Package 'gDNAx'

March 25, 2024

Type Package

Title Diagnostics for assessing genomic DNA contamination in RNA-seq data

Version 1.0.2

Description Provides diagnostics for assessing genomic DNA contamination in RNA-seq data, as well as plots representing these diagnostics. Moreover, the package can be used to get an insight into the strand library protocol used and, in case of strand-specific libraries, the strandedness of the data. Furthermore, it provides functionality to filter out reads of potential gDNA origin.

License Artistic-2.0

Encoding UTF-8

Depends R (>= 4.3)

Imports methods, BiocGenerics, BiocParallel, Biostrings, S4Vectors, IRanges, GenomeInfoDb, GenomicRanges, GenomicFiles, GenomicAlignments, GenomicFeatures, Rsamtools, AnnotationHub, RColorBrewer, AnnotationDbi, bitops, plotrix, SummarizedExperiment, grDevices, graphics, stats, utils

Suggests BiocStyle, knitr, rmarkdown, RUnit, TxDb.Hsapiens.UCSC.hg38.knownGene, gDNAinRNAseqData

biocViews Transcription, Transcriptomics, RNASeq, Sequencing, Preprocessing, Software, GeneExpression, Coverage, DifferentialExpression, FunctionalGenomics, SplicedAlignment, Alignment

VignetteBuilder knitr

RoxygenNote 7.3.1

URL https://github.com/functionalgenomics/gDNAx

BugReports https://github.com/functionalgenomics/gDNAx/issues

Collate 'AllGenerics.R' 'AllClasses.R' 'dx.R' 'filterBAMtx.R' 'utils.R' 'strandedness.R' 'gDNAx.R'

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

git_branch RELEASE_3_18

git_last_commit 0fb2969

git_last_commit_date 2024-03-15

Repository Bioconductor 3.18

Date/Publication 2024-03-25

Author Beatriz Calvo-Serra [aut, cre],
Robert Castelo [aut]

Maintainer Beatriz Calvo-Serra <beatriz.calvo@upf.edu>

R topics documented:

`gDNAx-package` *gDNAx: diagnostics for assessing genomic DNA contamination in RNA-seq data*

Description

The gDNAx package provides diagnostics for assessing genomic DNA contamination in RNA-seq data, as well as plots representing these diagnostics. Moreover, the package can be used to get an insight into the strand library protocol used and, in case of strand-specific libraries, the strandedness of the data. Furthermore, it provides functionality to filter out reads of potential gDNA origin.

Details

The main functions are:

- `identifyStrandMode` - identify `strandMode` (strandedness) in RNA-seq data samples based on the proportion of reads aligning to the same or opposite strand as transcripts in the annotations
- `gDNAdx` - calculate diagnostics for assessing the presence of genomic DNA in RNA-seq data over a subset of the alignments in the input BAM files
- `getDx` and codeplot - get and plot statistics on genomic DNA contamination levels, respectively
- `filterBAMtxFlag` and codefilterBAMtx - filter alignments in a BAM file using criteria based on a transcriptome annotation

For detailed information on usage, see the package vignette, by typing `vignette("gDNAx")`.

All questions and bug reports should be posted to the Bioconductor Support Site:

https://support.bioconductor.org

The code of the development version of the package is available at the GitHub repository:

https://github.com/functionalgenomics/gDNAx

Author(s)

Maintainer: Beatriz Calvo-Serra <beatriz.calvo@upf.edu>

Authors:

- Robert Castelo <robert.castelo@upf.edu>

See Also

Useful links:

- https://github.com/functionalgenomics/gDNAx
- Report bugs at https://github.com/functionalgenomics/gDNAx/issues

---

`filterBAMtx` *Filter alignments in a BAM file using a transcriptome*

Description

Filter alignments in a BAM file using criteria based on a transcriptome annotation.

