Package 'mlemur'

January 5, 2023

Type Package

Title MLE MUtatution Rate Calculator

Version 0.9.6

Date 2023-01-05

Author Krystian Lazowski [aut, cre] (<https://orcid.org/0000-0001-5163-7026>), Qi Zheng [ctb]

Maintainer Krystian Lazowski <klazowski@ibb.waw.pl>

Description A tool for calculating maximum likelihood-based mutation rates, confidence intervals, and P values. Offers graphical user interface built with shiny. Based on rSalvador by Qi Zheng <https://github.com/eeeeeric/rSalvador>.

License GPL (>= 2)

SystemRequirements gmp (>=6.2.1), mpfr (>=4.1.0)

SystemRequirementsNote 'MPFR' (MP Floating-Point Reliable Library, https://www.mpfr.org/) and 'GMP' (GNU Multiple Precision library, https://gmplib.org/), see >> README.md

Depends R (>= 4.1.0), shiny

Imports Rcpp (>= 1.0.9), stats, utils, shinyFeedback, shinyWidgets, shinyjqui, shinyjs, writexl, readxl, clipr, reactable, Ryacas, boot

LinkingTo Rcpp, BH

RoxygenNote 7.2.3

NeedsCompilation yes

StagedInstall yes

Encoding UTF-8

R topics documented:

`mlemur-package` *mlemur: MLE Mutation Rate calculator*

Description

A tool for calculating maximum likelihood-based mutation rates, confidence intervals, and P values. Offers graphical user interface built with shiny.

Details

Based on rSalvador by Qi Zheng <https://github.com/eeeeeric/rSalvador>; GPL 2.0

Author(s)

Krystian Lazowski, klazowski@ibb.waw.pl

contributor: Qi Zheng

Maintainer: Krystian Lazowski <klazowski@ibb.waw.pl>

`drake.formula` *Estimate m using Drake method*

Description

The basic formula comes from the Drake paper. Corrections for partial plating, phenotypic delay (**only non-stochastic**), residual mutations, and differential growth were taken from other Angerer and Koch. Correction for mutant death is based on the Angerer's rationale regarding phenotypic lag, with Newcombe correction for extra cellular divisions.

Usage

```
drake.formula(data, e = 1, w = 1, poisson = 0, lag = 0, death = 0)
```

Arguments

data	a vector of integer-valued colony counts.
e	plating efficiency; a positive number not bigger than 1.
w	mutant relative fitness; a positive number.
poisson	average number of residual Poisson-distributed mutations on the plate; a non-negative number.
lag	phenotypic lag (**only non-stochastic**); a non-negative number.
death	death probability of wild-type and mutant cells; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).

Value

a single non-negative value of m.

References

Newcombe HB. Delayed Phenotypic Expression of Spontaneous Mutations in Escherichia Coli. Genetics. 1948;33: 447–476. doi:10.1093/genetics/33.5.447

Koch AL. Mutation and growth rates from Luria-Delbrück fluctuation tests. Mutat Res - Fundam Mol Mech Mutagen. 1982;95: 129–143. doi:10.1016/0027-5107(82)90252-4

Drake JW. A constant rate of spontaneous mutation in DNA-based microbes. Proc Natl Acad Sci U S A. 1991;88: 7160–7164. doi:10.1073/pnas.88.16.7160

Angerer WP. A note on the evaluation of fluctuation experiments. Mutat Res - Fundam Mol Mech Mutagen. 2001;479: 207–224. doi:10.1016/S0027-5107(01)00203-2

Examples

```
drake.formula(c(67, 12, 112, 24, 2151, 159, 102, 60, 32, 26))
```

---

jones.median *Estimate m using Jones method of the median*

Description

The algorithm comes from rSalvador (jones.median.plating). It is included for the sake of self-containedness.

Usage

```
jones.median(data, eff = 1)
```

Arguments

data	a vector of integer-valued colony counts.
eff	plating efficiency; a positive number not bigger than 1.

Value

a single non-negative value of m.

References

Zheng Q. rSalvador: An R package for the fluctuation experiment. G3 Genes, Genomes, Genet. 2017;7: 3849–3856. doi:10.1534/g3.117.300120

Examples

```
jones.median(c(67, 12, 112, 24, 2151, 159, 102, 60, 32, 26))
```

---

koch.quartiles *Estimate m using Koch method of quartiles*

Description

The formula comes from Rosche and Foster (2000).

