Package ‘phylosamp’

May 24, 2023

Type Package

Title Sample Size Calculations for Molecular and Phylogenetic Studies

Version 1.0.1

Date 2023-04-25

Maintainer Justin Lessler <`jlessler@unc.edu`>

Description
Implements novel tools for estimating sample sizes needed for phylogenetic studies, including studies focused on estimating the probability of true pathogen transmission between two cases given phylogenetic linkage and studies focused on tracking pathogen variants at a population level. Methods described in Wohl, Giles, and Lessler (2021) and in Wohl, Lee, DiPrete, and Lessler (2023).

License GPL-2

URL `https://github.com/HopkinsIDD/phylosamp`

BugReports `https://github.com/HopkinsIDD/phylosamp/issues`

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Depends R (>= 2.10), stats

Imports cli, lifecycle, rlang

Suggests cowplot, ggplot2, knitr, purrr, RColorBrewer, reshape2, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Shirlee Wohl [aut, ctb],
Elizabeth C Lee [aut, ctb],
Lucy D'Agostino McGowan [aut, ctb]
(<`https://orcid.org/0000-0002-6983-2759`>),
John R Giles [aut, ctb],
Justin Lessler [aut, cre]

Repository CRAN
Date/Publication 2023-05-23 23:12:10 UTC

R topics documented:

`exp_links` *Calculate expected number of links in a sample*

Description

[Deprecated] This function calculates the expected number of observed pairs in the sample that are linked by the linkage criteria. The function requires the sensitivity η and specificity χ of the linkage criteria, and sample size M. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
exp_links(eta, chi, rho, M, R = NULL, assumption = "mtml")
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`chi`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen (default=NULL)
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are:

1. `'stsl'` for the single-transmission single-linkage assumption (`prob_trans_stsl()`).
2. `'mtsl'` for the multiple-transmission single-linkage assumption (`prob_trans_mtsl()`).
3. `'mtml'` for the multiple-transmission multiple-linkage assumption (`prob_trans_mtml()`).

Value

scalar or vector giving the expected number of observed links in the sample

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other obs_pairs: `obs_pairs_mtml()`, `obs_pairs_mtsl()`, `obs_pairs_stsl()`

Examples

```
# The simplest case: single-transmission, single-linkage, and perfect sensitivity
exp_links(eta=1, chi=0.9, rho=0.5, M=100, assumption='stsl')

# Multiple-transmission and imperfect sensitivity
exp_links(eta=0.99, chi=0.9, rho=1, M=50, R=1, assumption='mtsl')

# Small outbreak, larger sampling proportion
exp_links(eta=0.99, chi=0.95, rho=1, M=50, R=1, assumption='mtml')

# Large outbreak, small sampling proportion
exp_links(eta=0.99, chi=0.95, rho=0.05, M=1000, R=1, assumption='mtml')
```

`falsediscoveryrate` *Calculate false discovery rate of a sample*

Description

[Deprecated] This function calculates the false discovery rate (proportion of linked pairs that are false positives) in a sample given the sensitivity η and specificity χ of the linkage criteria, and sample size M. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
falsediscoveryrate(eta, chi, rho, M, R = NULL, assumption = "mtml")
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`chi`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen (default=NULL)
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption (`prob_trans_stsl()`). 2. `'mtsl'` for the multiple-transmission single-linkage assumption (`prob_trans_mtsl()`). 3. `'mtml'` for the multiple-transmission multiple-linkage assumption (`prob_trans_mtml()`).

Value

scalar or vector giving the true discovery rate

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other discovery_rate: `truediscoveryrate`()

Examples

```
# The simplest case: single-transmission, single-linkage, and perfect sensitivity
falsediscoveryrate(eta=1, chi=0.9, rho=0.5, M=100, assumption='stsl')

# Multiple-transmission and imperfect sensitivity
falsediscoveryrate(eta=0.99, chi=0.9, rho=1, M=50, R=1, assumption='mtsl')

# Small outbreak, larger sampling proportion
falsediscoveryrate(eta=0.99, chi=0.95, rho=1, M=50, R=1, assumption='mtml')

# Large outbreak, small sampling proportion
falsediscoveryrate(eta=0.99, chi=0.95, rho=0.5, M=1000, R=1, assumption='mtml')
```

`genDistSim` *Simulations of the genetic distance distribution*

Description

This data object contains the genetic distance distributions for 168 values of R between 1.3 and 18. The distributions represent the the average of 1000 simulations for each value, which can be used as a reasonable proxy for the generation distribution for large outbreaks.

