Package ‘bambu’

July 17, 2020

Type Package

Title Reference-guided isoform reconstruction and quantification for long read RNA-Seq data

Version 0.3.0

Description Multi-sample transcript discovery and quantification using long read RNA-Seq data.

License GPL-3

Encoding UTF-8

LazyData true

Depends R(>= 4.0.0),
data.table(>= 1.1.8),
dplyr,
SummarizedExperiment(>= 1.1.6),
GenomicRanges,
BiocManager,
GenomicFeatures,
ggplot2

Suggests knitr,
rmarkdown,
fs,
testthat,
ComplexHeatmap,
circlize,
ggbio,
RColorBrewer,
gridExtra,
BSgenome.Hsapiens.NCBI.GRCh38,
TxDb.Hsapiens.UCSC.hg38.knownGene,
AnnotationDbi,
BSgenome,
BiocGenerics,
BiocParallel,
Biostrings

Enhances parallel

SystemRequirements

biocViews FeatureExtraction,
GeneExpression,
GenomeAnnotation,
ImmunoOncology,
Normalization,
RNASeq,
Regression,
Sequencing,
Software,
Transcription,
Transcriptomics

bugReports https://github.com/GoekeLab/bambu/issues

URL `https://github.com/GoekeLab/bambu`

RoxygenNote 7.1.0

LinkingTo Rcpp,
RcppArmadillo,
RcppProgress

Imports S4Vectors(>= 0.22.1),
IRanges,
GenomicAlignments,
glmnet,
Rsamtools,
Rcpp,
RcppArmadillo,
RcppProgress,
progress

VignetteBuilder knitr,
rmarkdown

R topics documented:

bambu *long read isoform reconstruction and quantification*

Description

This function takes bam file of genomic alignments and performs isoform recontruction and gene and transcript expression quantification. It also allows saving of read class files of alignments, extending provided annotations, and quantification based on extended annotations. When multiple samples are provided, extended annotations will be combined across samples to allow comparison.

Usage

```
bambu(
  reads = NULL,
  readClass.file = NULL,
  readClass.outputDir = NULL,
  annotations = NULL,
  genomeSequence = NULL,
  stranded = FALSE,
  ncore = 1,
  yieldSize = NULL,
  isoreParameters = NULL,
  emParameters = NULL,
  extendAnnotations = TRUE,
  verbose = FALSE
)
```

Arguments

`reads` A string or a vector of strings specifying the paths of bam files for genomic alignments, or a `BamFile` object or a `BamFileList` object (see `Rsamtools`).

`readClass.file` A string or a vector of strings specifying the read class files that are saved during previous run of `bambu`.

`readClass.outputDir`
A string variable specifying the path to where read class files will be saved.

`annotations` A `TxDb` object or A GRangesList object obtained by `prepareAnnotations` or `prepareAnnotationsFromGTF`.

`genomeSequence` A fasta file or a BSGenome object.

`stranded` A boolean for strandedness, defaults to FALSE.

`ncore` specifying number of cores used when parallel processing is used, defaults to 1.

`yieldSize` see `Rsamtools`.

`isoreParameters`
A list of controlling parameters for isoform reconstruction process:

- prefix specifying prefix for new gene Ids (genePrefix.number), defaults to empty

- remove.subsetTx indicating whether filter to remove read classes which are a subset of known transcripts(), defaults to TRUE
- min.readCount specifying minimun read count to consider a read class valid in a sample, defaults to 2
- min.readFractionByGene specifying minimum relative read count per gene, highly expressed genes will have many high read count low relative abundance transcripts that can be filtered, defaults to 0.05
- min.sampleNumber specifying minimum sample number with minimum read count, defaults to 1
- min.exonDistance specifying minum distance to known transcript to be considered valid as new, defaults to 35
- min.exonOverlap specifying minimum number of bases shared with annotation to be assigned to the same gene id, defaults 10 base pairs

`emParameters` A list of controlling parameters for quantification algorithm estimation process:

- maxiter specifying maximum number of run interations, defaults to 10000.
- bias specifying whether to correct for bias, defaults to FALSE.
- conv specifying the covergence trheshold control, defaults to 0.0001.

`extendAnnotations`
A logical variable indicating whether annotations are to be extended for quantification.

`verbose` A logical variable indicating whether processing messages will be printed.

Details

Main function

Value

A list of two SummarizedExperiment object for transcript expression and gene expression.

Examples

```
## =====================
## Minimum read support 5
## Increase EM convergence threshold to 10^(-6)
 test.bam <- system.file("extdata",
 "SGNex_A549_directRNA_replicate5_run1_chr9_1_1000000.bam",
 package = "bambu")
 gr <- readRDS(system.file("extdata",
 "annotationGranges_txdbGrch38_91_chr9_1_1000000.rds",
 package = "bambu"))
 fa.file <- system.file("extdata",
 "Homo_sapiens.GRCh38.dna_sm.primary_assembly_chr9_1_1000000.fa",
 package = "bambu")
 se = bambu(reads = test.bam,  annotations = gr,
 genomeSequence = fa.file, extendAnnotations = FALSE)
```

plot *plot*

Description

redefine plot for plot.bambu

Usage

```
plot(
  se,
  ...,
  group.variable = NULL,
  type = c("annotation", "pca", "heatmap"),
  gene_id = NULL,
  transcript_id = NULL
)
```

Arguments

`se`	An summarized experiment object obtained from `bambu` or `transcriptToGeneExpression`.
`...`	Additional arguments
`group.variable`	Variable for grouping in plot, has be to provided if choosing to plot PCA.
`type`	plot type variable, a values of annotation for a single gene with heatmap for isoform expressions, pca, or heatmap, see details.
`gene_id`	specifying the gene_id for plotting gene annotation, either gene_id or transcript_id has to be provided when type = "annotation".
`transcript_id`	specifying the transcript_id for plotting transcript annotation, either gene_id or transcript_id has to be provided when type = "annotation"

Value

A heatmap plot for all samples

plot.bambu *plot.bambu*

Description

plotSEOuptut

Usage

```
## S3 method for class 'bambu'
plot(
  se,
  group.variable = NULL,
  type = c("annotation", "pca", "heatmap"),
  gene_id = NULL,
  transcript_id = NULL
)
```

Arguments

`se` An summarized experiment object obtained from `bambu` or `transcriptToGeneExpression`.

