULL if a connection to ExperimentHub could not be established, or if the resource could not be downloaded.

Functions

- `chrZ_genome`: Returns the location of the genome.fa file of the chrZ reference
- `chrZ_gtf`: Returns the location of the transcripts.gtf file of the chrZ reference
- `example_bams`: Fetches data from ExperimentHub and places them in the given path; returns the locations of the 6 example bam files
- `get_mappability_exclusion`: Fetches data from ExperimentHub and places a copy in the given path; returns the location of this Mappability exclusion BED file

References

Generation of the mappability files was performed using NxtIRF/SpliceWiz using a method analogous to that described in:

Middleton R, Gao D, Thomas A, Singh B, Au A, Wong JJ, Bomane A, Cosson B, Eyras E, Rasko JE, Ritchie W. IRFinder: assessing the impact of intron retention on mammalian gene expression. Genome Biol. 2017 Mar 15;18(1):51. https://doi.org/10.1186/s13059-017-1184-4

Examples

```
# returns the location of the genome.fa file of the chrZ reference

genome_path <- chrZ_genome()
```

```
# returns the location of the transcripts.gtf file of the chrZ reference

gtf_path <- chrZ_gtf()

# Fetches data from ExperimentHub and places them in the given path
# returns the locations of the 6 example bam files

bam_paths <- example_bams(path = tempdir())

# Fetches data from AnnotationHub and places them in the given path

# returns the Mappability exclusion for hg38 directly as GRanges object

hg38.MapExcl.gr <- get_mappability_exclusion(
    genome_type = "hg38",
    as_type = "GRanges"
)

# returns the location of the Mappability exclusion gzipped BED for hg38

gzippedBEDpath <- get_mappability_exclusion(
    genome_type = "hg38",
    as_type = "bed.gz",
    path = tempdir()
)

# Getting NxtIRFdata directly from ExperimentHub

require(ExperimentHub)
eh <- ExperimentHub()
NxtIRF_hub <- query(eh, "NxtIRF")
```

Index