Usage

```
genDistSim
```

Format

dataframe

Author(s)

Shirlee Wohl, John Giles, and Justin Lessler

Examples

```
data(genDistSim)
```

`gendist_distribution` *Calculate genetic distance distribution*

Description

Function calculates the distribution of genetic distances in a population of viruses with the given parameters

Usage

```
gendist_distribution(
  mut_rate,
  mean_gens_pdf,
  max_link_gens = 1,
  max_gens = NULL,
  max_dist = NULL
)
```

Arguments

`mut_rate`	mean number of mutations per generation, assumed to be Poisson distributed
`mean_gens_pdf`	the density distribution of the mean number of generations between cases; the index of this vector is assumed to be the discrete distance between cases
`max_link_gens`	the maximum generations of separation for linked pairs
`max_gens`	the maximum number of generations to consider, if `NULL` (default) value is set to the highest number of generations in mean_gens_pdf with a non-zero probability
`max_dist`	the maximum distance to calculate, if `NULL` (default) value is set to max_gens * 99.9th percentile of mut_rate Poisson distribution

Value

a data frame with distances and probabilities

Author(s)

Shirlee Wohl and Justin Lessler

See Also

Other genetic distance functions: `gendist_roc_format()`, `gendist_sensspec_cutoff()`

Examples

```
# ebola-like pathogen
R <- 1.5
mut_rate <- 1

# use simulated generation distributions from the provided 'genDistSim' data object
data('genDistSim')
mean_gens_pdf <- as.numeric(genDistSim[genDistSim$R == R, -(1:2)])

# get theoretical genetic distance dist based on mutation rate and generation parameters
gendist_distribution(mut_rate = mut_rate,
                     mean_gens_pdf = mean_gens_pdf,
                     max_link_gens = 1)
```

`gendist_roc_format` *Make ROC curve from sensitivity and specificity*

Description

This is a wrapper function that takes output from the `gendist_sensspec_cutoff()` function and constructs values for the Receiver Operating Characteristic (ROC) curve

Usage

```
gendist_roc_format(
  cutoff,
  mut_rate,
  mean_gens_pdf,
  max_link_gens = 1,
  max_gens = NULL,
  max_dist = NULL
)
```

Arguments

`cutoff`	the maximum genetic distance at which to consider cases linked
`mut_rate`	mean number of mutations per generation, assumed to be Poisson distributed
`mean_gens_pdf`	the density distribution of the mean number of generations between cases; the index of this vector is assumed to be the discrete distance between cases
`max_link_gens`	the maximum generations of separation for linked pairs
`max_gens`	the maximum number of generations to consider, if `NULL` (default) value set to the highest number of generations in mean_gens_pdf with a non-zero probability
`max_dist`	the maximum distance to calculate, if `NULL` (default) value set to max_gens * 99.9th percentile of mut_rate Poisson distribution

Value

data frame with cutoff, sensitivity, and 1-specificity

Author(s)

Shirlee Wohl and Justin Lessler

See Also

Other genetic distance functions: `gendist_distribution()`, `gendist_sensspec_cutoff()`

Other ROC functions: `optim_roc_threshold()`

Examples

```
# ebola-like pathogen
R <- 1.5
mut_rate <- 1

# use simulated generation distributions
data('genDistSim')
mean_gens_pdf <- as.numeric(genDistSim[genDistSim$R == R, -(1:2)])

# get theoretical genetic distance dist based on mutation rate and generation parameters
dists <- as.data.frame(gendist_distribution(mut_rate = mut_rate,
                       mean_gens_pdf = mean_gens_pdf,
```

```
                     max_link_gens = 1))

dists <- reshape2::melt(dists,
                        id.vars = 'dist',
                        variable.name = 'status',
                        value.name = 'prob')

# get sensitivity and specificity using the same paramters
roc_calc <- gendist_roc_format(cutoff = 1:(max(dists$dist)-1),
                               mut_rate = mut_rate,
                               mean_gens_pdf = mean_gens_pdf)
```