Use 'filterBAMtxFlag()' to set what types of alignment in a BAM file should be filtered using the function 'filterBAMtx()', among being splice-compatible with one or more junctions, splice-compatible exonic, intronic or intergenic.

Usage

```
filterBAMtx(
  object,
  path = ".",
  txflag = filterBAMtxFlag(),
  param = ScanBamParam(),
  yieldSize = 1e+06,
  verbose = TRUE,
  BPPARAM = SerialParam(progressbar = verbose)
)

filterBAMtxFlag(
  isSpliceCompatibleJunction = FALSE,
  isSpliceCompatibleExonic = FALSE,
  isIntronic = FALSE,
```

```
  isIntergenic = FALSE
)

testBAMtxFlag(flag, value)
```

Arguments

object	gDNAx object obtained with the function 'gDNAdx()'.
path	Directory where to write the output BAM files.
txflag	A value from a call to the function 'filterBAMtxFlag()'.
param	A 'ScanBamParam' object.
yieldSize	(Default 1e6) Number of records in the input BAM file to yield each time the file is read. The lower the value, the smaller memory consumption, but in the case of large BAM files, values below 1e6 records may decrease the overall performance.
verbose	(Default TRUE) Logical value indicating if progress should be reported through the execution of the code.
BPPARAM	An object of a BiocParallelParam subclass to configure the parallel execution of the code. By default, a SerialParam object is used, which does not use any parallelization, with the flag `progress=TRUE` to show progress through the calculations.
isSpliceCompatibleJunction	(Default FALSE) Logical value indicating if spliced alignments overlapping a transcript in a "splice compatible" way should be included in the BAM file. For paired-end reads, one or both alignments must have one or more splice site(s) compatible with splicing. See `OverlapEncodings`.
isSpliceCompatibleExonic	(Default FALSE) Logical value indicating if alignments without a splice site, but that overlap a transcript in a "splice compatible" way, should be included in the BAM file. For paired-end reads, none of the alignments must be spliced, and each pair can be in different exons (or in the same one), as long as they are "splice compatible". See `OverlapEncodings`.
isIntronic	(Default FALSE) Logical value indicating if alignments mapping to introns should be included in the BAM file.
isIntergenic	(Default FALSE) Logical value indicating if alignments aligned to intergenic regions should be included in the BAM file.
flag	A value from a call to the function 'filterBAMtxFlag()'.
value	A character vector with the name of a flag.

Value

A vector of output filename paths.

Examples

```
library(gDNAinRNAseqData)

library(TxDb.Hsapiens.UCSC.hg38.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg38.knownGene

# Getting the 'gDNAx' object
bamfiles <- LiYu22subsetBAMfiles()
bamfiles <- bamfiles[c(1,7)] # using a subset of samples
gdnax <- gDNAdx(bamfiles, txdb, singleEnd=FALSE, strandMode=NA)

# Filtering splice-compatible alignments and writing them into new BAM files
fbf <- filterBAMtxFlag(isSpliceCompatibleJunction=TRUE,
                       isSpliceCompatibleExonic=TRUE)
dir <- tempdir()
fstats <- filterBAMtx(gdnax, path=dir, txflag=fbf)
list.files(dir, pattern="*.bam$")


# Filtering splice-compatible alignments and writing them into new BAM files
fbf <- filterBAMtxFlag(isSpliceCompatibleJunction=TRUE,
                       isSpliceCompatibleExonic=FALSE,
                       isIntronic=FALSE,
                       isIntergenic = FALSE)

testBAMtxFlag(fbf, "isSpliceCompatibleJunction")
```

---

gDNAdx *Calculate gDNA diagnostics*

Description

Calculate diagnostics for assessing the presence of genomic DNA (gDNA) in RNA-seq data over a subset of the alignments in the input BAM files.

Plot diagnostics calculated with gDNAdx()

Using the output from gDNAdx(), plot the genomic origin of the alignments.

