

Package: LRDE (via r-universe)

May 30, 2026

Type Package

Title Differential Expression Analysis with Long Read RNA-Seq Data

Version 0.99.6

Description Provides hurdle negative binomial models for differential expression analysis with long-read RNA-Seq data.

Depends R (>= 4.5.0)

Imports stats, SummarizedExperiment

Suggests BiocStyle, knitr, rmarkdown, BiocGenerics, testthat (>= 3.0.0)

biocViews Software, DifferentialExpression, Sequencing, RNASeq, LongRead, GeneExpression, Regression

URL <https://github.com/ziyang773/LRDE>

BugReports <https://github.com/ziyang773/LRDE/issues>

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

VignetteBuilder knitr

Config/testthat/edition 3

Config/pak/sysreqs zlib1g-dev

Repository <https://bioc.r-universe.dev>

Date/Publication 2026-04-08 03:55:08 UTC

RemoteUrl <https://github.com/bioc/LRDE>

RemoteRef HEAD

RemoteSha 449980e0deedafe9365599a6aceda35f61589046

Contents

LRDE-package	2
hurdle_LRT	3
hurdle_Wald_Test	4
prepareDGE	5
sizeFactorsEst	6
tagwiseEst	8
Index	9

LRDE-package	<i>LRDE: Differential Expression Analysis for Long-Read RNA-Seq Data</i>
--------------	--

Description

The **LRDE** package provides statistical methods for differential expression analysis of long-read RNA sequencing (RNA-Seq) data using a hurdle negative binomial generalized linear model (hurdle-NB GLM).

It implements procedures for:

- Estimation of sample-specific size factors for normalization
- Modeling zero inflation via group-specific expression probabilities
- Gene-wise (tag-wise) dispersion estimation
- Statistical testing for differential expression

These methods are designed to address key challenges in long-read RNA-Seq data, including limited sample sizes and excess zero counts (dropout events).

Details

The main functions in this package include:

- **prepareDGE**: Prepare count data for analysis. Converts supported input types (`matrix`, `data.frame`, `DGEList`, `DESeqDataSet`, and `SummarizedExperiment`) into a standardized format.
- **sizeFactorsEst**: Estimate sample-specific size factors for normalization.
- **tagwiseEst**: Estimate gene-specific (tag-wise) dispersion parameters for a hurdle negative binomial model using prior information from bin-level estimates.
- **hurdle_LRT**: Perform gene-wise likelihood ratio tests (LRT) for differential expression.
- **hurdle_Wald_Test**: Perform gene-wise Wald tests for differential expression.

Typical workflow:

1. Prepare data using `prepareDGE`
2. Normalize counts with `sizeFactorsEst`
3. Estimate tag-wise dispersions using `tagwiseEst`
4. Perform differential expression testing with `hurdle_LRT` or `hurdle_Wald_Test`

Author(s)

Ziyang Liu <jacobleo773@gmail.com>

See Also

[prepareDGE](#), [sizeFactorsEst](#), [tagwiseEst](#), [hurdle_LRT](#), [hurdle_Wald_Test](#)

Examples

```
# Load the package
library(LRDE)

# Simulate count data
set.seed(123)
mat <- matrix(rnbinom(300, size = 5, mu = 5), nrow = 50)
grp <- factor(c("A", "A", "A", "B", "B", "B"))

# Prepare data
y <- prepareDGE(mat, grp)

# Normalize counts
y <- sizeFactorsEst(y)

# Estimate dispersions
y <- tagwiseEst(y)

# Differential expression testing
y <- hurdle_Wald_Test(y)
y <- hurdle_LRT(y)

# Access results
head(y$lrt_stats)
head(y$p.values)
```

hurdle_LRT

Likelihood Ratio Test for Hurdle Negative Binomial Model

Description

Performs gene-wise likelihood ratio tests (LRT) for differential expression using a hurdle negative binomial model with fixed zero probabilities and tag-wise dispersions.

Usage

```
hurdle_LRT(y)
```

Arguments

y A list-like object returned from `tagwiseEst()` containing:

- counts** Numeric matrix of gene expression counts (genes x samples).
- samples** Data frame with columns `group` (factor) and `size.factor` (numeric).
- tagwise.disp** Numeric vector of estimated tag-wise dispersions.
- zero_prob_matrix** Matrix of zero probabilities per group per gene.

Details

For each gene:

- The null model assumes a single shared mean across groups.
- The alternative model estimates group-specific means.
- Zero probabilities and dispersions are fixed from prior estimates.
- When one group has all zero counts, a one-sided Z-test is applied instead.

Value

The input object `y` with two additional elements:

lrt_stats Numeric vector of LRT statistics for each gene.

p.values Numeric vector of corresponding p-values.

Examples

```
set.seed(123)
mat <- matrix(rnbinom(30, size = 5, mu = 5), nrow = 5)
grp <- c("A", "A", "A", "B", "B", "B")
y <- prepareDGE(mat, grp)
y <- sizeFactorsEst(y)
y <- tagwiseEst(y)
y <- hurdle_LRT(y)
```

hurdle_Wald_Test

Wald Test for Hurdle Negative Binomial Model

Description

Performs gene-wise Wald tests for differential expression using a hurdle negative binomial model with fixed zero probabilities and tag-wise dispersions.

Usage

```
hurdle_Wald_Test(y)
```

Arguments

y A list-like object returned from `tagwiseEst()` containing:

- counts** Numeric matrix of gene expression counts (genes x samples).
- samples** Data frame with columns `group` (factor) and `size.factor` (numeric).
- tagwise.disp** Numeric vector of estimated tag-wise dispersions.
- zero_prob_matrix** Matrix of zero probabilities per group per gene.