---

gendist_sensspec_cutoff

Calculate sensitivity and specificity of a genetic distance cutoff

---

Description

Function to calculate the sensitivity and specificity of a genetic distance cutoff given an underlying mutation rate and mean number of generations between cases

Usage

```
gendist_sensspec_cutoff(
  cutoff,
  mut_rate,
  mean_gens_pdf,
  max_link_gens = 1,
  max_gens = NULL,
  max_dist = NULL
)
```

Arguments

`cutoff`	the maximum genetic distance at which to consider cases linked
`mut_rate`	mean number of mutations per generation, assumed to be Poisson distributed
`mean_gens_pdf`	the density distribution of the mean number of generations between cases; the index of this vector is assumed to be the discrete distance between cases
`max_link_gens`	the maximum generations of separation for linked pairs
`max_gens`	the maximum number of generations to consider, if `NULL` (default) value set to the highest number of generations in mean_gens_pdf with a non-zero probability
`max_dist`	the maximum distance to calculate, if `NULL` (default) value set to max_gens * 99.9th percentile of mut_rate Poisson distribution

Value

a data frame with the sensitivity and specificity for a particular genetic distance cutoff

Author(s)

Shirlee Wohl and Justin Lessler

See Also

Other genetic distance functions: `gendist_distribution()`, `gendist_roc_format()`

Examples

```
# calculate the sensitivity and specificity for a specific genetic distance threshold of 2 mutations
gendist_sensspec_cutoff(cutoff=2,
                        mut_rate=1,
                        mean_gens_pdf=c(0.02,0.08,0.15,0.75),
                        max_link_gens=1)

# calculate the sensitivity and specificity for a a range of genetic distance thresholds
gendist_sensspec_cutoff(cutoff=1:10,
                        mut_rate=1,
                        mean_gens_pdf=c(0.02,0.08,0.15,0.75),
                        max_link_gens=1)
```

`gen_dists` *Calculate genetic distance distribution*

Description

[Deprecated] Function calculates the distribution of genetic distances in a population of viruses with the given parameters

Usage

```
gen_dists(
  mut_rate,
  mean_gens_pdf,
  max_link_gens = 1,
  max_gens = NULL,
  max_dist = NULL
)
```

Arguments

`mut_rate`	mean number of mutations per generation, assumed to be Poisson distributed
`mean_gens_pdf`	the density distribution of the mean number of generations between cases; the index of this vector is assumed to be the discrete distance between cases
`max_link_gens`	the maximum generations of separation for linked pairs

`max_gens`	the maximum number of generations to consider, if `NULL` (default) value is set to the highest number of generations in mean_gens_pdf with a non-zero probability
`max_dist`	the maximum distance to calculate, if `NULL` (default) value is set to max_gens * 99.9th percentile of mut_rate Poisson distribution

Value

a data frame with distances and probabilities

Author(s)

Shirlee Wohl and Justin Lessler

See Also

Other mutrate_functions: `get_optim_roc()`, `sens_spec_calc()`, `sens_spec_roc()`

Examples

```
# ebola-like pathogen
R <- 1.5
mut_rate <- 1

# use simulated generation distributions from the provided 'genDistSim' data object
data('genDistSim')
mean_gens_pdf <- as.numeric(genDistSim[genDistSim$R == R, -(1:2)])

# get theoretical genetic distance dist based on mutation rate and generation parameters
gen_dists(mut_rate = mut_rate,
          mean_gens_pdf = mean_gens_pdf,
          max_link_gens = 1)
```

`get_optim_roc` *Find optimal ROC threshold*

Description

[Deprecated] This function takes the dataframe output of the `sens_spec_roc()` function and finds the optimal threshold of sensitivity and specificity by minimizing the distance to the top left corner of the Receiver Operating Characteristic (ROC) curve

Usage

```
get_optim_roc(roc)
```

Arguments

`roc`	a dataframe produced by the `sens_spec_roc()` function containing the Receiver Operating Characteristic (ROC) curve

Value

vector containing optimal thresholds of sensitivity and specificity

Author(s)

