Package 'HTSeqGenie'

March 26, 2013

Imports BiocGenerics (>= 0.2.0), IRanges (>= 1.14.3), GenomicRanges (>= 1.7.12), Rsamtools (>= 1.8.5), Biostrings (>= 2.24.1),chipseq (>= 1.7.1), rtracklayer (>= 1.17.19), GenomicFeatures (>= 1.9.31)

Maintainer Gregoire Pau <pau.gregoire@gene.com>

License Artistic-2.0

Title High-throughput sequencing data analysis pipeline.

Type Package

LazyLoad yes

Author Gregoire Pau, Cory Barr, Jens Reeder, Michael Lawrence,Jeremiah Degenhardt, Tom Wu, Melanie Huntley, Matt Brauer

Description A software package to analyse high-throughput sequencing experiments

Version 1.0.0

Date 2012-03-01

Depends
R (>= 2.15.0), ShortRead (>= 1.14.4), parallel, hwriter,Cairo, tools, gmapR (>= 0.99.23), logging

Suggests TxDb.Hsapiens.UCSC.hg19.knownGene, GenomicFeatures,LungCancerLines, org.Hs.eg.db

Collate
'alignReads.R' 'bamUtils.R' 'checkConfig.R' 'config.R''detectAdapterContam.R' 'filterQuality.R' 'io.R''preprocessR

R topics documented:

buildGenomicFeaturesFromTxDb *Build genomic features from a TxDb object*

Description

Build genomic features from a TxDb object

Usage

```
buildGenomicFeaturesFromTxDb(txdb)
```

Arguments

txdb	A TxDb object.

Value

A list named list of GRanges objects containing the biological entities to account for.

Author(s)

Gregoire Pau

Examples

```
## Not run:
 library("TxDb.Hsapiens.UCSC.hg19.knownGene")
 txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
 genomic_features <- buildGenomicFeaturesFromTxDb(txdb)

## End(Not run)
```

getTabDataFromFile *Load tabular data from the NGS pipeline result directory*

Description

Load tabular data from the NGS pipeline result directory

Usage

```
getTabDataFromFile(save_dir, object_name)
```

Arguments

save_dir	A character string containing an NGS pipeline output directory.
object_name	A character string ontaining the regular expression matching a filename in dir_path

Value

A data frame.

HTSeqGenie *Package overview*

Description

The HTSeqGenie package is a robust and efficient software to analyze high-throughput sequencing experiments in a reproducible manner. It supports the RNA-Seq and Exome-Seq protocols and provides: quality control reporting (using the ShortRead package), detection of adapter contamination, read alignment versus a reference genome (using the gmapR package), counting reads in genomic regions (using the GenomicRanges package), and read-depth coverage computation.

Package content

To run the pipeline:

- runPipeline

To access the pipeline output data:

- getTabDataFromFile

To build the genomic features object:

- buildGenomicFeaturesFromTxDb
- TP53GenomicFeatures

Examples

```
## Not run:
 ## build genome and genomic features
 tp53Genome <- TP53Genome()
 tp53GenomicFeatures <- TP53GenomicFeatures()

 ## get the FASTQ files
fastq1 <- system.file("extdata/H1993_TP53_subset2500_1.fastq.gz", package="HTSeqGenie")
fastq2 <- system.file("extdata/H1993_TP53_subset2500_2.fastq.gz", package="HTSeqGenie")

 ## run the pipeline
 save_dir <- runPipeline(
    ## input
    input_file=fastq1,
    input_file2=fastq2,
    paired_ends=TRUE,
    quality_encoding="illumina1.8",

    ## output
    save_dir="test",
    prepend_str="test",
    overwrite_save_dir="erase",

    ## aligner
    path.gsnap_genomes=path(directory(tp53Genome)),
    alignReads.genome=genome(tp53Genome),
  alignReads.additional_parameters="--indel-penalty=1 --novelsplicing=1 --distant-splice-penalty=1",
```

```
    ## gene model
    path.genomic_features=dirname(tp53GenomicFeatures),
    countGenomicFeatures.gfeatures=basename(tp53GenomicFeatures)
    )

## End(Not run)
```

---

runPipeline *Run the NGS analysis pipeline*

Description

Run the NGS analysis pipeline

Usage

```
runPipeline(...)
```

Arguments

...	A list of parameters. See the vignette for details.

Details

This function starts the pipeline. It first preprocesses the input FASTQ reads, align them, count the read overlaps with genomic features and compute the coverage. See the vignette for details.

Value

The path to the NGS output directory.

Author(s)

Jens Reeder, Gregoire Pau

See Also

TP53Genome, TP53GenomicFeatures

Examples

```
## Not run:
## build genome and genomic features
tp53Genome <- TP53Genome()
tp53GenomicFeatures <- TP53GenomicFeatures()

 ## get the FASTQ files
fastq1 <- system.file("extdata/H1993_TP53_subset2500_1.fastq.gz", package="HTSeqGenie")
fastq2 <- system.file("extdata/H1993_TP53_subset2500_2.fastq.gz", package="HTSeqGenie")

## run the pipeline
save_dir <- runPipeline(
    ## input
    input_file=fastq1,
```

```
    input_file2=fastq2,
    paired_ends=TRUE,
    quality_encoding="illumina1.8",

    ## output
    save_dir="test",
    prepend_str="test",
    overwrite_save_dir="erase",

    ## aligner
    path.gsnap_genomes=path(directory(tp53Genome)),
    alignReads.genome=genome(tp53Genome),
   alignReads.additional_parameters="--indel-penalty=1 --novelsplicing=1 --distant-splice-penalty=1",

    ## gene model
    path.genomic_features=dirname(tp53GenomicFeatures),
    countGenomicFeatures.gfeatures=basename(tp53GenomicFeatures)
    )

## End(Not run)
```

TP53GenomicFeatures *Demo genomic features around the TP53 gene*

Description

Build the genomic features of the TP53 demo region

Usage

```
TP53GenomicFeatures()
```

Details

Returns a list of genomic features (gene, exons, transcripts) annotating a region of UCSC hg19 sequence centered on the region of the TP53 gene, with 1 Mb flanking sequence on each side. This is intended as a test/demonstration to run the NGS pipeline in conjunction with the LungCancerLines data package.

Value

A list of GRanges objects containing the genomic features

Author(s)

Gregoire Pau

See Also

TP53Genome, buildGenomicFeaturesFromTxDb, runPipeline

Index