Usage

```
koch.quartiles(data, lag = 0)
```

Arguments

data	a vector of integer-valued colony counts.
lag	phenotypic lag; a non-negative number.

Value

a vector of length 4 containing mean m as well as first, second and third quartile.

References

Rosche WA, Foster PL. Determining Mutation Rates in Bacterial Populations. Methods. 2000;20: 4–17. doi:10.1006/meth.1999.0901

Examples

```
koch.quartiles(c(67, 12, 112, 24, 2151, 159, 102, 60, 32, 26))
```

`lea.coulson.median` *Estimate m using Lea-Coulson method*

Description

The basic formula comes from a well-known paper by Lea & Coulson. Corrections for partial plating, phenotypic delay (**only non-stochastic**), residual mutations, and differential growth were taken from other Angerer and Koch. Correction for mutant death is based on the Angerer's rationale regarding phenotypic lag, with Newcombe correction for extra cellular divisions.

Usage

```
lea.coulson.median(
  data,
  e = 1,
  w = 1,
  poisson = 0,
  lag = 0,
  death = 0,
  confint = FALSE
)
```

Arguments

`data`	a vector of integer-valued colony counts.
`e`	plating efficiency; a positive number not bigger than 1.
`w`	mutant relative fitness; a positive number.
`poisson`	average number of residual Poisson-distributed mutations on the plate; a non-negative number.
`lag`	phenotypic lag (**only non-stochastic**); a non-negative number.
`death`	death probability of wild-type and mutant cells; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
`confint`	if TRUE, 95 percent confidence intervals will be estimated by bootstrap using boot package.

Value

a single non-negative value of m, or a vector of length 3 containing estimate of m and lower and upper limits of 95 percent CI.

References

Lea DE, Coulson CA. The distribution of the numbers of mutants in bacterial populations. J Genet. 1949;49: 264–285. doi:10.1007/BF02986080

Newcombe HB. Delayed Phenotypic Expression of Spontaneous Mutations in Escherichia Coli. Genetics. 1948;33: 447–476. doi:10.1093/genetics/33.5.447

Koch AL. Mutation and growth rates from Luria-Delbrück fluctuation tests. Mutat Res - Fundam Mol Mech Mutagen. 1982;95: 129–143. doi:10.1016/0027-5107(82)90252-4

Angerer WP. A note on the evaluation of fluctuation experiments. Mutat Res - Fundam Mol Mech Mutagen. 2001;479: 207–224. doi:10.1016/S0027-5107(01)00203-2

Examples

```
lea.coulson.median(c(67, 12, 112, 24, 2151, 159, 102, 60, 32, 26))
```

`lrt.mutations` *Calculate p-values using Likelihood Ratio Test*

Description

This function calculates LRT-based p-values to assess the statistical significance of the differences between two mutation rates (X and Y). It is loosely based on LRT.LD and LRT.LD.plating functions from rSalvador, simplified and optimised to avoid redundant computations. It uses a hybrid Newton-bisection algorithm as well as arbitrary-precision arithmetic if necessary for better stability.

Usage

```
lrt.mutations(
  datax,
  datay,
  ex = 1,
  ey = 1,
  wx = 1,
  wy = 1,
  lagx = 0,
  lagy = 0,
  poissonx = 0,
  poissony = 0,
  deathx = 0,
  deathy = 0,
  phix = 0,
  phiy = 0,
  cvx = 0,
  cvy = 0,
  Nx = 1,
  Ny = 1,
  Mx = NA,
  My = NA,
  verbose = FALSE
)
```

Arguments

`datax`	a vector of integer-valued colony counts for strain X.
`datay`	a vector of integer-valued colony counts for strain Y.
`ex`	plating efficiency for strain X; a positive number not bigger than 1.
`ey`	plating efficiency for strain Y; a positive number not bigger than 1.
`wx`	mutant relative fitness for strain X; a positive number.