Details

For each gene:

- Zero probabilities and dispersions are fixed from prior estimates.
- The model estimates group-specific mean parameters.
- When one group has all zero counts, a one-sided Z-test is applied instead.
- Otherwise, a standard two-sided Wald test is applied on the log-difference of group means.

Value

The input object `y` with two additional elements:

wald_stats Numeric vector of Wald statistics for each gene.

p.values Numeric vector of corresponding p-values.

Examples

```
set.seed(123)
mat <- matrix(rnbinom(30, size = 5, mu = 5), nrow = 5)
grp <- c("A", "A", "A", "B", "B", "B")
y <- prepareDGE(mat, grp)
y <- sizeFactorsEst(y)
y <- tagwiseEst(y)
y <- hurdle_Wald_Test(y)
```

```
prepareDGE
```

Prepare Count Data for Differential Expression Analysis

Description

Converts various supported input types to a standardized list format for downstream differential expression analysis. Supports `matrix`, `data.frame`, `DGEList`, `DESeqDataSet`, and `SummarizedExperiment` objects.

Usage

```
prepareDGE(data, group)
```

Arguments

<code>data</code>	A numeric matrix, data.frame, or supported object containing counts.
<code>group</code>	A vector of group labels for the columns/samples of data. Must be the same length as the number of columns in data.

Details

This function performs input validation.

- Checks for non-negative numeric values and absence of NA.
- Ensures group labels match the number of samples.
- Automatically assigns column names if missing.
- Returns a list suitable for use with hurdle model-based DE functions.

Value

A list with two elements:

counts An integer matrix of counts.

samples A data.frame containing sample-level metadata: `group`, `lib.size`, and `size.factor`.

Examples

```
# Example with a matrix
set.seed(123)
mat <- matrix(rnbinom(30, size = 5, mu = 5), nrow = 5)
grp <- c("A", "A", "A", "B", "B")
y <- prepareDGE(mat, grp)

# Example with a SummarizedExperiment
if (requireNamespace("SummarizedExperiment", quietly = TRUE)) {
  se <- SummarizedExperiment::SummarizedExperiment(assays = list(counts = mat))
  y_se <- prepareDGE(se, grp)
  y
}
```

sizeFactorsEst

Estimate Size Factors for Normalization

Description

Computes sample-specific size factors for long-read RNA-Seq data, used to normalize counts for differential expression analysis.

Usage

```
sizeFactorsEst(y, type = c("poscounts", "ratio"), locfunc = stats::median)
```

Arguments

y	A count matrix (<code>matrix</code> or <code>data.frame</code>) or the output of <code>prepareDGE()</code> . If a <code>matrix/data.frame</code> is provided, the function assumes two equal-sized groups.
type	Character string specifying the method for estimating size factors: "poscounts" Geometric mean-based method. "ratio" Simple ratio method using the mean of log counts. Default is "poscounts".
locfunc	Function to summarize log-ratios across genes. Defaults to <code>median</code> .

Details

This function implements two methods for size factor estimation:

- **poscounts**: Computes a geometric mean of positive counts per gene, then calculates ratios for each sample. Normalizes so that the geometric mean of size factors equals 1.
- **ratio**: Uses the mean of log-counts per gene across samples to compute ratios.

The function automatically normalizes counts using the estimated size factors and stores gene-level normalized means in `baseMean`.

Value

A list (same structure as `prepareDGE()` output) with:

counts Original count matrix (integer).

samples Data frame with sample information, updated `size.factor`.

baseMean Normalized mean of counts per gene.

Examples

```
# Using a count matrix
#' set.seed(123)
mat <- matrix(rnbinom(30, size = 5, mu = 5), nrow = 5)
grp <- c("A", "A", "A", "B", "B", "B")
y <- prepareDGE(mat, grp)
y <- sizeFactorsEst(y, type = "poscounts")
```

`tagwiseEst`*Tag-wise Dispersion Estimation for Hurdle Negative Binomial Model*

Description

Estimate gene-specific (tag-wise) dispersion parameters for a hurdle negative binomial model using prior information derived from bin-level estimates.

Usage

```
tagwiseEst(y)
```

Arguments

`y` A list object created by `prepareDGE` with size factors estimated, containing counts and sample information.

Details

This function performs the following steps:

1. Retrieves bin-level prior estimates of zero probabilities and log-dispersion for each gene via `priorEst`.
2. Fixes the zero probabilities and optimizes only the mean parameters and dispersion for each gene individually.
3. Uses the internal function `nll_hurdle_fixed_P` to compute the negative log-likelihood with fixed zero probabilities.

The resulting `tagwise.disp` will be used for downstream differential expression analysis.

Value

The input `y` object augmented with:

tagwise.disp Numeric vector of estimated gene-wise dispersions.

zero_prob_matrix Numeric matrix of fixed zero probabilities for each gene and group.

Examples

```
set.seed(123)
mat <- matrix(rnbinom(30, size = 5, mu = 5), nrow = 5)
grp <- c("A", "A", "A", "B", "B", "B")
y <- prepareDGE(mat, grp)
y <- sizeFactorsEst(y)
y <- tagwiseEst(y)
head(y$tagwise.disp)
head(y$zero_prob_matrix)
```

Index

* **package**

LRDE-package, 2

hurdle_LRT, 2, 3, 3

hurdle_Wald_Test, 2, 3, 4

LRDE (LRDE-package), 2

LRDE-package, 2

median, 7

prepareDGE, 2, 3, 5, 7, 8

priorEst, 8

sizeFactorsEst, 2, 3, 6

tagwiseEst, 2, 3, 8