Shirlee Wohl, John Giles, and Justin Lessler

See Also

Other mutrate_functions: `gen_dists()`, `sens_spec_calc()`, `sens_spec_roc()`

Examples

```
# ebola-like pathogen
R <- 1.5
mut_rate <- 1

# use simulated generation distributions
data(genDistSim)
mean_gens_pdf <- as.numeric(genDistSim[genDistSim$R == R, -(1:2)])

# get theoretical genetic distance dist based on mutation rate and generation parameters
dists <- as.data.frame(gen_dists(mut_rate = mut_rate,
                                 mean_gens_pdf = mean_gens_pdf,
                                 max_link_gens = 1))

# reshape dataframe for plotting
dists <- reshape2::melt(dists,
                        id.vars = 'dist',
                        variable.name = 'status',
                        value.name = 'prob')

# get sensitivity and specificity using the same paramters
roc_calc <- sens_spec_roc(cutoff = 1:(max(dists$dist)-1),
                          mut_rate = mut_rate,
                          mean_gens_pdf = mean_gens_pdf)

# get the optimal value for the ROC plot
optim_point <- get_optim_roc(roc_calc)
```

`obs_pairs_mtml` *Expected number of observed pairs assuming multiple-transmission and multiple-linkage*

Description

[Deprecated] This function calculates the expected number of pairs observed in a sample of size `M`. The multiple-transmission and multiple-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to multiple cases j in the sampled population (M).
3. Linkage events are independent of one another (i.e, linkage of case i to case j has no bearing on linkage of case i to any other sample).

Usage

```
obs_pairs_mtml(chi, eta, rho, M, R)
```

Arguments

`chi`	scalar or vector giving the specificity of the linkage criteria
`eta`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the expected number of linked pairs observed in the sample

Author(s)

John Giles, Shirlee Wohl and Justin Lessler

See Also

Other obs_pairs: `exp_links()`, `obs_pairs_mtsl()`, `obs_pairs_stsl()`

Examples

```
# Perfect sensitivity and specificity
obs_pairs_mtml(eta=1, chi=1, rho=0.5, M=100, R=1)

obs_pairs_mtml(eta=0.99, chi=0.9, rho=1, M=50, R=1)

obs_pairs_mtml(eta=0.99, chi=0.9, rho=0.5, M=100, R=1)
```

`obs_pairs_mtsl` *Expected number of observed pairs assuming multiple-transmission and single-linkage*

Description

[Deprecated] This function calculates the expected number of pairs observed in a sample of size `M`. The multiple-transmission and single-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
obs_pairs_mtsl(chi, eta, rho, M, R)
```

Arguments

`chi`	scalar or vector giving the specificity of the linkage criteria
`eta`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the expected number of linked pairs observed in the sample

Author(s)

John Giles, Shirlee Wohl and Justin Lessler

See Also

Other obs_pairs: `exp_links()`, `obs_pairs_mtml()`, `obs_pairs_stsl()`

Examples

```
# Perfect sensitivity and specificity
obs_pairs_mtsl(eta=1, chi=1, rho=0.5, M=100, R=1)

obs_pairs_mtsl(eta=0.99, chi=0.9, rho=1, M=50, R=1)

obs_pairs_mtsl(eta=0.99, chi=0.9, rho=0.5, M=100, R=1)
```

`obs_pairs_stsl` *Expected number of observed pairs assuming single-transmission and single-linkage*

Description

[Deprecated] This function calculates the expected number of link pairs observed in a sample of size `M`. The single-transmission and single-linkage method assumes the following:

1. Each case i is linked by transmission to only one other case j in the population (N).
2. Each case i is linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
obs_pairs_stsl(eta, chi, rho, M)
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`chi`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled

Value

scalar or vector giving the expected number of linked pairs observed in the sample

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other obs_pairs: `exp_links()`, `obs_pairs_mtml()`, `obs_pairs_mtsl()`

Examples

```
# perfect sensitivity and specificity
obs_pairs_stsl(eta=1, chi=1, rho=0.5, M=100)

obs_pairs_stsl(eta=0.99, chi=0.9, rho=1, M=50)

obs_pairs_stsl(eta=0.99, chi=0.9, rho=0.5, M=100)
```

`optim_roc_threshold` *Find optimal ROC threshold*

Description

This function takes the dataframe output of the `gendist_roc_format()` function and finds the optimal threshold of sensitivity and specificity by minimizing the distance to the top left corner of the Receiver Operating Characteristic (ROC) curve

Usage