wy	mutant relative fitness for strain Y; a positive number.
lagx	phenotypic lag for strain X; a non-negative number.
lagy	phenotypic lag for strain Y; a non-negative number.
poissonx	average number of residual Poisson-distributed mutations on the plate for strain X; a non-negative number.
poissony	average number of residual Poisson-distributed mutations on the plate for strain Y; a non-negative number.
deathx	death probability of wild-type and mutant cells for strain X; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
deathy	death probability of wild-type and mutant cells for strain Y; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
phix	relative size of the inoculum (N0/Nt) for strain X; a non-negative number.
phiy	relative size of the inoculum (N0/Nt) for strain Y; a non-negative number.
cvx	coefficient of variation of the final number of cells in each culture for strain X.
cvy	coefficient of variation of the final number of cells in each culture for strain Y.
Nx	average number of cells in culture for strain X.
Ny	average number of cells in culture for strain Y.
Mx	if known, MLE of m for strain X can be put here to speed up computations; mostly for internal use.
My	if known, MLE of m for strain Y can be put here to speed up computations; mostly for internal use.
verbose	if TRUE, mlemur will print messages to the console.

Value

a single value of p-value between 0 and 1.

References

Zheng Q. Statistical and algorithmic methods for fluctuation analysis with SALVADOR as an implementation. Math Biosci. 2002;176: 237–252. doi:10.1016/S0025-5564(02)00087-1

Zheng Q. New algorithms for Luria-Delbrück fluctuation analysis. Math Biosci. 2005;196: 198–214. doi:10.1016/j.mbs.2005.03.011

Zheng Q. On Bartlett's formulation of the Luria-Delbrück mutation model. Math Biosci. 2008;215: 48–54. doi:10.1016/j.mbs.2008.05.005

Zheng Q. Comparing mutation rates under the Luria–Delbrück protocol. Genetica. 2016;144: 351–359. doi:10.1007/s10709-016-9904-3

Zheng Q. rSalvador: An R package for the fluctuation experiment. G3 Genes, Genomes, Genet. 2017;7: 3849–3856. doi:10.1534/g3.117.300120

Examples

```
lrt.mutations(datax = c(26, 9, 16, 34, 15, 25, 77, 13, 14, 19), ex = 0.5,
datay = c(67, 12, 112, 24, 2151, 159, 102, 60, 32, 26))
lrt.mutations(datax = c(37, 1, 5, 43, 53, 11, 82, 2, 19, 58, 28, 70, 9, 14, 5,
9, 25, 55, 2, 41, 19), datay = c(37, 1, 5, 43, 53, 11, 82, 2, 19, 58, 28,
70, 9, 14, 5, 9, 25, 55, 2, 41, 19), lagy = 2)
```

`lrt.rate` *Calculate p-values using Likelihood Ratio Test using pairs of colony counts on selective and non-selective medium*

Description

This function calculates LRT-based p-values to assess the statistical significance of the differences between two mutation rates (1 and 2). The algorithm uses pairs of colony counts on selective and non-selective medium for each test tube. It uses a hybrid Newton-bisection algorithm as well as arbitrary-precision arithmetic if necessary for better stability.

Usage

```
lrt.rate(
  data1,
  data2,
  e1 = 1,
  e2 = 1,
  w1 = 1,
  w2 = 1,
  lag1 = 0,
  lag2 = 0,
  poisson_rate1 = 0,
  poisson_rate2 = 0,
  death1 = 0,
  death2 = 0,
  inoculum1 = 0,
  inoculum2 = 0,
  Nt1,
  Nt2,
  verbose = FALSE
)
```

Arguments

`data1`	a vector of integer-valued colony counts for strain 1.
`data2`	a vector of integer-valued colony counts for strain 2.
`e1`	plating efficiency for strain 1; a positive number not bigger than 1.
`e2`	plating efficiency for strain 2; a positive number not bigger than 1.
`w1`	mutant relative fitness for strain 1; a positive number.
`w2`	mutant relative fitness for strain 2; a positive number.
`lag1`	phenotypic lag for strain 1; a non-negative number.
`lag2`	phenotypic lag for strain 2; a non-negative number.
`poisson_rate1`	average mutation rate of Poisson-distributed mutations on the plate for strain 1; a non-negative number.
`poisson_rate2`	average mutation rate of Poisson-distributed mutations on the plate for strain 2; a non-negative number.

death1	death probability of wild-type and mutant cells for strain 1; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
death2	death probability of wild-type and mutant cells for strain 2; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
inoculum1	number of cells in the inoculum for strain 1; a non-negative integer
inoculum2	number of cells in the inoculum for strain 2; a non-negative integer
Nt1	a vector of integer-valued culture sizes for strain 1; must be of the same length as data.
Nt2	a vector of integer-valued culture sizes for strain 2; must be of the same length as data.
verbose	if TRUE, mlemur will print messages to the console.

