Package 'countfitteR'

July 1, 2025

Type Package

Title Comprehensive Automatized Evaluation of Distribution Models for Count Data

Version 1.5

Maintainer Jaroslaw Chilimoniuk <`jaroslaw.chilimoniuk@gmail.com`>

Description A large number of measurements generate count data. This is a statistical data type that only assumes non-negative integer values and is generated by counting. Typically, counting data can be found in biomedical applications, such as the analysis of DNA double-strand breaks. The number of DNA double-strand breaks can be counted in individual cells using various bioanalytical methods. For diagnostic applications, it is relevant to record the distribution of the number data in order to determine their biomedical significance (Roediger, S. et al., 2018. Journal of Laboratory and Precision Medicine. <doi:10.21037/jlpm.2018.04.10>). The software offers functions for a comprehensive automated evaluation of distribution models of count data. In addition to programmatic interaction, a graphical user interface (web server) is included, which enables fast and interactive data-scientific analyses. The user is supported in selecting the most suitable counting distribution for his own data set.

License GPL-3

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Suggests dplyr, DT, gridExtra, knitr, pander, reshape2, rmarkdown, shinythemes, shinycssloaders, shinyWidgets, spelling, testthat

Date 2025-07-01

URL `https://github.com/BioGenies/countfitteR`

BugReports `https://github.com/BioGenies/countfitteR/issues`

RoxygenNote 7.3.2

Imports ggplot2, MASS, shiny, stats, pscl, tools, utils

Language en-US

NeedsCompilation no

Author Jaroslaw Chilimoniuk [cre, ctb] (ORCID: <https://orcid.org/0000-0001-5467-018X>),
Alicja Gosiewska [ctb] (ORCID: <https://orcid.org/0000-0001-6563-5742>),
Jadwiga Słowik [ctb] (ORCID: <https://orcid.org/0000-0003-3466-8933>),
Michal Burdukiewicz [aut] (ORCID: <https://orcid.org/0000-0001-8926-582X>),
Stefan Roediger [ctb] (ORCID: <https://orcid.org/0000-0002-1441-6512>)

Repository CRAN

Date/Publication 2025-07-01 10:30:11 UTC

Contents

`countfitteR-package` *countfitteR - a framework for fitting count distributions in R*

Description

The `countfitteR` package is a toolbox for the analysis of count data.

Acknowledgements

countfitteR is a wrapper around existing count models in R. To standardize error messages and ease up the integration, we slightly modified the `zeroinfl` function by Achim Zeileis.

Author(s)

Jaroslaw Chilimoniuk, Stefan Roediger, Michal Burdukiewicz

See Also

Useful links:

- `https://github.com/BioGenies/countfitteR`
- Report bugs at `https://github.com/BioGenies/countfitteR/issues`

Examples

```
set.seed(15390)
library(countfitteR)
df <- data.frame(pois = rpois(25, 0.3),
                 binom = rbinom(25, 1, 0.8))

cmp <- compare_fit(df, fitlist = fit_counts(df, model = "all"))
```

`case_study` *Short version of the* `case_study_FITC`

Description

shorter version of the `case_study_FITC`. Used as an example in shiny app, when the user will not load his own count data.

Usage

```
case_study
```

`case_study_all` *Case study with two fluorescent dyes*

Description

example data extracted from Aklides system and merged into one file. Counts in this file will not fit properly, due to the fact that we integrated into the file counts with two different fluorescent dyes used.

Usage

```
case_study_all
```

`case_study_APC` *Case study for APC dye*

Description

example data extracted from Aklides system. Counts with only APC fluorescent dye were merged.

Usage

```
case_study_APC
```

`case_study_FITC` *Case study for FITC dye*

Description

example data extracted from Aklides system. Counts with only FITC fluorescent dye were merged.

Usage

```
case_study_FITC
```

`compare_fit` *Compare fits*

Description

Compare empirical distribution of counts with the distribution defined by the model fitted to counts.