```
optim_roc_threshold(roc)
```

Arguments

`roc`	a dataframe produced by the `gendist_roc_format()` function containing the Receiver Operating Characteristic (ROC) curve

Value

vector containing optimal thresholds of sensitivity and specificity

Author(s)

Shirlee Wohl, John Giles, and Justin Lessler

See Also

Other ROC functions: `gendist_roc_format()`

Examples

```
# ebola-like pathogen
R <- 1.5
mut_rate <- 1

# use simulated generation distributions
data("genDistSim")
mean_gens_pdf <- as.numeric(genDistSim[genDistSim$R == R, -(1:2)])

# get theoretical genetic distance dist based on mutation rate and generation parameters
dists <- as.data.frame(gendist_distribution(mut_rate = mut_rate,
                       mean_gens_pdf = mean_gens_pdf,
                       max_link_gens = 1))

# reshape dataframe for plotting
dists <- reshape2::melt(dists,
                        id.vars = "dist",
                        variable.name = "status",
```

```
                       value.name = "prob")

# get sensitivity and specificity using the same paramters
roc_calc <- gendist_roc_format(cutoff = 1:(max(dists$dist)-1),
                         mut_rate = mut_rate,
                         mean_gens_pdf = mean_gens_pdf)

# get the optimal value for the ROC plot
optim_point <- optim_roc_threshold(roc_calc)
```

`prob_trans_mtml` *Probability of transmission assuming multiple-transmission and multiple-linkage*

Description

[Deprecated] This function calculates the probability that two cases are linked by direct transmission given that they have been linked by phylogenetic criteria. The multiple-transmission and multiple-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to multiple cases j in the sampled population (M).
3. Linkage events are independent of one another (i.e, linkage of case i to case j has no bearing on linkage of case i to any other sample).

Usage

```
prob_trans_mtml(eta, chi, rho, M, R)
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`chi`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the probability of transmission between two cases given linkage by phylogenetic criteria

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other prob_trans: `prob_trans_mtsl()`, `prob_trans_stsl()`

Examples

```
# Perfect sensitivity and specificity
prob_trans_mtml(eta=1, chi=1, rho=0.5, M=100, R=1)

prob_trans_mtml(eta=0.99, chi=0.9, rho=1, M=50, R=1)

prob_trans_mtml(eta=0.99, chi=0.9, rho=0.5, M=100, R=1)
```

`prob_trans_mtsl` *Probability of transmission assuming multiple-transmission and single-linkage*

Description

[Deprecated] This function calculates the probability that two cases are linked by direct transmission given that they have been linked by phylogenetic criteria. The multiple-transmission and single-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
prob_trans_mtsl(chi, eta, rho, M, R)
```

Arguments

`chi`	scalar or vector giving the specificity of the linkage criteria
`eta`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the probability of transmission between two cases given linkage by phylogenetic criteria

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other prob_trans: `prob_trans_mtml()`, `prob_trans_stsl()`

Examples

```
# Perfect sensitivity and specificity
prob_trans_mtsl(eta=1, chi=1, rho=0.5, M=100, R=1)

prob_trans_mtsl(eta=0.99, chi=0.9, rho=1, M=50, R=1)

prob_trans_mtsl(eta=0.99, chi=0.9, rho=0.5, M=100, R=1)
```

`prob_trans_stsl` *Probability of transmission assuming single-transmission and single-linkage*

Description

[Deprecated] This function calculates the probability that two cases are linked by direct transmission given that they have been linked by phylogenetic criteria. The single-transmission and single-linkage method assumes the following:

1. Each case i is linked by transmission to only one other case j in the population (N).
2. Each case i is linked by the linkage criteria to only one other case j in the sampled population (M).

For perfect sensitivity, set `eta = 1`.

Usage

```
prob_trans_stsl(eta, chi, rho, M)
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`chi`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled

Value

scalar or vector giving the probability of transmission between two cases given linkage by phylogenetic criteria

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other prob_trans: `prob_trans_mtml()`, `prob_trans_mtsl()`

Examples

```
# perfect sensitivity and specificity
prob_trans_stsl(eta=1, chi=1, rho=0.2, M=100)

# perfect sensitivity only
prob_trans_stsl(eta=1, chi=0.95, rho=0.2, M=100)

prob_trans_stsl(eta=0.99, chi=0.95, rho=0.9, M=50)

prob_trans_stsl(eta=0.99, chi=0.95, rho=0.05, M=100)
```

`relR_power` *Calculate power for detecting differential transmission given a sample size*

Description

Function to calculate the power given a sample size. This is the top level function to be called to calculate power given a sample size m and a proportion sampled.