Value

a single value of p-value between 0 and 1.

References

Zheng Q. Statistical and algorithmic methods for fluctuation analysis with SALVADOR as an implementation. Math Biosci. 2002;176: 237–252. doi:10.1016/S0025-5564(02)00087-1

Zheng Q. New algorithms for Luria-Delbrück fluctuation analysis. Math Biosci. 2005;196: 198–214. doi:10.1016/j.mbs.2005.03.011

Zheng Q. On Bartlett's formulation of the Luria-Delbrück mutation model. Math Biosci. 2008;215: 48–54. doi:10.1016/j.mbs.2008.05.005

Zheng Q. Comparing mutation rates under the Luria–Delbrück protocol. Genetica. 2016;144: 351–359. doi:10.1007/s10709-016-9904-3

Zheng Q. rSalvador: An R package for the fluctuation experiment. G3 Genes, Genomes, Genet. 2017;7: 3849–3856. doi:10.1534/g3.117.300120

Examples

```
lrt.rate(data1 = c(14, 20, 59, 46, 77, 38, 86, 17, 13, 25),
Nt1 = c(1030417500, 635899434, 721044961, 1496257034, 1276590167,
1977307240, 954768054, 1192512084, 632462708, 748020573), data2 = c(387, 630,
140, 50, 1187, 18, 161, 30, 32, 61), e2 = 0.5, Nt2 = c(3543784628,
2384685853, 4744886069, 2758252418, 4388557268, 1098581801, 4363192697,
1314405541, 1677598351, 3420509139))
```

---

luria.delbruck.mean *Estimate m using Luria-Delbruck method of the mean*

Description

The formula comes from Rosche and Foster (2000).

Usage

```
luria.delbruck.mean(data)
```

Arguments

data	a vector of integer-valued colony counts.

Value

a single non-negative value of m.

References

Rosche WA, Foster PL. Determining Mutation Rates in Bacterial Populations. Methods. 2000;20: 4–17. doi:10.1006/meth.1999.0901

Examples

```
luria.delbruck.mean(c(67, 12, 112, 24, 2151, 159, 102, 60, 32, 26))
```

---

mle.fold *Calculate profile likelihood confidence intervals for an arbitrary function of mutation rates*

Description

Inspired by Zheng 2021, this function calculates MLE and profile likelihood confidence intervals of an arbitrary function of data, such as fold (X1/X2), subtraction (X1-X2), fold with background subtraction ((X1-X2)/(X3-X2)), or a user-defined function utilising basic mathematical operations: addition, subtraction, multiplication or division. Up to 6 datasets can be used.

Usage

```
mle.fold(
  data,
  e = NULL,
  w = NULL,
  lag = NULL,
  poisson = NULL,
  death = NULL,
  phi = NULL,
  cv = NULL,
  Nt = NULL,
  fun = "fold X1/X2",
  ci.level = 0.95,
  verbose = FALSE
)
```

Arguments

data	a list of length 2 to 6 containing numeric vectors with colony counts.
e	a list or vector of length 2 to 6 (same as data) containing respective values of plating efficiency.
w	a list or vector of length 2 to 6 (same as data) containing respective values of relative mutant fitness.

lag	a list or vector of length 2 to 6 (same as data) containing respective values of phenotypic lag.
poisson	a list or vector of length 2 to 6 (same as data) containing respective values of the number of residual mutations.
death	a list or vector of length 2 to 6 (same as data) containing respective values of relative death rates.
phi	a list or vector of length 2 to 6 (same as data) containing respective values of inoculum sizes as a fraction of total culture sizes.
cv	a list or vector of length 2 to 6 (same as data) containing respective values of coefficient of variation.
Nt	a list or vector of length 2 to 6 (same as data) containing respective values of average number of cells in culture.
fun	describes the function of mutation rates for which confidence intervals should be calculated. Must be a character vector of length 1 containing either one of default options ("fold X1/X2", "subtraction X1-X2", "double fold (X1/X2)/(X3/X4)", "background subtraction fold (X1-X2)/(X3-X2)") or a user-defined function containing phrases X1, X2, X3, X4, X5, X6, which denote mutation rates for Strain 1, Strain 2, ... Only addition +, subtraction -, multiplication *, and division /, are currently supported.
ci.level	confidence interval size, default 0.95.
verbose	if TRUE, mlemur will print messages to the console.

