

# Package ‘bambu’

February 27, 2021

**Type** Package

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

**Version** 1.0.2

**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

**LazyData** true

**Depends** R(>= 4.0.0), SummarizedExperiment(>= 1.1.6), S4Vectors(>= 0.22.1), IRanges

**Suggests** AnnotationDbi, Biostrings, BiocFileCache, ggplot2, ComplexHeatmap, circlize, ggbio, gridExtra, knitr, testthat, BSgenome.Hsapiens.NCBI.GRCh38, TxDb.Hsapiens.UCSC.hg38.knownGene, ExperimentHub (>= 1.15.3), DESeq2, NanoporeRNASeq, BSgenome, 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, GenomeInfoDb, GenomicAlignments, GenomicFeatures, GenomicRanges, stats, glmnet, Rsamtools, methods, Rcpp

**VignetteBuilder** knitr

**git\_url** <https://git.bioconductor.org/packages/bambu>

**git\_branch** RELEASE\_3\_12

**git\_last\_commit** 5340451

**git\_last\_commit\_date** 2020-11-10

**Date/Publication** 2021-02-26

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

**Maintainer** Ying Chen <chen\_ying@gis.a-star.edu.sg>

## R topics documented:

bambu . . . . .	2
plotBambu . . . . .	4
prepareAnnotations . . . . .	5
readFromGTF . . . . .	5
transcriptToGeneExpression . . . . .	6
writeBambuOutput . . . . .	7
writeToGTF . . . . .	7

<b>Index</b>	<b>9</b>
--------------	----------

---

bambu	<i>long read isoform reconstruction and quantification</i>
-------	--

---

### Description

This function takes bam file of genomic alignments and performs isoform reconstruction 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,
  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).
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: <ul style="list-style-type: none"> <li>• prefix specifying prefix for new gene Ids (genePrefix.number), defaults to empty</li> <li>• remove.subsetTx indicating whether filter to remove read classes which are a subset of known transcripts(), defaults to TRUE</li> <li>• min.readCount specifying minimum read count to consider a read class valid in a sample, defaults to 2</li> <li>• 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</li> <li>• min.sampleNumber specifying minimum sample number with minimum read count, defaults to 1</li> <li>• min.exonDistance specifying minimum distance to known transcript to be considered valid as new, defaults to 35</li> <li>• min.exonOverlap specifying minimum number of bases shared with annotation to be assigned to the same gene id, defaults 10 base pairs</li> </ul>
opt.em	A list of controlling parameters for quantification algorithm estimation process: <ul style="list-style-type: none"> <li>• maxiter specifying maximum number of run iterations, defaults to 10000.</li> <li>• bias specifying whether to correct for bias, defaults to FALSE.</li> <li>• conv specifying the coverage threshold control, defaults to 0.0001.</li> </ul>
discovery	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**

```
## =====
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 = FALSE)
```

---

plotBambu

*plot.bambu*


---

**Description**

plotSEOutput

**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 <a href="#">bambu</a> or <a href="#">transcriptToGeneExpression</a> .
group.variable	Variable for grouping in plot, has to be 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	<i>prepare annotations from txdb object or gtf file</i>
--------------------	---

---

**Description**

Function to prepare tables and genomic ranges for transcript 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	<i>convert a GTF file into a GRangesList</i>
-------------	--

---

**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	<i>Write bambu results to GTF and transcript/gene-count files</i>
------------------	---

---

### Description

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

### Usage

```
writeBambuOutput(se, path, prefix = "")
```

### Arguments

se	a <a href="#">SummarizedExperiment</a> object from <a href="#">bambu</a> .
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	<i>write GRangesList into GTF file</i>
------------	--

---

### 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_chr9_1_1000000.rds",
  package = "bambu"
))
writeToGTF(gr, outputGtfFile)
```



# Index

bambu, [2](#), [3](#), [4](#), [6](#), [7](#)

plotBambu, [4](#)

prepareAnnotations, [3](#), [5](#)

readFromGTF, [5](#)

SummarizedExperiment, [7](#)

transcriptToGeneExpression, [4](#), [6](#)

type, [4](#)

writeBambuOutput, [7](#)

writeToGTF, [7](#)