

# Package ‘EBSEA’

December 27, 2024

**Type** Package

**Title** Exon Based Strategy for Expression Analysis of genes

**Version** 1.35.0

**Author** Arfa Mehmood, Asta Laiho, Laura L. Elo

**Maintainer** Arfa Mehmood <arfa.mehmood@utu.fi>

**Description** Calculates differential expression of genes based on exon counts of genes obtained from RNA-seq sequencing data.

**License** GPL-2

**biocViews** Software, DifferentialExpression, GeneExpression, Sequencing

**Imports** DESeq2, graphics, stats, EmpiricalBrownsMethod

**RoxygenNote** 7.1.1

**Encoding** UTF-8

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Depends** R (>= 4.0.0)

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

**git\_branch** devel

**git\_last\_commit** be7ea14

**git\_last\_commit\_date** 2024-10-29

**Repository** Bioconductor 3.21

**Date/Publication** 2024-12-27

## Contents

EBSEA	2
exonCounts	3
filterCounts	3
visualizeGenes	4
<b>Index</b>	<b>6</b>

EBSEA

*Exon Based Strategy for Expression Analysis of genes***Description**

EBSEA takes the filtered raw exon-level read counts as input, normalizes and performs a two-group statistical comparison with DESeq2. The exon-level results are aggregated to the gene-level using empirical Brown's method. The samples in the two groups can be paired.

**Usage**

```
EBSEA(data, columnData, design, test = "Wald", contrasts = NULL, plot = FALSE)
```

**Arguments**

<code>data</code>	A dataframe of raw exon-counts
<code>columnData</code>	A dataframe indicated the groups of the samples.
<code>design</code>	Design matrix (see more information od design matrixes in DESeq2 reference manual)
<code>test</code>	The statistical test to be carried out. It can be either Wald or Likelihood Ratio Test. For further details about the methods you can look into DESeq2 refernce manual. Default: Wald
<code>contrasts</code>	a character vector with exactly three elements: the name of a factor in the design formula, the name of the numerator level for the fold change, and the name of the denominator level for the fold change Default: NULL
<code>plot</code>	A logical value indicating a volcano plot is produced. Default: FALSE

**Value**

The function returns a list containing containing exon and gene-level results. ExonTable is a data frame containing an average expression, log2 fold-change, p-value and adjusted p-value. GeneTable is a data frame containing the gene-level p-value, and adjusted-value. Other returned elements include the raw and normalised exon-level read counts, group information and design matrix used.

**References**

Laiho, A., & Elo, L. L. (2014). A note on an exon-based strategy to identify differentially expressed genes in RNA-seq experiments. *PLoS One*, 9(12), e115964.

**Examples**

```
# The exon-based analysis for unpaired samples can be performed as follows:
data(exonCounts)
group <- data.frame('group' = as.factor(c('G1', 'G1', 'G1', 'G2', 'G2', 'G2', 'G2')))
row.names(group) <- colnames(exonCounts)
design <- ~group
```

```
ebsea.out <- EBSEA(exonCounts, group, design)
# The exon-based analysis for paired samples with contrast provided can be performed as follows:
data(exonCounts)
group <- data.frame('group' = as.factor(c('G1', 'G1', 'G1', 'G2', 'G2', 'G2', 'G2')),
  'paired' = as.factor(c(1,2,3,1,2,3,3)))
row.names(group) <- colnames(exonCounts)
design <- ~group
contrastInfo <- c('group', 'G2', 'G1')
ebsea.out <- EBSEA(exonCounts, group, design, contrasts = contrastInfo)
```

---

exonCounts	<i>Subset of Pasilla Dataset</i>
------------	----------------------------------

---

### Description

exonCounts consists of a subset of the exon counts from the pasilla dataset.

### Usage

```
data("exonCounts")
```

### Format

A data frame with 1000 rows and 7 variables

### Source

Exoncounts from Pasilla package <https://bioconductor.org/packages/release/data/experiment/html/pasilla.html>

### References

Huber W, Reyes A (2020). pasilla: Data package with per-exon and per-gene read counts of RNA-seq samples of Pasilla knock-down by Brooks et al., Genome Research 2011

---

filterCounts	<i>Filter Count Data</i>
--------------	--------------------------

---

### Description

Filtering of exons based on their expression levels

### Usage

```
filterCounts(x, mean = 1, exonCount = 1)
```

**Arguments**

x	A numeric dataframe of exon counts across the samples. Exon number in format GeneName:Exonnumber should be indicated in the row name and sample names as column names.
mean	Exons with average count value across the dataset less than mean are filtered out. Default: 1
exonCount	After filtering the individual exons, only genes with at least the given number of exons remaining will be retained. Default: 1

**Value**

A dataframe of filtered counts of exons

**Examples**

```
data(exonCounts)
res <- filterCounts(exonCounts)
```

---

visualizeGenes

*Visualize gene*

---

**Description**

Produces a visualization summarizing the normalized read count in each sample group and expression difference across the expressed exons. Top panel contains the log<sub>2</sub> fold-change for each expressed exon. Asterisk denotes the significance level (\*: < 0.05, \*\*: < 0.01). Bottom panel shows the averaged normalized read count for each sample group. The title of the figure shows the gene name and the adjusted gene-level p-value (fdr)

**Usage**

```
visualizeGenes(gene, ebsea.out)
```

**Arguments**

gene	Gene name matching the input data.
ebsea.out	Plots the mean count and fold-change the exons of the specified gene.

**Value**

Plots the mean count and fold-change across the exons of the specified gene.

**Examples**

```
data(exonCounts)
group <- data.frame('group' = as.factor(c('G1', 'G1', 'G1', 'G2', 'G2', 'G2', 'G2')))
row.names(group) <- colnames(exonCounts)
design <- ~group
ebsea.out <- EBSEA(exonCounts, group, design)
visualizeGenes('FBgn000017', ebsea.out)
```

# Index

## \* **datasets**

    exonCounts, [3](#)

EBSEA, [2](#)

exonCounts, [3](#)

filterCounts, [3](#)

visualizeGenes, [4](#)