Value

a list containing two elements: "fun", a vector of length 3 containg MLE of an arbitrary function of mutation rates as well as lower and upper profile likelihood confidence limits; "rates", a vector containing MLEs of rates used to calculate the value of fun.

References

Venzon DJ, Moolgavkar SH. A Method for Computing Profile-Likelihood-Based Confidence Intervals. Appl Stat. 1988;37: 87. doi:10.2307/2347496

Zheng Q. New approaches to mutation rate fold change in Luria–Delbrück fluctuation experiments. Math Biosci. 2021;335: 108572. doi:10.1016/j.mbs.2021.108572

Examples

```
mle.fold(list(c(26, 9, 16, 34, 15, 25, 77, 13, 14, 19), c(67, 12, 112, 24,
2151, 159, 102, 60, 32, 26)), e = c(0.5, 1))
mle.fold(list(c(33,17,15), c(1,4,10), c(45,86,156)), fun="X1*X2/X3")
```

`mle.mutations` *Estimate m using Maximum Likelihood Method and calculate 95 percent CI using inverted Likelihood Ratio Test*

Description

This function finds Maximum Likelihood Estimate of m, the average number of mutations in culture. Then it proceeds to find Likelihood Ratio-based confidence limits for m. It is loosely based on newton.LD, newton.LD.plating, newton.B0, confint.LD, confint.LD.plating, and confint.B0 functions from rSalvador, simplified and optimised to avoid redundant computations. It uses a hybrid Newton-bisection algorithm as well as arbitrary-precision arithmetic if necessary for better stability.

Usage

```
mle.mutations(
  data,
  e = 1,
  w = 1,
  lag = 0,
  poisson = 0,
  death = 0,
  phi = 0,
  cv = 0,
  confint = TRUE,
  ci.level = 0.95,
  verbose = FALSE
)
```

Arguments

data	a vector of integer-valued colony counts.
e	plating efficiency; a positive number not bigger than 1.
w	mutant relative fitness; a positive number.
lag	phenotypic lag; a non-negative number.
poisson	average number of residual Poisson-distributed mutations on the plate; a non-negative number.
death	death probability of wild-type and mutant cells; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
phi	relative size of the inoculum (N0/Nt); a non-negative number.
cv	coefficient of variation of the final number of cells in each culture.
confint	if TRUE (default), confidence intervals at ci.level will be calculated.
ci.level	confidence interval size, default 0.95.
verbose	if TRUE, mlemur will print messages to the console.

Value

a single non-negative value of m, or a vector of length 3 containing MLE of m as well lower and upper limit of 95 percent CI.

References

Zheng Q. Statistical and algorithmic methods for fluctuation analysis with SALVADOR as an implementation. Math Biosci. 2002;176: 237–252. doi:10.1016/S0025-5564(02)00087-1

Zheng Q. New algorithms for Luria-Delbrück fluctuation analysis. Math Biosci. 2005;196: 198–214. doi:10.1016/j.mbs.2005.03.011

Zheng Q. On Bartlett's formulation of the Luria-Delbrück mutation model. Math Biosci. 2008;215: 48–54. doi:10.1016/j.mbs.2008.05.005

Zheng Q. rSalvador: An R package for the fluctuation experiment. G3 Genes, Genomes, Genet. 2017;7: 3849–3856. doi:10.1534/g3.117.300120

Examples

```
mle.mutations(data = c(67, 12, 112, 24, 2151, 159, 102, 60, 32, 26))
mle.mutations(data = c(71, 19, 4, 32, 12, 74, 0, 35, 8, 13, 9, 5000, 31, 6, 10, 106, 0, 22, 4, 69,
30, 47, 237, 15, 74, 89, 135, 11, 30, 1), lag = 2, cv = 0.3, ci.level = 0.68)
```

`mle.rate` *Estimate mutation rate using Maximum Likelihood Method and calculate 95 percent CI using inverted Likelihood Ratio Test using pairs of colony counts on selective and non-selective medium*

Description

This function finds Maximum Likelihood Estimate of mu, the average mutation rate. Then it proceeds to find Likelihood Ratio-based confidence limits for mu. The algorithm uses pairs of colony counts on selective and non-selective medium for each test tube. It uses a hybrid Newton-bisection algorithm as well as arbitrary-precision arithmetic if necessary for better stability.

