

Package ‘countfitter’

February 3, 2019

Type Package

Title Comprehensive Automatized Evaluation of Distribution Models for Count Data

Version 1.0

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, utils, testthat, shinythemes, rhandsontable, gridExtra, pscl, DT, rmarkdown, knitr

Date 2019-01-18

URL <https://github.com/jarochi/countfitter>

BugReports <https://github.com/jarochi/countfitter/issues>

RoxygenNote 6.1.1

Imports ggplot2, MASS, shiny, stats, tools

NeedsCompilation no

Author Jaroslaw Chilimoniuk [cre, ctb]
 (<<https://orcid.org/0000-0001-5467-018X>>),
 Michal Burdukiewicz [aut] (<<https://orcid.org/0000-0001-8926-582X>>),
 Stefan Roediger [ctb] (<<https://orcid.org/0000-0002-1441-6512>>)

Repository CRAN

Date/Publication 2019-02-03 15:33:19 UTC

R topics documented:

| | |
|-------------------------------|---|
| countfitter-package | 2 |
| case_study | 3 |
| case_study_all | 3 |
| case_study_APC | 3 |
| case_study_FITC | 4 |
| compare_fit | 4 |
| countfitter_gui | 5 |
| fit_counts | 5 |
| plot_fitcmp | 6 |
| process_counts | 6 |
| sim_dat | 7 |
| summary_fitlist | 8 |
| zinb | 8 |
| zip | 9 |

Index **10**

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

| | |
|------------|---|
| case_study | <i>Short version of the case_study_FITC</i> |
|------------|---|

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 | <i>Case study with two fluorescent dyes</i> |
|----------------|---|

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 | <i>Case study for APC dye</i> |
|----------------|-------------------------------|

Description

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

Usage

case_study_APC

| | |
|-----------------|--------------------------------|
| case_study_FITC | <i>Case study for FITC dye</i> |
|-----------------|--------------------------------|

Description

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

Usage

```
case_study_FITC
```

| | |
|-------------|------------------------------|
| compare_fit | <i>Compare distributions</i> |
|-------------|------------------------------|

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 | Uses fit_count function to calculate value for each unique observation using different distribution models. |

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 | <i>countfitter Graphical User Interface</i> |
|-----------------|---|

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 Burdukiewicz

| | |
|------------|--|
| fit_counts | <i>Function that fits counts to distribution models.</i> |
|------------|--|

Description

Function that fits counts to distribution models.

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 | <i>plot_fitcmp</i> |
|-------------|--------------------|

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 | <i>Process counts</i> |
|----------------|-----------------------|

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 NAs and NaNs effectively omitting them (as the [is.na](#) function).

Value

A list of counts.

Examples

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

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 | <i>Summary of estimates</i> |
|-----------------|-----------------------------|

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 | List is created by fitlist = fit_counts(count_list, model = "all"). For each count it creates distribution model |
|---------|--|

Value

Data frame with summarised results of all distribution models.

- Count the name of the original count.
- lambda λ - Poisson mean, lower and upper confidence intervals.
- BIC Bayesian information criterion
- theta θ - dispersion parameter
- r probability of excess zeros.

Examples

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

| | |
|------|---|
| zinb | <i>Zero-inflated negative binomial distribution</i> |
|------|---|

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 | <i>Zero-inflated Poisson distribution</i> |
|-----|---|

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

- *Topic **Poisson**
 - countfitter_gui, 5
- *Topic **count**
 - countfitter_gui, 5
- *Topic **datasets**
 - case_study, 3
 - case_study_all, 3
 - case_study_APC, 3
 - case_study_FITC, 4
 - sim_dat, 7
- *Topic **zero-inflated**
 - countfitter_gui, 5

- case_study, 3
- case_study_all, 3
- case_study_APC, 3
- case_study_FITC, 4
- compare_fit, 4
- countfitter (countfitter-package), 2
- countfitter-package, 2
- countfitter_gui, 5

- dZINB (zinb), 8
- dZIP (zip), 9

- fit_counts, 5

- is.na, 7

- NegBinomial, 9

- plot_fitcmp, 6
- Poisson, 9
- process_counts, 6

- rZINB (zinb), 8
- rZIP (zip), 9

- sim_dat, 7
- summary_fitlist, 8

- zeroinfl, 2

- ZINB (zinb), 8
- zinb, 8
- ZIP (zip), 9
- zip, 9