Package 'bambu'

April 12, 2022

Type Package

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

Version 2.0.6

Description bambu is a R package for multi-sample transcript discovery and quantification using long read RNA-Seq data. You can use bambu after read alignment to obtain expression estimates for known and novel transcripts and genes. The output from bambu can directly be used for visualisation and downstream analysis such as differential gene expression or transcript usage.

License GPL-3 + file LICENSE

Encoding UTF-8

ByteCompile true

Depends R(>= 4.1), SummarizedExperiment(>= 1.1.6), S4Vectors(>= 0.22.1), BSgenome, IRanges

Suggests AnnotationDbi, Biostrings, rmarkdown, BiocFileCache, ggplot2, ComplexHeatmap, circlize, ggbio, gridExtra, knitr, testthat, BSgenome.Hsapiens.NCBI.GRCh38, TxDb.Hsapiens.UCSC.hg38.knownGene, ExperimentHub (>= 1.15.3), DESeq2, NanoporeRNASeq, apeglm, utils, DEXSeq

Enhances parallel

SystemRequirements

biocViews Alignment, Coverage, DifferentialExpression, FeatureExtraction, GeneExpression, GenomeAnnotation, GenomeAssembly, ImmunoOncology, MultipleComparison, Normalization, RNASeq, Regression, Sequencing, Software, Transcription, Transcriptomics

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

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

RoxygenNote 7.1.1

LinkingTo Rcpp, RcppArmadillo

Imports BiocGenerics, BiocParallel, data.table, dplyr, tidyr, GenomeInfoDb, GenomicAlignments, GenomicFeatures, GenomicRanges, stats, Rsamtools, methods, Rcpp, xgboost

VignetteBuilder knitr

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

git_branch RELEASE_3_14

git_last_commit e340a81

git_last_commit_date 2022-02-23

Date/Publication 2022-04-12

Author Ying Chen [cre, aut], Andre Sim [aut], Yuk Kei Wan [aut], Jonathan Goeke [aut]

Maintainer Ying Chen <chen_ying@gis.a-star.edu.sg>

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,
    rcFile = NULL,
    rcOutDir = NULL,
    annotations = NULL,
    genome = NULL,
```

```
    stranded = FALSE,
    ncore = 1,
    yieldSize = NULL,
    opt.discovery = NULL,
    opt.em = NULL,
    discovery = TRUE,
    quant = TRUE,
    verbose = FALSE,
    lowMemory = 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).
rcFile	A string or a vector of strings specifying the read class files that are saved during previous run of bambu.
rcOutDir	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.
genome	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.
opt.discovery	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 35bp

min.exonOverlap specifying minimum number of bases shared with annotation to be assigned to the same gene id, defaults to 10bp

min.primarySecondaryDist specifying the minimum number of distance threshold, defaults to 5bp

min.primarySecondaryDistStartEnd1 specifying the minimum number of distance threshold, used for extending annotation, defaults to 5bp

min.primarySecondaryDistStartEnd2 specifying the minimum number of distance threshold, used for estimating distance to annotation, defaults to 5bp

min.txScore.multiExon specifying the minimum transcript level threshold for multi-exon transcripts during sample combining, defaults to 0

min.txScore.singleExon specifying the minimum transcript level threshold for single-exon transcripts during sample combining, defaults to 1

max.txNDR specifying the maximum NDR rate to novel transcript output from detected read classes, defaults to 0.1

`opt.em` A list of controlling parameters for quantification algorithm estimation process:

maxiter specifying maximum number of run iterations, defaults to 10000

degradationBias correcting for degradation bias, defaults to TRUE

conv specifying the covergence threshold control, defaults to 0.0001

minvalue specifying the minvalue for convergence consideration, defaults to 0.00000001

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

`quant` A logical variable indicating whether quantification will be performed

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

`lowMemory` Read classes will be processed by chromosomes when lowMemory is specified. This option provides an efficient way to process big samples.

Details

Main function

Value

`bambu` will output different results depending on whether *quant* mode is on. By default, *quant* is set to TRUE, so `bambu` will generate a *SummarizedExperiment* object that contains the transcript expression estimates. Transcript expression estimates can be accessed by *counts()*, including the following variables

counts expression estimates

CPM sequencing depth normalized estimates

fullLengthCounts estimates of read counts mapped as full length reads for each transcript

partialLengthCounts estimates of read counts mapped as partial length reads for each transcript

uniqueCounts counts of reads that are uniquely mapped to each transcript

theta raw estimates

Output annotations that are usually the annotations with/without novel transcripts/genes added, depending on whether *discovery* mode is on can be accessed by *rowRanges()* Transcript to gene map can be accessed by *rowData()*, with *eqClass* that defining equivalent class for each transcript

In the case when *quant* is set to FALSE, i.e., only transcript discovery is performed, `bambu` will report the *grangeslist* of the extended annotations.

Examples

```
## ====================
test.bam <- system.file("extdata",
    "SGNex_A549_directRNA_replicate5_run1_chr9_1_1000000.bam",
    package = "bambu")
fa.file <- system.file("extdata",
    "Homo_sapiens.GRCh38.dna_sm.primary_assembly_chr9_1_1000000.fa",
    package = "bambu")
gr <- readRDS(system.file("extdata",
    "annotationGranges_txdbGrch38_91_chr9_1_1000000.rds",
    package = "bambu"))
se <- bambu(reads = test.bam, annotations = gr,
    genome = fa.file, discovery = TRUE, quant = TRUE)
```

`plotBambu` *plot.bambu*

Description

plotSEOuptut

Usage

```
plotBambu(
    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"))
plotBambu(se, type = "PCA")
```

`prepareAnnotations` *prepare annotations from txdb object or gtf file*

Description

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

Usage

```
prepareAnnotations(x)
```

Arguments

x	A `TxDb` object or a gtf file

Value

A `GRangesList` object

Examples

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

readFromGTF *convert a GTF file into a GRangesList*

Description

Outputs GRangesList object from reading a GTF file

Usage

```
readFromGTF(file, keep.extra.columns = NULL)
```

Arguments

file	a .gtf file
keep.extra.columns	a vector with names of columns to keep from the the attributes in the gtf file. For ensembl, this could be keep.extra.columns=c('gene_name','gene_biotype', 'transcript_biotype', 'transcript_name')

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, prefix = "")
```

Arguments

se	a SummarizedExperiment object from bambu.
path	the destination of the output files (gtf, transcript counts, and gene counts)
prefix	the prefix of the output files

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