Usage

```
mle.rate(
  data,
  e = 1,
  w = 1,
  lag = 0,
  poisson_rate = 0,
  death = 0,
  inoculum = 0,
  Nt,
  confint = TRUE,
  ci.level = 0.95,
  verbose = FALSE,
  show_logprob = FALSE
)
```

Arguments

`data`	a vector of integer-valued colony counts.
`e`	plating efficiency; a positive number not bigger than 1.
`w`	mutant relative fitness; a positive number.
`lag`	phenotypic lag; a non-negative number.
`poisson_rate`	average mutation rate of Poisson-distributed mutations on the plate; a non-negative number smaller than 1.

death	death probability of wild-type and mutant cells; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
inoculum	number of cells in the inoculum; a non-negative integer.
Nt	a vector of integer-valued culture sizes; must be of the same length as data.
confint	if TRUE (default), confidence intervals at ci.level will be calculated.
ci.level	confidence interval size, default 0.95.
verbose	if TRUE, mlemur will print messages to the console.
show_logprob	if TRUE and confint is FALSE, the function will return the point estimate of mutation rate and the value of the log-likelihood function. This is mostly for internal use.

Value

a single non-negative value of m, or a vector of length 3 containing MLE of m as well lower and upper limit of 95 percent CI.

References

Zheng Q. Statistical and algorithmic methods for fluctuation analysis with SALVADOR as an implementation. Math Biosci. 2002;176: 237–252. doi:10.1016/S0025-5564(02)00087-1

Zheng Q. New algorithms for Luria-Delbrück fluctuation analysis. Math Biosci. 2005;196: 198–214. doi:10.1016/j.mbs.2005.03.011

Zheng Q. On Bartlett's formulation of the Luria-Delbrück mutation model. Math Biosci. 2008;215: 48–54. doi:10.1016/j.mbs.2008.05.005

Zheng Q. rSalvador: An R package for the fluctuation experiment. G3 Genes, Genomes, Genet. 2017;7: 3849–3856. doi:10.1534/g3.117.300120

Examples

```
mle.rate(data = c(14, 20, 59, 46, 77, 38, 86, 17, 13, 25), Nt = c(1030417500,
635899434, 721044961, 1496257034, 1276590167, 1977307240, 954768054,
1192512084, 632462708, 748020573))
```

---

mlemur *Run mlemur in a browser window*

Description

This function initializes mlemur in graphical mode.

Usage

```
mlemur(options = list(display.mode = "normal", launch.browser = T))
```

Arguments

options	arguments passed to shiny::shinyApp(options=...)

p0 *Estimate m using p0 method*

Description

Estimate m using p0 method

Usage

```
p0(data, e = 1, w = 1, d = 0, lag = 0, phi = 0, poisson = 0)
```

Arguments

data	a vector of integer-valued colony counts.
e	plating efficiency; a positive number not bigger than 1.
w	mutant relative fitness; a positive number.
d	death probability of wild-type and mutant cells; a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
lag	phenotypic lag; a non-negative number.
phi	relative size of the inoculum (N0/Nt); a non-negative number.
poisson	average number of residual Poisson-distributed mutations on the plate; a non-negative number.

Value

a single non-negative value of m.

Examples

```
p0(c(0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 3, 1, 0, 10, 0, 0, 0, 1))
```

power.est *Calculate power of likelihood ratio test*

Description

This function calculates theoretical power of the likelihood ratio test given prescribed mutation rates, final culture sizes, and sample sizes.

Usage

```
power.est(
  n1 = 30,
  n2 = NULL,
  rate1 = 1e-09,
  rate2 = 2e-09,
  Nt1 = 1e+09,
  Nt2 = 1e+09,
  e1 = 1,
  e2 = 1,
  w1 = 1,
  w2 = 1,
  lag1 = 0,
  lag2 = 0,
  death1 = 0,
  death2 = 0,
  phi1 = 0,
  phi2 = 0,
  cv1 = 0,
  cv2 = 0,
  poisson1 = 0,
  poisson2 = 0,
  verbose = FALSE
)
```