Plot fragments length distributions estimated with gDNAdx()

Usage

```
gDNAdx(
  bfl,
  txdb,
  singleEnd = TRUE,
  strandMode = 1L,
  stdChrom = TRUE,
  yieldSize = 100000L,
```

```
  verbose = TRUE,
  BPPARAM = SerialParam(progressbar = verbose)
)

## S4 method for signature 'gDNAx,ANY'
plot(x, group = 1L, labelpoints = FALSE, ...)

plotAlnOrigins(x, group = 1L)

plotFrgLength(x)
```

Arguments

bfl	A BamFile or BamFileList object, or a character string vector of BAM filenames.
txdb	A character string of a TxDb package, or a TxDb object, with gene and transcript annotations. For accurate calculations, it is important that the version of these annotations matches the version of the annotations used to inform the alignment of spliced reads, by the short-read aligner software that generated the input BAM files.
singleEnd	(Default FALSE) Logical value indicating if reads are single (TRUE) or paired-end (FALSE).
strandMode	(Default 1L) Numeric vector which can take values 0, 1, 2 or NA. The strand mode is a per-object switch on GAlignmentPairs objects that controls the behavior of the strand getter. See GAlignmentPairs class for further detail. If singleEnd = TRUE, then strandMode is ignored. For not strand-specific libraries, use NA
stdChrom	(Default TRUE) Logical value indicating whether only alignments in the 'standard chromosomes' should be used. Consult the help page of the function keepStandardChromosomes from the package GenomeInfoDb for further information.
yieldSize	(Default 1e5) Number of records to read from each input BAM file to calculate the diagnostics.
verbose	(Default TRUE) Logical value indicating if progress should be reported through the execution of the code.
BPPARAM	An object of a BiocParallelParam subclass to configure the parallel execution of the code. By default, a SerialParam object is used, which does not use any parallelization, with the flag progress=TRUE to show progress through the calculations.
x	A 'gDNAx' object.
group	A string character vector or a factor, with as many values as BAM files analyzed in 'x', whose values define groups among those BAM files.
labelpoints	(Default FALSE) A logical indicator that labels points in those plots where each point represents a BAM file. Labels correspond to the index number of the BAM file in 'x'.
...	Named arguments to be passed to plot.

Value

A gDNAx object.

Examples

```
library(gDNAinRNAseqData)

library(TxDb.Hsapiens.UCSC.hg38.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg38.knownGene

# Retrieving BAM files
bamfiles <- LiYu22subsetBAMfiles()
bamfiles <- bamfiles[c(1,4,7)] # using a subset of samples

# Getting information about the gDNA concentrations of each BAM file
pdat <- LiYu22phenoData(bamfiles)

gdnax <- gDNAdx(bamfiles, txdb, singleEnd=FALSE, strandMode=NA)
gdnax

# plot gDNA diagnostic measures
plot(gdnax, group=pdat$gDNA, pch=19)

# plot origin of alignments per sample
plotAlnOrigins(gdnax, group=pdat$gDNA)

# plot fragments length distributions
plotFrgLength(gdnax)
```

---

gDNAx-class *gDNAx class*

Description

This is a class for storing the results of a call to the 'gDNAdx()' function.

Usage

```
## S4 method for signature 'gDNAx'
getDx(x)

## S4 method for signature 'gDNAx'
show(object)

## S4 method for signature 'gDNAx'
getIgc(x)

## S4 method for signature 'gDNAx'
getInt(x)
```

Arguments

x	A gDNAx object.
object	A gDNAx object.

Value

`features()`: A `GRanges` object with intergenic ranges.

`features()`: A `GRanges` object with intron ranges.

Slots

`bfl` A BamFileList object.

`txdbpkg` A TxDb object.

`singleEnd` Logical value indicating if reads are single (`TRUE`) or paired-end (`FALSE`).

`strandMode` Numeric vector which can take values 0, 1 or 2. The strand mode is a per-object switch on `GAlignmentPairs` objects that controls the behavior of the strand getter. See `GAlignmentPairs` class for further detail.