Usage

```
relR_power(
  m,
  R_a,
  R_b,
  p_a,
  N = NULL,
  rho = NULL,
  alpha = 0.05,
  alternative = c("two_sided", "less", "greater"),
  sensitivity = 1,
  specificity = 1,
  overdispersion = NULL
)
```

Arguments

`m`	the sample size.
`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.

`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.
`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`rho`	Numeric. The proportion of the infected pool sampled. Only one of `rho` or `N` should be specified. Values should be between 0 and 1.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`sensitivity`	Numeric. The sensitivity of the linkage criteria. Must be between 0 and 1. Default: 1.
`specificity`	Numeric. The specificity of the linkage criteria. Must be between 0 and 1. Default: 1.
`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.

Value

The power given `m`

`relR_power_simulated` *Simulate power for detecting differential transmission*

Description

Simulate power for detecting differential transmission

Usage

```
relR_power_simulated(
  m,
  R_a,
  R_b,
  p_a,
  N,
  alpha = 0.05,
  alternative = c("two_sided", "less", "greater"),
  sensitivity = 1,
  specificity = 1,
  overdispersion = NULL,
  nsims = 1e+05
)
```

Arguments

`m`	the sample size.
`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.
`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.
`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`sensitivity`	Numeric. The sensitivity of the linkage criteria. Must be between 0 and 1. Default: 1.
`specificity`	Numeric. The specificity of the linkage criteria. Must be between 0 and 1. Default: 1.
`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.
`nsims`	Numeric. The number of simulations. Default: 100000

Value

Simulated power

`relR_samplesize` *Calculate sample size needed to detect differential transmission*

Description

Function for calculating sample size given a set of assumptions. This is the high level wrapper function that users should call directly.

Usage

```
relR_samplesize(
  R_a,
  R_b,
  p_a,
  N,
  alpha = 0.05,
```

```
  alternative = c("two_sided", "less", "greater"),
  power = 0.8,
  sensitivity = 1,
  specificity = 1,
  overdispersion = NULL,
  correct_for_imbalance = FALSE
)
```

Arguments

`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.
`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.
`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`power`	Numeric. The desired power. Must be a value between 0 and 1. Default: 0.8.
`sensitivity`	Numeric. The sensitivity of the linkage criteria. Must be between 0 and 1. Default: 1.
`specificity`	Numeric. The specificity of the linkage criteria. Must be between 0 and 1. Default: 1.
`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.
`correct_for_imbalance`	Logical. Should we use simulation to correct for being over/under powered due to large differences in group sizes? Default: `FALSE`.

Value

Sample size needed achieve desired type I and II error rates under assumptions. Will return `NA` and throw a warning if impossible.

Examples

```
## Calculate sample size needed to detect a difference between groups where
## group A has a reproductive value of 2, group B has a reproductive
## value of 2.5, the groups are balanced, and the total outbreak size is
## 1,000
```

```
relR_samplesize(R_a = 2,
                R_b = 2.5,
                p_a = 0.5,
                N = 1000)

## Update the above calculation to account for imperfect sensitivity = 0.7
relR_samplesize(R_a = 2,
                R_b = 2.5,
                p_a = 0.5,
                N = 1000,
                sensitivity = 0.7)

## Update the above calculation to allow for overdispersion
relR_samplesize(R_a = 2,
                R_b = 2.5,
                p_a = 0.5,
                N = 1000,
                sensitivity = 0.7,
                overdispersion = 2000)
```

`relR_samplesize_basic` *Calculate simple derived sample size for detecting differential transmission*

Description

Function that does the simple derived sample size calculation with no corrections. I.e., directly applies the math as if sensitivity and specificity are perfect.