Arguments

n1	A vector containing sample size of the first strain. If longer than one, powr will be calculated for each pair of n1 and n2.
n2	A vector containing sample size of the second strain. If NULL, it will be taken as equal to n1.
rate1	Mutation rate of the first strain.
rate2	Mutation rate of the second strain.
Nt1	Average culture size of the first strain.
Nt2	Average culture size of the second strain.
e1	Plating efficiency of the first strain.
e2	Plating efficiency of the second strain.
w1	Relative mutant fitness of the first strain.
w2	Relative mutant fitness of the second strain.
lag1	Phenotypic lag of the first strain.
lag2	Phenotypic lag of the second strain.
death1	Relative death rate of the first strain.
death2	Relative death rate of the second strain.
phi1	Size of inoculum in proportion to the total culture size of the first strain.
phi2	Size of inoculum in proportion to the total culture size of the second strain.
cv1	Coefficient of variation of the culture sizes of the first strain.
cv2	Coefficient of variation of the culture sizes of the second strain.

poisson1	Average number of residual Poisson-distributed mutations on the plate of the first strain.
poisson2	Average number of residual Poisson-distributed mutations on the plate of the second strain.
verbose	if TRUE, mlemur will print messages to the console.

Value

A vector of length 1 or a matrix of size length(n1) x length(n2) containing values of the power.

References

Gudicha DW., Schmittmann VD. and Vermunt JK. Statistical power of likelihood ratio and Wald tests in latent class models with covariates. Behav. Res. Methods 2017;49: 1824–1837. doi:10.3758/s13428-016-0825-y

Self SG., Mauritsen RH. and Ohara J.Power Calculations for Likelihood Ratio Tests in Generalized Linear Models. Biometrics 1992;48: 31. doi:10.2307/2532736

Examples

```
power.est(n1=15, n2 = 10, rate1 = 1e-9, rate2 = 2e-9, Nt1 = 1e9, Nt2 = 5e8)
power.est(n1=c(10, 20, 30), rate1 = 1e-9, rate2 = 5e-9, Nt1 = 1e9, Nt2 = 1e9, e1 = 1, e2 = 0.1)
```

---

rluria *Simulate fluctuation test*

Description

This function simulates a fluctuation assay consisting of n cultures, starting with N0 wild-type cells growing exponentially until reaching Nt wild-type cells. If cv is not 0, the number of wild-type cells per culture is drawn from gamma distribution. Wild-type cell growth is taken as deterministic while mutant cell growth is taken as stochastic according to a simple birth-and-death process. If plating efficiency is less than perfect, the success of plating each mutant cell is simulated using binomial distribution.

Usage

```
rluria(
  n = 10,
  rate = 1e-08,
  N0 = 1,
  Nt = 1e+09,
  mut_fit = 1,
  death_prob = 0,
  lag = 0,
  e = 1,
  cv = 0,
  trim = 0,
  ret = "list"
)
```

Arguments

n	number of parallel cultures in the experiment; a positive integer.
rate	average mutation rate; a positive number smaller than 1.
N0	size of inoculum; a positive integer.
Nt	final number of cells in culture; a positive integer.
mut_fit	relative fitness of the mutant cells vs. wild-type cells; a positive number. Should not be used together with lag.
death_prob	death probability (same for mutants and wild-type cells); a non-negative number smaller than 0.5; relative (d)eath rate and death (p)robability are connected by the relation d = p/(1-p); p = d/(1+d).
lag	mean phenotypic lag of mutant cells in generations; a non-negative number. Should not be used together with mut_fit.
e	plating efficiency; a positive number not bigger than 1.
cv	coefficient of variation of the number of cells in culture; a non-negative number usually smaller than 1.
trim	a non-negative integer; if specified, any mutant cell count bigger than this number will be replaced with this number.
ret	return parameter; either "list" (default) or "vector".

Details

- Final culture size is taken either as a constant or as a random variable drawn from the gamma distribution.
- Growth (and death if applicable) of the non-mutant cells is assumed to be deterministic and exponential. Time of culture growth is calculated per tube using starting and final number of cells as well as non-mutant death rate. Average number of mutations, which is proportional to the number of cellular divisions, is calculated using average mutation rate, growth rate, death rate, and time of culture growth.
- The actual number of mutations in the test tube is drawn from Poisson distribution using average number of mutations from the previous step.
- Moments of mutation (expressed in terms of number of individual cell divisions in that moment as a fraction of total number of cell divisions at the end of culture growth) are drawn from uniform distribution and then mutation epochs are calculated. For each mutant clone, the length of phenotypic lag is drawn from Poisson distribution with mean being the average length of phenotypic lag (supplied as a number of generations). If a particular mutation epoch exceeds total time of culture growth minus the extent of phenotypic lag, the whole mutant lineage is discarded.
- The size of the mutant clone is drawn from the distribution of the number of cells in a simple birth-and-death process using (8.46) in Bailey 1964.
- If plating is not perfect, the number of mutant colonies on the plates is drawn from binomial distribution.