Usage

```
compare_fit(count_list, fitlist = fit_counts(count_list, model = "all"))
```

Arguments

`count_list`	A `list` of counts. Each count should be in separate column, rows should represent values of these counts.
`fitlist`	a list of fits, as created by `fit_counts`.

Value

A `data.frame` with distribution values for each unique count. Count is the name of the original count, model is the name of distribution model, x is unique count value, n is the frequency of unique counts, value is result of calculations made by chosen distribution model.

Examples

```
df <- data.frame(poisson = rpois(25, 0.3), binomial = rbinom(25, 1, 0.8))
compare_fit(df, fitlist = fit_counts(df, model = "all"))
```

`countfitteR_gui` *countfitteR Graphical User Interface*

Description

Launches graphical user interface that analyses given count data and chooses the best performing distribution model.

Usage

```
countfitteR_gui()
```

Warning

Any ad-blocking software may cause malfunctions.

Author(s)

Jaroslaw Chilimoniuk, Stefan Roediger, Michal Burdukiewcz

Examples

```
if(interactive()) {
  countfitteR_gui()
}
```

`decide` *Make a decision based on the BIC value*

Description

Select the most appropriate distribution for the count data in the html-friendly format.

Usage

```
decide(summary_fit, separate)
```

Arguments

`summary_fit`	a result of the `summary_fitlist` function.
`separate`	`logical`. If `TRUE`, each count is separately fitted to the model. If `FALSE`, all counts are fitted to the same models having the count name as the independent variable.

See Also

`fit_counts`

Examples

```
df <- data.frame(poisson = rpois(25, 0.3), binomial = rbinom(25, 1, 0.8))
fc <- fit_counts(df, model = "all")
summ <- summary_fitlist(fc)
decide(summ, separate = FALSE)
```

`fit_counts` *Fit counts to distributions*

Description

Fit counts to distributions

Usage

```
fit_counts(counts_list, separate = TRUE, model, level = 0.95, ...)
```

Arguments

`counts_list`	A `list` of count data. Each count should be in separate column, rows should represent values of that counts.
`separate`	`logical`. If `TRUE`, each count is separately fitted to the model. If `FALSE`, all counts are fitted to the same models having the count name as the independent variable.
`model`	single `character`: `"pois"`, `"nb"`, `"zinb"`, `"zip"`, `"all"`. If `"all"`, all possible model are fitted.
`level`	Confidence level, default is 0.95.
`...`	Dots parameters are ignored.

Value

The list of fitted models. Names are names of original counts, an underline and a name of model used. confint is a `matrix` with the number of rows equal to the number of parameters. Rownames are names of parameters. The columns contain respectively lower and upper confidence intervals.

Examples

```
df <- data.frame(poisson = rpois(25, 0.3), binomial = rbinom(25, 1, 0.8))
fit_counts(df, model = "pois")
```

`plot_fitcmp` *plot_fitcmp*

Description

Compare empirical distribution of counts with the distribution defined by the model fitted to counts. The bar charts represent theoretical counts depending on the chosen distribution. Red dots describe the real number of counts.

Usage

```
plot_fitcmp(fitcmp)
```

Arguments

`fitcmp`	You need to input data frame that is created by compare_fit function.

Examples

```
df <- data.frame(poisson = rpois(25, 0.3), binomial = rbinom(25, 1, 0.8))
fitcmp <- compare_fit(df, fitlist = fit_counts(df, model = "all"))
plot_fitcmp(fitcmp)
```

`process_counts` *Process counts*

Description

Converts data in a table-like formats into lists of counts.

Usage

```
process_counts(x)
```

Arguments

`x`	`data.frame` or `matrix`.

Details

`case_study` does not consider `NA`s and `NaN`s effectively omitting them (as per the `is.na` function).

Value

A `list` of counts.

