Package ‘SLDAssay’

November 2, 2017

Type Package
Title Software for Analyzing Limiting Dilution Assays
Version 1.8
Date 2017-11-01
Author Michael Hudgens, Ilana Trumble, Andrew Allmon
Maintainer Ilana Trumble <itrumble@unc.edu>
Description Calculates maximum likelihood estimate, exact and asymptotic confidence intervals, and exact and asymptotic goodness of fit p-values for concentration of infectious units from serial limiting dilution assays. This package uses the likelihood equation, exact goodness of fit p-values, and exact confidence intervals described in Meyers et al. (1994) <http://jcm.asm.org/content/32/3/732.full.pdf>. This software is also implemented as a web application through the Shiny R package <https://iupm.shinyapps.io/sldassay/>.
License GPL-3
LazyData TRUE
Imports compiler
Depends R (>= 3.2.1)
RoxygenNote 5.0.1
NeedsCompilation no
Repository CRAN
Date/Publication 2017-11-02 04:10:21 UTC

R topics documented:

get.mle ................................................................. 2

Index 4
get.mle

**IUPM, PGOF, and CI**

**Description**

Calculates the maximum likelihood estimate of concentration of infectious units from a single serial limiting dilution (SLD) assay. Also calculates corresponding exact and asymptotic confidence intervals, and a goodness-of-fit p-value. While this package was developed with the purpose of estimating IUPM, it is applicable to SLD assays in general.

**Usage**

get.mle(pos, replicates, dilutions, monte = 15000, conf.level = 0.95, iupm = TRUE, na.rm = FALSE)

**Arguments**

- **pos**
  Vector of number of positive wells at each dilution level (outcome of SLD Assay)
- **replicates**
  Vector of number of replicates at each dilution level
- **dilutions**
  Vector of number of cells per well at each dilution level
- **monte**
  Number of Monte Carlo samples. Default is exact (no MC sampling), unless more than 15,000 possible positive well outcomes exist, in which case 15,000 MC samples are taken. Use monte=F for exact computation.
- **conf.level**
  Confidence level of the interval.
- **iupm**
  Boolean variable, indicates whether to return MLE as IUPM (TRUE) or probability a cell is infected (FALSE)
- **na.rm**
  Boolean variable, indicates whether dilution levels valued NA should be stripped before the computation proceeds (FALSE)

**Value**

- **MLE**
  Maximum likelihood estimate for the given outcome vector.
- **BC_MLE**
  Bias corrected maximum likelihood estimate for the given outcome vector.
- **Exact_PGOF**
  P value for goodness of fit. PGOF is the probability of an experimental result as rare as or rarer than that obtained, assuming that the model is correct. Low values of PGOF, (e.g. PGOF < 0.01), indicate rare or implausible experimental results. Samples with a very low PGOF might be considered for retesting.
- **Asymp_PGOF**
  P value calculated using an asymptotic Chi-Squared distribution with D-1 degrees of freedom, where D is the number of dilution levels in an SLD assay.
- **Exact_CI**
  Exact confidence interval, computed from the likelihood ratio test (recommended)
- **Asymp_CI**
  Wald asymptotic confidence interval, based on the normal approximation to the binomial distribution.
References

Examples

# Duplicates row 4 of Table 4 from Myers, et. al.
# Myers et. al. divides IUPM space into discrete values. This package searches
# entire parameter space, yielding a slightly different and more accurate MLE.
row4 <- get.mle(pos=c(2,1,0,0,0), replicates=rep(2,6), dilutions=c(1e6,2e5,4e4,8e3,1600,320), conf.level=0.95, iupm=TRUE, # Display MLE in infected units per million
)

# Duplicates row 21 of Table 4 from Myers, et. al.
# Low PGOF example
# Myers et. al. divides IUPM space into discrete values. This package searches
# entire parameter space, yielding a slightly different and more accurate MLE.
row21 <- get.mle(pos=c(2,2,2,1,0), replicates=rep(2,6), dilutions=c(1e6,2e5,4e4,8e3,1600,320), conf.level=0.95, iupm=TRUE)

# Example calculating IUs per cell for an assay with 1 DL.
iu.example <- get.mle(pos=7, replicates=8, dilutions=25, conf.level=0.95, iupm=FALSE)

# Monte Carlo example
# 67,081 total possible positive well outcomes, therefore
# Monte Carlo sampling is used to reduce computation time.
MC.example <- get.mle(pos=c(30,9,1,0), replicates=c(36,36,6,6), dilutions=c(2.5e6,5e5,1e5,2.5e4), conf.level=0.95, monte = 5000, iupm=TRUE )
Index

get.mle, 2