Package 'LRDE'

November 12, 2026

Type Package

Title Differential Expression Analysis with Long Read RNA-Seq Data

Version 1.0.4

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

Depends R (>= 4.6.0)

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

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

git_branch RELEASE_3_23

git_last_commit ee929ae

git_last_commit_date 2026-09-24

Repository Bioconductor 3.23

Date/Publication 2026-11-11

Author Ziyang Liu [aut, cre] (ORCID: <https://orcid.org/0009-0004-2098-434X>),
Hongxu Ding [aut, fnd],
Xiaoxiao Sun [aut, fnd],
Ziyuan Wang [aut, fnd]

Maintainer Ziyang Liu <jacobleo773@gmail.com>

Contents

LRDE-package . . . 2
hurdle.LRT . . . 3
hurdle.Wald.Test . . . 4
hurdle_LRT.dist . . . 5
hurdle_LRT.mean . . . 6
hurdle_Wald_Test.dist . . . 8
hurdle_Wald_Test.mean . . . 9
prepareDGE . . . 10
priorEst . . . 11
sizeFactorsEst . . . 12
tagwiseEst . . . 13

Index **15**

`LRDE-package` *LRDE: Differential Expression Analysis for Long-Read RNA-Seq Data*

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 differential expression testing using a hurdle negative binomial model. The function can test differences in either the expression mean or the overall expression distribution between groups.

Usage

```
hurdle.LRT(y, test = c("mean", "distribution"))
```

Arguments

y	A list-like object returned from tagwiseEst containing the count matrix, sample information, and estimated tag-wise dispersions.
test	Character string specifying the hypothesis to test. Use "mean" to test differences in the positive-count mean while fixing zero probabilities and dispersions, or "distribution" to jointly test differences in the zero probability and positive-count mean. The default is "mean".

Details

When test = "mean", the function calls hurdle_LRT.mean. When test = "distribution", it calls hurdle_LRT.dist.

Value

The input object y with two additional elements:

lrt_stats Numeric vector of test 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)

mean_test <- hurdle.LRT(y, test = "mean")
distribution_test <- hurdle.LRT(y, test = "distribution")
```

---

hurdle.Wald.Test *Wald Test for Hurdle Negative Binomial Model*

Description

Performs gene-wise differential expression testing using a hurdle negative binomial model. The function can test differences in either the expression mean or the overall expression distribution between groups.

Usage

```
hurdle.Wald.Test(y, test = c("mean", "distribution"))
```

Arguments

y	A list-like object returned from tagwiseEst containing the count matrix, sample information, and estimated tag-wise dispersions.
test	Character string specifying the hypothesis to test. Use "mean" to test differences in the positive-count mean while fixing zero probabilities and dispersions, or "distribution" to jointly test differences in the zero probability and positive-count mean. The default is "mean".

Details

When test = "mean", the function calls hurdle_Wald_Test.mean. When test = "distribution", it calls hurdle_Wald_Test.dist.

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)

mean_test <- hurdle.Wald.Test(y, test = "mean")
distribution_test <- hurdle.Wald.Test(y, test = "distribution")
```

---

hurdle_LRT.dist — *Distributional Likelihood Ratio Test for Hurdle Negative Binomial Model*

Description

Performs gene-wise likelihood ratio tests (LRTs) for distributional differential expression using a hurdle negative binomial model. The test jointly evaluates differences in the zero probability and positive-count mean between groups while holding the tag-wise dispersion fixed.

Usage

```
hurdle_LRT.dist(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` and `size.factor`. **tagwise.disp** Numeric vector of estimated tag-wise dispersions.

Details

For each gene:

- The null model assumes a shared nonzero probability and a shared positive-count mean across groups.
- The alternative model estimates group-specific nonzero probabilities and positive-count means.
- The dispersion is fixed at its tag-wise estimate obtained from `tagwiseEst`.
- The LRT uses two degrees of freedom when both groups contain positive counts and one degree of freedom when either group contains only zero counts.

Thus, rejection of the null hypothesis indicates a difference in the overall expression distribution arising from the zero probability, positive-count mean, or both.

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.dist(y)
```

`hurdle_LRT.mean` *Mean-Based Likelihood Ratio Test for Hurdle Negative Binomial Model*

Description

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

Usage

```
hurdle_LRT.mean(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 at estimates obtained from `tagwiseEst`.
- 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.mean(y)
```

`hurdle_Wald_Test.dist` *Distributional Wald Test for Hurdle Negative Binomial Model*

Description

Performs gene-wise Wald tests for distributional differential expression using a hurdle negative binomial model. The test jointly evaluates differences in the zero probability and positive-count mean between groups while holding the tag-wise dispersion fixed.