`group.variable` Variable for grouping in plot, has be to provided if choosing to plot PCA.

`type` plot type variable, a values of annotation for a single gene with heatmap for isoform expressions, pca, or heatmap, see details.

`gene_id` specifying the gene_id for plotting gene annotation, either gene_id or transcript_id has to be provided when type = "annotation".

`transcript_id` specifying the transcript_id for plotting transcript annotation, either gene_id or transcript_id has to be provided when type = "annotation"

Details

`type` indicates the type of plots to be plotted. There are two types of plots can be chosen, PCA or heatmap.

Value

A heatmap plot for all samples

Examples

```
se <- readRDS(system.file("extdata",
"seOutputCombined_SGNex_A549_directRNA_replicate5_run1_chr9_1_1000000.rds",
 package = "bambu"))
colnames(se) <- colData(se)$name <- c("sample1","sample2")
assays(se)$CPM[,2]  <- pmax(0, rnorm(length(assays(se)$CPM[,2]),
assays(se)$CPM[,2],10))
plot.bambu(se, type = "heatmap")
```

prepareAnnotations *prepare annotations from txdb object*

Description

Function to prepare tables and genomic ranges for transript reconstruction using a txdb object

Usage

```
prepareAnnotations(txdb)
```

Arguments

txdb	a TxDb object

Value

A GRangesList object

Examples

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

prepareAnnotationsFromGTF *Prepare annotation granges object from GTF file into a GRangesList object*

Description

Prepare annotation granges object from GTF file

Usage

```
prepareAnnotationsFromGTF(file)
```

Arguments

file	a GTF file

Details

Unlike readFromGTF, this function finds out the equivalence classes between the transcripts, with mcols data having three columns:

- TXNAME specifying prefix for new gene Ids (genePrefix.number), defaults to empty
- GENEID indicating whether filter to remove read classes which are a subset of known transcripts(), defaults to TRUE
- eqClass specifying minimun read count to consider a read class valid in a sample, defaults to 2

Value

A GRangesList object

Examples

```
gtf.file <- system.file("extdata",
"Homo_sapiens.GRCh38.91_chr9_1_1000000.gtf",
package = "bambu")
gr <- prepareAnnotationsFromGTF(gtf.file)
```

readFromGTF *convert a GTF file into a GRangesList*

Description

Outputs GRangesList object from reading a GTF file

Usage

```
readFromGTF(file)
```

Arguments

file	a .gtf file

Value

grlist a GRangesList object, with two columns

- TXNAME specifying prefix for new gene Ids (genePrefix.number), defaults to empty
- GENEID indicating whether filter to remove read classes which are a subset of known transcripts(), defaults to TRUE

Examples

```
gtf.file <- system.file("extdata",
"Homo_sapiens.GRCh38.91_chr9_1_1000000.gtf",
package = "bambu")
readFromGTF(gtf.file)
```

`transcriptToGeneExpression` *transcript to gene expression*

Description

Reduce transcript expression to gene expression

Usage

```
transcriptToGeneExpression(se)
```

Arguments

`se` a summarizedExperiment object from `bambu`

Value

A SummarizedExperiment object

Examples

```
se <- readRDS(system.file(”extdata”,
”seOutput_SGNex_A549_directRNA_replicate5_run1_chr9_1_1000000.rds”,
package = ”bambu”))
transcriptToGeneExpression(se)
```

`writeBambuOutput` *Write bambu results to GTF and transcript/gene-count files*

Description

Outputs a GTF file, transcript-count file, and gene-count file from bambu

Usage

```
writeBambuOutput(se, path)
```

Arguments

`se` a `SummarizedExperiment` object from `bambu`

`path` the destination of the output files (gtf, transcript counts, and gene counts)

Value

The function will generate three files, a .gtf file for the annotations, two .txt files for transcript and gene counts respectively.

Examples

```
se <- readRDS(system.file("extdata",
"seOutput_SGNex_A549_directRNA_replicate5_run1_chr9_1_1000000.rds",
package = "bambu"))
path <- tempdir()
writeBambuOutput(se,path)
```

---

writeToGTF *write GRangeslist into GTF file*

Description

Write annotation GRangesList into a GTF file

Usage

```
writeToGTF(annotation, file, geneIDs = NULL)
```

Arguments

annotation	a GRangesList object
file	the output gtf file name
geneIDs	an optional dataframe of geneIDs (column 2) with the corresponding transcriptIDs (column 1)

Value

gtf a GTF dataframe

Examples

```
outputGtfFile <- tempfile()
gr <- readRDS(system.file("extdata",
"annotationGranges_txdbGrch38_91_chr9_1_1000000.rds",
package = "bambu"))
writeToGTF(gr, outputGtfFile)
```

Index