`stdChrom` Logical value indicating whether only alignments in the 'standard chromosomes' should be used. Consult the help page of the function `keepStandardChromosomes` from the package `GenomeInfoDb` for further information.

`readLength` Integer value storing the read length.

`yieldSize` Integer value storing the number of alignments employed by the function `gDNAdx()`.

`diagnostics` A 'data.frame' object storing the diagnostics calculated by the function 'gDNAdx()'.

`igcfrglen` A 'list' object storing the fragment lengths derived from alignments in intergenic regions.

`intfrglen` A 'list' object storing the fragment lengths derived from alignments in intronic regions.

`scjfrglen` A 'list' object storing the fragment lengths derived from spliced-compatible junction alignments in transcripts.

`scefrglen` A 'list' object storing the fragment lengths derived from spliced-compatible exonic alignments in transcripts.

`sicfrglen` A 'list' object storing the fragment lengths derived from splice-incompatible alignments in transcripts.

`intergenic` A 'GRanges' object storing the intergenic feature annotations.

`intronic` A 'GRanges' object storing the intronic feature annotations.

`transcripts` A 'GRangesList' object storing the transcript annotations.

`tx2gene` A string character vector storing the correspondence between transcripts and genes according to an 'TxDb' object.

Examples

```
# Getting the 'gDNAx' object. Can be done using the commented code:
# library(gDNAinRNAseqData)
# library(TxDb.Hsapiens.UCSC.hg38.knownGene)
# txdb <- TxDb.Hsapiens.UCSC.hg38.knownGene
# bamfiles <- LiYu22subsetBAMfiles() # Retrieving BAM files
# gdnax <- gDNAdx(bamfiles, txdb, singleEnd=FALSE, strandMode=NA)

# Here to reduce example running time, the 'gDNAx' object is loaded
gdnax_f <- file.path(system.file("extdata", package="gDNAx"), "gdnax.rds")
gdnax <- readRDS(gdnax_f)
gdnax


# Getting statistics
dx <- getDx(gdnax)
head(dx)

gdnax

igc <- getIgc(gdnax)
head(igc, n=3)

int <- getInt(gdnax)
head(int, n=3)
```

`identifyStrandMode` *Identify strandMode*

Description

Identify `strandMode` (strandedness) in RNA-seq data samples based on the proportion of reads aligning to the same or opposite strand as transcripts in the annotations.

Compute strandedness for each feature in RNA-seq data samples based on the proportion of reads aligning to the same strand as feature annotations in relation to the total number of reads aligning to that feature.

Usage

```
identifyStrandMode(
  bfl,
  txdb,
  singleEnd = TRUE,
  stdChrom = TRUE,
  yieldSize = 1000000L,
  verbose = TRUE,
  BPPARAM = SerialParam(progressbar = verbose)
```

```
)

strnessByFeature(
  bfl,
  features,
  singleEnd = TRUE,
  strandMode = 1L,
  yieldSize = 1000000L,
  ambiguous = FALSE,
  p = 0.6,
  verbose = TRUE,
  BPPARAM = SerialParam(progressbar = verbose)
)
```

