

# Package ‘VaSP’

September 5, 2024

**Type** Package

**Version** 1.16.0

**Title** Quantification and Visualization of Variations of Splicing in Population

**Description** Discovery of genome-wide variable alternative splicing events from short-read RNA-seq data and visualizations of gene splicing information for publication-quality multi-panel figures in a population. (Warning: The visualizing function is removed due to the dependent package Sushi deprecated. If you want to use it, please change back to an older version.)

**URL** <https://github.com/yuhuihui2011/VaSP>

**BugReports** <https://github.com/yuhuihui2011/VaSP/issues>

**License** GPL (>= 2.0)

**Depends** R (>= 4.0), ballgown

**Imports** IRanges, GenomicRanges, S4Vectors, parallel, matrixStats, GenomicAlignments, GenomeInfoDb, Rsamtools, cluster, stats, graphics, methods

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**biocViews** RNASeq, AlternativeSplicing, DifferentialSplicing, StatisticalMethod, Visualization, Preprocessing, Clustering, DifferentialExpression, KEGG, ImmunoOncology

**Encoding** UTF-8

**LazyData** false

**RoxygenNote** 7.1.1

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

**git\_branch** RELEASE\_3\_19

**git\_last\_commit** 96d1dec

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.19

**Date/Publication** 2024-09-04

**Author** Huihui Yu [aut, cre] (<<https://orcid.org/0000-0003-2725-1937>>),  
 Qian Du [aut] (<<https://orcid.org/0000-0003-3864-8745>>),  
 Chi Zhang [aut] (<<https://orcid.org/0000-0002-1827-8137>>)

**Maintainer** Huihui Yu <yuhuihui2011@foxmail.com>

## Contents

|                        |          |
|------------------------|----------|
| BMfinder . . . . .     | 2        |
| getDepth . . . . .     | 3        |
| getGeneinfo . . . . .  | 4        |
| rice.bg . . . . .      | 5        |
| spliceGene . . . . .   | 6        |
| spliceGenome . . . . . | 7        |
| <b>Index</b>           | <b>9</b> |

---

|          |   |
|----------|---|
| BMfinder | <i>Discover bimodal distrubition features</i> |
|----------|---|

---

## Description

Find bimodal distrubition features and divide the samples into 2 groups by k-means clustering.

## Usage

```
BMfinder(x, p.value = 0.01, maf = 0.05, miss = 0.05, fold = 2, log = FALSE,
         cores = detectCores() - 1)
```

## Arguments

|         |  |
|---------|--|
| x       | a numeric matrix with feature rows and sample columns, e.g., splicing score matrix from <a href="#">spliceGenome</a> or <a href="#">spliceGene</a> function. |
| p.value | p.value threshold for bimodal distrubition test  |
| maf     | minor allele frequency threshold in k-means clustering   |
| miss    | missing grouping rate threshold in k-means clustering  |
| fold    | fold change threshold between the two groups   |
| log     | whether the scores are to be logarithmic. If TRUE, all the scores are log2 transformed before k-means clustering: $x = \log_2(x+1)$ .                        |
| cores   | threads to be used. This value is passed to <b>?mclapply</b> in <b>parallel</b> package  |

## Details

The matrix contains 1, 2 and NA, and values of 'x' in group 2 are larger than group 1.

**Value**

a matrix with feature rows and sample columns.

**Examples**

```
data(rice.bg)
score<-spliceGene(rice.bg, 'MSTRG.183',junc.type='score')
score<-round(score,2)
as<-BMfinder(score,cores=1) # 4 bimodal distrubition features found

##compare
as
score[rownames(score)%in%rownames(as),]
```

---

getDepth

*Get Read Depth*


---

**Description**

Get read depth from a BAM file (in bedgraph format)

**Usage**

```
getDepth(x, chrom, start, end)
```

**Arguments**

|       |                                       |
|-------|---------------------------------------|
| x     | path to a BAM file                    |
| chrom | chromosome of a region to be searched |
| start | start position                        |
| end   | end position                          |

**Value**

a data.frame in bedgraph file format.