Usage

```
hurdle_Wald_Test.dist(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` and `size.factor`.
	tagwise.disp Numeric vector of estimated tag-wise dispersions.

Details

For each gene:

- When both groups contain positive counts, the model estimates group-specific nonzero probabilities and mean parameters while fixing the dispersion at its tag-wise estimate.
- A joint two-degree-of-freedom Wald test evaluates the nonzero probability and positive-count mean differences between groups.
- When either group contains only zero counts, the test compares only the nonzero probabilities using corrected log odds, with 0.5 added to each positive- and zero-count frequency.
- The resulting squared Z statistic is compared with a chi-squared distribution with one degree of freedom in the all-zero-group case.

Rejection of the null hypothesis indicates a difference in the overall expression distribution arising from the zero probability, positive-count mean, or both.

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.dist(y)
```

---

hurdle_Wald_Test.mean *Mean-Based Wald Test for Hurdle Negative Binomial Model*

Description

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

Usage

```
hurdle_Wald_Test.mean(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` and `size.factor`.
	tagwise.disp Numeric vector of estimated tag-wise dispersions.
	zero_prob_matrix Matrix of estimated zero probabilities for each gene and group.

Details

For each gene:

- The model estimates group-specific mean parameters while fixing the zero probabilities and dispersion at estimates obtained from `tagwiseEst`.
- When both groups contain positive counts, a two-sided Wald test is applied to the log mean difference between groups.
- When either group contains only zero counts, a one-sided Z-test is applied to the estimated log mean of the nonzero group using the corresponding Hessian-based standard error.

Value

The input object y with two additional elements:

wald_stats Numeric vector of Wald or one-sided Z 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.mean(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

`data`	A numeric matrix, data.frame, or supported object containing counts.
`group`	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", "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
}
```

---

priorEst *Estimate Bin-wise Priors for Hurdle Model Parameters (Internal)*

Description

Internal function to group genes by similar mean expression levels and perform bin-wise estimation of prior parameters for the hurdle negative binomial model. Within each bin, genes are pooled and a global model is fitted to estimate group-specific non-zero probabilities and log dispersion.

Usage

```
priorEst(y, n_bins = NULL)
```

Arguments

y	A list object produced by `sizeFactorsEst()`, containing `counts`, `samples`, and `baseMean`.
n_bins	Optional integer specifying the number of bins. If `NULL`, the number of bins is determined automatically.

Details

Genes are partitioned into bins based on `log(baseMean)`. For each bin, a hurdle negative binomial model is fitted using `nll_hurdle()` on pooled counts to obtain bin-level estimates of:

- Group-specific non-zero probabilities
- Log-dispersion parameter (log(phi))

The estimated parameters are mapped back to individual genes to provide stabilized priors for downstream modeling.

Value

The input object y with added components:

prob_matrix Matrix of group-specific non-zero probabilities.

prior_log_phi_gene Gene-wise log(phi) estimates.

prior_bins Data.frame of bin-level parameter estimates.

---

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 (`matrix` or `data.frame`) or the output of `prepareDGE()`. If a matrix/data.frame 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 `median`.

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, small.sample = TRUE, prior = TRUE)
```

Arguments

`y`	A list object created by `prepareDGE` with size factors estimated, containing counts and sample information.
`small.sample`	Logical. If `TRUE`, borrows information from genes with similar expression patterns. If `FALSE`, estimates dispersion using each gene independently. Setting `FALSE` is recommended for adaptive single-cell differential expression analysis.
`prior`	Logical. Whether to use prior information for dispersion shrinkage when `small.sample = TRUE`. This argument is ignored when `small.sample = FALSE`.

Details

When `small.sample = TRUE`, the function calls `tagwiseEst.smallSample`, which 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.

When `small.sample = FALSE`, the function calls `tagwiseEst.largeSample`, which fits a hurdle negative binomial model separately to each gene without borrowing information across genes.

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

∗ **internal**
priorEst, 11
∗ **package**
LRDE-package, 2

hurdle.LRT, *2, 3*, 3
hurdle.Wald.Test, *2, 3*, 4
hurdle_LRT.dist, *4*, 5
hurdle_LRT.mean, *4*, 6
hurdle_Wald_Test.dist, *5*, 8
hurdle_Wald_Test.mean, *5*, 9

LRDE *(*LRDE-package*)*, 2
LRDE-package, 2

median, *12*

prepareDGE, *2, 3*, 10, *12, 13*
priorEst, 11, *13*

sizeFactorsEst, *2, 3*, 12

tagwiseEst, *2–9*, 13