Arguments

bfl	A BamFile or BamFileList object, or a character string vector of BAM filenames.
txdb	A character string of a TxDb package, or a TxDb object, with gene and transcript annotations. For accurate calculations, it is important that the version of these annotations matches the version of the annotations used to inform the alignment of spliced reads, by the short-read aligner software that generated the input BAM files.
singleEnd	(Default FALSE) Logical value indicating if reads are single (TRUE) or paired-end (FALSE).
stdChrom	(Default TRUE) Logical value indicating whether only alignments in the 'standard chromosomes' should be used. Consult the help page of the function keepStandardChromosomes from the package GenomeInfoDb for further information.
yieldSize	(Default 5e5) Field inherited from BamFile. The BAM is read by chunks. yieldSize represents the number of records to read for each chunk.
verbose	(Default TRUE) Logical value indicating if progress should be reported through the execution of the code.
BPPARAM	An object of a BiocParallelParam subclass to configure the parallel execution of the code. By default, a SerialParam object is used, which does not use any parallelization, with the flag progress=TRUE to show progress through the calculations.
features	A GRanges or GRangesList object with annotations of features (e.g. genes, transcripts, etc.).
strandMode	(Default 1L) Numeric vector which can take values 0, 1, or 2. The strand mode is a per-object switch on GAlignmentPairs objects that controls the behavior of the strand getter. See GAlignmentPairs class for further detail. If singleEnd = TRUE, then strandMode is ignored.
ambiguous	(Default FALSE) Logical value indicating if reads that overlap a region with features annotated to both strands should be included in the strandedness value computation.

p (Default 0.6) Numeric value for the exact binomial test performed for the strandedness value of each feature, representing the hypothesized probability of success (i.e. the strandedness value expected for a non-stranded dataset).

Details

If the value in the "strandMode1" column is > 0.90, `strandMode` is set to 1L. If "strandMode2" column is > 0.90, `strandMode` is set to 2L. If "strandMode1" and "strandMode2" are comprised between 0.40 and 0.60, `strandMode` is set to `NA` If none of the three previous criteria are met, `strandMode` is set to "ambiguous". This criteria can be conservative in some cases (e.g. when there is genomic DNA contamination), for this reason we recommend to check the data.frame with strandedness values.

In case of single-end data, the same criteria are used, but the interpretation of `strandMode = 1L` and `strandMode = 2L` changes: when `strandMode = 1L` the strand of the read is concordant with the reference annotations, when `strandMode = 2L` the correct read strand is the opposite to the one of the read.

A subset of 200,000 alignments overlapping gene annotations are used to compute strandedness.

Strandedness is computed for each feature and BAM file according to the `strandMode` specified in case of paired-end data. For single-end, the original strand of reads is used. All alignments from the BAM file(s) are considered to compute the strandedness.

The p value should be close to 0.5, representing the strandedness expected for a non-stranded RNA-seq library.

Value

A list object with two elements:

- "strandMode": the `strandMode` of the sample(s) following `GAlignmentPairs` class definition. If all samples have the same `strandMode`, the length of the vector is 1. It can take values: `NA` (library is not strand-specific), 1 (strand of pair is strand of its first alignment), 2 (strand of pair is strand of its second alignment) or "ambiguous" (additional category used here for samples not fitting any of the three previous categories). See "Details" section below to know the classification criteria, as well as to how interpret results for single-end data.
- "Strandedness": data.frame with one row per sample and 3 columns. "strandMode1": proportion of alignments aligned to the same strand than a transcript according to the strand of its first alignment. "strandMode2": proportion of alignments aligned to the same strand than a transcript according to the strand of its second alignment. "ambiguous": alignments aligned to regions with transcripts in both strands.

A SummarizedExperiment with three assays:

- "strness": contains strandedness values for each feature and sample.
- "counts": number of reads aligning to each feature on the same strand (according to `strandMode`).
- "counts_invstrand": number of reads aligning to each feature but on the opposite strand (according to `strandMode`).

Examples

```
library(gDNAinRNAseqData)

library(TxDb.Hsapiens.UCSC.hg38.knownGene)
txdb <- TxDb.Hsapiens.UCSC.hg38.knownGene

# Retrieving BAM files
bamfiles <- LiYu22subsetBAMfiles()
bamfiles <- bamfiles[c(1,7)] # using a subset of samples

strandM <- identifyStrandMode(bamfiles, txdb, singleEnd=FALSE)
strandM$strandMode
head(strandM$Strandedness)


features <- range(exonsBy(txdb, by="gene"))
features <- features[1:100]
sByFeature <- strnessByFeature(bamfiles, features, singleEnd=FALSE,
                               strandMode=2L)
sByFeature
```

Index