**Examples**

```
path <- system.file('extdata',package='VaSP')
bam_files<-list.files(path,'bam$')
bam_files

depth<-getDepth(file.path(path, bam_files[1]), 'Chr1',
                 start=1171800, end=1179400)
head(depth)

# library(Sushi)
# plotBedgraph(depth, 'Chr1',chromstart=1171800, chromend=1179400,yaxt='s')
```

```
# mtext('Depth',side=2,line=2.5,cex=1.2,font=2)
# labelgenome('Chr1',1171800,1179400,side=1,scipen=20,n=5,scale='Kb')
```

---

getGeneinfo                      *Get Gene Informaton from a ballgown object*

---

## Description

Get gene informaton from a ballgown object by genes or by genomic regions

## Usage

```
getGeneinfo(genes = NA, bg, chrom, start, end, samples = sampleNames(bg),
            trans.select = NA)
```

## Arguments

|              |   |
|--------------|---|
| genes        | a character vector specifying gene IDs in 'bg'. Any values other than NA override genomic region (chrom, start, stop)                       |
| bg           | ballgown object   |
| chrom        | chromosome of a region  |
| start        | start postion   |
| end          | stop postion  |
| samples      | names of samples. The transcripts in these samples are subjected to 'trans.select'  |
| trans.select | logical expression-like string, indicating transcript rows to select from a matrix of transcript coverages: NA value keeps all transcripts. |

## Value

a data.frame in bed-like file format

## Examples

```
data(rice.bg)
unique(geneIDs(rice.bg))

gene_id <- c('MSTRG.181', 'MSTRG.182', 'MSTRG.183')
geneinfo <- getGeneinfo(genes=gene_id, rice.bg)
trans <- table(geneinfo$name) # show how many exons each transcript has
trans

# library(Sushi)
# chrom = geneinfo$chrom[1]
# chromstart = min(geneinfo$start) - 1e3
# chromend = max(geneinfo$stop) + 1e3
# color = rep(SushiColors(2)(length(trans)), trans)
```

```
# par(mar=c(3,1,1,1))
# plotGenes(geneinfo, chrom, chromstart, chromend, col = color, bheight = 0.2,
#           bentline = FALSE, plotgenotype = 'arrow', labeloffset = 0.5)
# labelgenome(chrom, chromstart, chromend, side = 1, n = 5, scale = 'Kb')
```

---

rice.bg

*Rice ballgown object*

---

## Description

Small ballgown object created with a subset of rice RNAseq data, for demonstration purposes

## Format

a ballgown object with 33 transcripts and 6 samples

## Details

The raw RNA-seq data were screened and trimmed using Trimmomatic (Bolger et al., 2014) and RNA-seq mapping, transcript assembly, and quantification were conducted with HISAT, StringTie, and Ballgown by following the method described by Pertea et al. (Pertea et al., 2016). The rice.bg is a subset ballgown object with 33 transcripts and 6 samples (Yu et al., 2021).

## Source

The raw RNA-seq data were from the project of variation in transcriptional responses to salt stress in rice (SRA Accession: [SRP106054](https://www.ncbi.nlm.nih.gov/sra/SRP106054))

## References

Yu, H., Du, Q., Campbell, M., Yu, B., Walia, H. and Zhang, C. (2021), Genome-wide discovery of natural variation in pre-mRNA splicing and prioritising causal alternative splicing to salt stress response in rice. *New Phytol.* <https://doi.org/10.1111/nph.17189>

Bolger, A.M., Lohse, M., and Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* 30, 2114-2120.

Pertea, M., Kim, D., Pertea, G.M., Leek, J.T., and Salzberg, S.L. (2016). Transcript-level expression analysis of RNA-seq experiments with HISAT, StringTie and Ballgown. *Nat Protoc* 11, 1650-1667.

## Examples

```
data(rice.bg)
rice.bg
# ballgown instance with 33 transcripts and 6 samples
```

spliceGene

*Calculate Splicing Scores for One Gene***Description**

Calculate splicing Scores from ballgown object for a given gene. This function can only calculate one gene. Please use function `spliceGenome` to obtain genome-wide splicing scores.