Examples

```
data(case_study)
process_counts(case_study)
```

`select_model` *Select the most appropriate model*

Description

Select the most appropriate model

Usage

```
select_model(fitlist)
```

Arguments

`fitlist`	a list of fits, as created by `fit_counts`.

Value

a `data.frame` with two columns: `count` representing the name of the count and `chosen model` with the model with the lowest BIC.

Examples

```
set.seed(1)
df <- data.frame(poisson1 = rpois(50, 2),
                 poisson2 = rpois(50, 5),
                 zip1 = rZIP(50, 2, 0.7),
                 zip2 = rZIP(50, 5, 0.7))
fitlist_separate <- fit_counts(df, model = c("pois", "zip"))
select_model(fitlist_separate)
```

`sim_dat` *Data created from simulation of NB Poiss*

Description

Data created from simulation of NB Poiss

Usage

```
sim_dat
```

Examples

```
# code used to generate the data
# be warned: the simulations will take some time
## Not run:
library(dplyr)
set.seed(15390)
sim_dat <- do.call(rbind, lapply(10^(-3L:2), function(single_theta)
  do.call(rbind, lapply(1L:10/2, function(single_lambda)
    do.call(rbind, lapply(1L:100, function(single_rep) {

    foci <- lapply(1L:10, function(dummy) rnbinom(600, size = single_theta, mu = single_lambda))
      names(foci) <- paste0("C", 1L:10)

      fit_counts(foci, separate = TRUE, model = "all") %>%
        summary_fitlist %>%
        mutate(between = single_lambda < upper & single_lambda > lower) %>%
        group_by(model) %>%
        summarize(prop = mean(between)) %>%
        mutate(replicate = single_rep, lambda = single_lambda, theta = single_theta)
    }))
  ))
))

## End(Not run)
```

`summary_fitlist` *Summary of estimates*

Description

Counts are fitted to model(s) using the count name as the explanatory variable. Estimates are presented in the table below along with the BIC values of their models. Estimated coefficients of models (`lambda` for all distributions, `theta` for NB and ZINB, `r` for ZIP and ZINB).

Usage

```
summary_fitlist(fitlist)
```

Arguments

`fitlist`	a list of fits, as created by `fit_counts`.

Value

Data frame with summarised results of all distribution models.

See Also

`fit_counts`

Examples

```
df <- data.frame(poisson = rpois(25, 0.3), binomial = rbinom(25, 1, 0.8))
fc <- fit_counts(df, model = "all")
summary_fitlist(fc)
```

`validate_counts` *Validate data*

Description

Validates count data.

Usage

```
validate_counts(x)
```

Arguments

`x`	`data.frame` or `matrix`.

Details

Errors if `x` has negative values or non-numeric values, otherwise `TRUE`.

Value

An input object.

Examples

```
data(case_study)
process_counts(case_study)
```

`zinb` *Zero-inflated negative binomial distrbution*

Description

Density and random generation for the zero-inflated negative binomial distribution.

Usage

```
rZINB(n, size, mu, r)

dZINB(x, size, mu, r)
```

Arguments

`n`	number of random values to return.
`size`	target for number of successful trials, or dispersion parameter (the shape parameter of the gamma mixing distribution). Must be strictly positive, need not be integer..
`mu`	mean.
`r`	probability of excess zeros.
`x`	vector of (non-negative integer) quantiles.

See Also

Negative binomial distribution: `NegBinomial`.

Examples

```
rZINB(15, 1.9, 0.9, 0.8)
```

`zip` *Zero-inflated Poisson distrbution*

Description

Density and random generation for the zero inflated Poisson distribution.

Usage

```
dZIP(x, lambda, r)

rZIP(n, lambda, r)
```

Arguments

`x`	vector of (non-negative integer) quantiles.
`lambda`	vector of (non-negative) means.
`r`	probability of excess zeros.
`n`	number of random values to return.

See Also

Poisson distribution: `Poisson`.

Examples

```
rZIP(15, 1.9, 0.9)
```

Index