Value

either a list of length two, each containing a vector of length n: "$mc" containing colony counts in parallel cultures while "$nt" contains the final number of cells in each culture; or a vector identical to "$mc".

References

Bailey NTJ. The Elements of Stochastic Processes with Applications to the Natural Sciences. 1st ed. John Wiley & Sons Inc; 1964.

Ycart B, Veziris N. Unbiased estimation of mutation rates under fluctuating final counts. PLoS One. 2014;9. doi:10.1371/journal.pone.0101434

Zheng Q. A second look at the final number of cells in a fluctuation experiment. J Theor Biol. 2016;401: 54–63. doi:10.1016/j.jtbi.2016.04.027

Mazoyer A, Drouilhet R, Despréaux S, Ycart B. Flan: An R package for inference on mutation models. R J. 2017;9: 334–351. doi:10.32614/rj-2017-029

Zheng Q. rSalvador: An R package for the fluctuation experiment. G3 Genes, Genomes, Genet. 2017;7: 3849–3856. doi:10.1534/g3.117.300120

Examples

```
rluria()
rluria(n=100, rate=1e-7, lag = 2, trim=5000, ret="v")
rluria(n=50, rate=1e-9, cv=0.5)
```

`sample.size` *Determine sample size to achieve prescribed power of the likelihood ratio test*

Description

This function calculates the sample size n1=n2 to achieve prescribed power in the likelihood ratio test given prescribed mutation rates and final culture sizes.

Usage

```
sample.size(
  power = 0.8,
  rate1 = 1e-09,
  rate2 = 2e-09,
  Nt1 = 1e+09,
  Nt2 = 1e+09,
  e1 = 1,
  e2 = 1,
  w1 = 1,
  w2 = 1,
  lag1 = 0,
  lag2 = 0,
  death1 = 0,
  death2 = 0,
  phi1 = 0,
  phi2 = 0,
  cv1 = 0,
  cv2 = 0,
  poisson1 = 0,
  poisson2 = 0,
  verbose = FALSE
)
```

Arguments

power	A vector containing desired statistical power. If longer than one, sample size will be calculated for each value of power.
rate1	Mutation rate of the first strain.
rate2	Mutation rate of the second strain.
Nt1	Average culture size of the first strain.
Nt2	Average culture size of the second strain.
e1	Plating efficiency of the first strain.
e2	Plating efficiency of the second strain.
w1	Relative mutant fitness of the first strain.
w2	Relative mutant fitness of the second strain.
lag1	Phenotypic lag of the first strain.
lag2	Phenotypic lag of the second strain.
death1	Relative death rate of the first strain.
death2	Relative death rate of the second strain.
phi1	Size of inoculum in proportion to the total culture size of the first strain.
phi2	Size of inoculum in proportion to the total culture size of the second strain.
cv1	Coefficient of variation of the culture sizes of the first strain.
cv2	Coefficient of variation of the culture sizes of the second strain.
poisson1	Average number of residual Poisson-distributed mutations on the plate of the first strain.
poisson2	Average number of residual Poisson-distributed mutations on the plate of the second strain.
verbose	if TRUE, mlemur will print messages to the console.

Value

A vector of length length(power) containing sample sizes (n1=n2).

References

Gudicha DW., Schmittmann VD. and Vermunt JK. Statistical power of likelihood ratio and Wald tests in latent class models with covariates. Behav. Res. Methods 2017;49: 1824–1837. doi:10.3758/s13428-016-0825-y

Self SG., Mauritsen RH. and Ohara J.Power Calculations for Likelihood Ratio Tests in Generalized Linear Models. Biometrics 1992;48: 31. doi:10.2307/2532736

Examples

```
sample.size(power = 0.8, rate1 = 1e-9, rate2 = 2e-9, Nt1 = 1e9, Nt2 = 5e8)
sample.size(power = c(0.2, 0.5, 0.9), rate1 = 1e-9, rate2 = 5e-9, Nt1 = 1e9,
Nt2 = 3e8, e1 = 1, e2 = 0.1)
```

Index