**Usage**

```
spliceGene(bg, gene, samples = sampleNames(bg), junc.type = c("score", "count"),
           trans.select = "rowMaxs(x)>=1", junc.select = "rowMaxs(x)>=5")
```

**Arguments**

|                           |  |
|---------------------------|--|
| <code>bg</code>           | ballgown object  |
| <code>gene</code>         | a character string specifying gene id  |
| <code>samples</code>      | names of samples   |
| <code>junc.type</code>    | type of junction estimate ('score' for junction score; 'count' for junction read count)  |
| <code>trans.select</code> | logical expression-like string, indicating transcript rows to select from a matrix of transcript coverages: NA value keeps all transcripts. e.g. use <code>trans.select='rowMaxs(x)&gt;=1'</code> to filter the transcripts with the maximum coverage among all the samples less than 1. |
| <code>junc.select</code>  | logical expression-like string, indicating junction rows to select from a matrix of junction counts: NA value keeps all junctions. e.g. use <code>junc.select='rowMaxs(x)&gt;=5'</code> to filter the junctions with the maximum read count among all the samples less than 5.           |

**Details**

`score` = junction count/gene-level per base read coverage. Row functions for matrices are useful to select transcripts and junctions. See `matrixStats` package.

**Value**

a matrix of junction scores with intron rows and sample columns.

**References**

Yu, H., Du, Q., Campbell, M., Yu, B., Walia, H. and Zhang, C. (2021), Genome-wide discovery of natural variation in pre-mRNA splicing and prioritising causal alternative splicing to salt stress response in rice. *New Phytol.* <https://doi.org/10.1111/nph.17189>

**See Also**

`spliceGenome`, which calculates splicing scores in whole genome.

**Examples**

```

data(rice.bg)
rice.bg
head(geneIDs(rice.bg))

score<-spliceGene(rice.bg, 'MSTRG.183',junc.type='score')
count<-spliceGene(rice.bg, 'MSTRG.183',junc.type='count')

## compare
tail(score)
tail(count)

## get intron structure
intron<-structure(rice.bg)$intron
intron[intron$id%in%rownames(score)]

```

---

spliceGenome

*Calculate Genome-wide Splicing Scores*


---

**Description**

Calculate splicing scores from ballgown objects for all genes.

**Usage**

```

spliceGenome(bg, gene.select = "rowQuantiles(x,probs = 0.05)>=1",
             intron.select = "rowQuantiles(x,probs = 0.95)>=5")

```

**Arguments**

|               |   |
|---------------|---|
| bg            | ballgown object   |
| gene.select   | logical expression-like string, indicating genes to select from a matrix of gene-level coverages: NA value keeps all genes. e.g. gene.select = 'rowQuantiles(x,probs = 0.05)>=1' keeps the genes with the read coverage greater than or equal to 1 in at least 95 (0.05 quantile). Used to filter low expressed genes.                            |
| intron.select | logical expression-like string, indicating introns to select from a matrix of junction counts: NA value keeps all introns. e.g. intron.select = 'rowQuantiles(x,probs = 0.95)>=5' keeps the introns with the read count greater than or euqal to 5 in at least 5 (0.95 quantile). Used to filter introns with very few junction reads supporting. |

**Details**

score = junction count/gene-level per base read coverage. Row functions for matrices in [matrixStats](#) package are useful to select genes and introns.

**Value**

a list of two elements: 'score' is matrix of intron splicing scores with intron rows and sample columns and 'intron' is a [GRanges](#) object of intron structure. See [structure](#) in **ballgown** package

**References**

Yu, H., Du, Q., Campbell, M., Yu, B., Walia, H. and Zhang, C. (2021), Genome-wide discovery of natural variation in pre-mRNA splicing and prioritising causal alternative splicing to salt stress response in rice. *New Phytol.* <https://doi.org/10.1111/nph.17189>

**See Also**

[spliceGene](#), which calculates splicing scores in one gene.

**Examples**

```
data(rice.bg)
rice.bg

splice<-spliceGenome(rice.bg, gene.select=NA, intron.select=NA)
names(splice)

head(splice$score)
splice$intron
```

# Index

## \* datasets

rice.bg, 5

BMfinder, 2

getDepth, 3

getGeneinfo, 4

GRanges, 8

matrixStats, 6, 7

rice.bg, 5

spliceGene, 2, 6, 8

spliceGenome, 2, 6, 7

structure, 8