Usage

```
relR_samplesize_basic(
  R_a,
  R_b,
  p_a,
  N,
  alpha = 0.05,
  alternative = c("two_sided", "less", "greater"),
  power = 0.8,
  overdispersion = NULL,
  allow_impossible_m = FALSE
)
```

Arguments

`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.

`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.
`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`power`	Numeric. The desired power. Must be a value between 0 and 1. Default: 0.8.
`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.
`allow_impossible_m`	Logical. Indicates whether a value for `m` can be returned that is greater than the input `N`. Default: `FALSE`.

Value

The required sample size. `NA` if larger than `N`.

`relR_samplesize_ci` *Calculate sample size for detecting differential transmission with uncertainty bounds*

Description

This function assumes you want to correct for imbalance, if not there is a closed form solution for the estimated sample size that does not include uncertainty bounds. (see `relR_samplesize`).

Usage

```
relR_samplesize_ci(
  R_a,
  R_b,
  p_a,
  N,
  alpha = 0.05,
  alternative = c("two_sided", "less", "greater"),
  power = 0.8,
  sensitivity = 1,
  specificity = 1,
  overdispersion = NULL,
  nsims = 1000,
```

```
  uncertainty_percent = 0.95,
  B = 1000
)
```

Arguments

`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.
`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.
`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`power`	Numeric. The desired power. Must be a value between 0 and 1. Default: 0.8.
`sensitivity`	Numeric. The sensitivity of the linkage criteria. Must be between 0 and 1. Default: 1.
`specificity`	Numeric. The specificity of the linkage criteria. Must be between 0 and 1. Default: 1.
`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.
`nsims`	The number of inner simulations run per estimate. Default: 10000
`uncertainty_percent`	The percent of the uncertainty interval. Default: .95
`B`	The number of outer simulations run to estimate the uncertainty. Default: 1000

Value

A vector with three quantities:

- sample size: Sample size needed achieve desired type I and II error rates under assumptions. Will return NA and throw a warning if impossible.
- lower bound: The lower bound of an uncertainty interval
- upper bound: The upper bound of an uncertainty interval

---

relR_samplesize_linkerr

Calculate sample size for detecting differential transmission correcting for sensitivity and specificity

---

Description

Function to run the sample size calculation correcting for imperfect sensitivity and specificity, but not doing any simulation based corrections.

Usage

```
relR_samplesize_linkerr(
  R_a,
  R_b,
  p_a,
  N,
  alpha = 0.05,
  alternative = c("two_sided", "less", "greater"),
  power = 0.8,
  sensitivity = 1,
  specificity = 1,
  overdispersion = NULL,
  allow_impossible_m = FALSE
)
```

Arguments

`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.
`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.
`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`power`	Numeric. The desired power. Must be a value between 0 and 1. Default: 0.8.
`sensitivity`	Numeric. The sensitivity of the linkage criteria. Must be between 0 and 1. Default: 1.
`specificity`	Numeric. The specificity of the linkage criteria. Must be between 0 and 1. Default: 1.

`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.
`allow_impossible_m`	Logical. Indicates whether a value for `m` can be returned that is greater than the input `N`. Default: `FALSE`.

Value

Sample size needed achieve desired type I and II error rates under assumptions. Will return NA and throw a warning if impossible.

`relR_samplesize_opterr` *Function to calculate the error in estimated sample size for use in optimize function*

Description

Function to calculate the error in estimated sample size for use in optimize function

Usage

```
relR_samplesize_opterr(
  m,
  R_a,
  R_b,
  p_a,
  N,
  alpha,
  alternative,
  power,
  sensitivity,
  specificity,
  overdispersion
)
```

Arguments

`m`	the sample size.
`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.
`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.

`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`power`	Numeric. The desired power. Must be a value between 0 and 1. Default: 0.8.
`sensitivity`	Numeric. The sensitivity of the linkage criteria. Must be between 0 and 1. Default: 1.
`specificity`	Numeric. The specificity of the linkage criteria. Must be between 0 and 1. Default: 1.
`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.

Value

Squared error between the input sample size and estimated sample size

---

`relR_samplesize_simsolve`

Calculate optimized sample size for detecting differential transmission

Description

Function to calculate optimized sample size by solving the transcendental equation that occurs when you replace the R values with ones that account for sensitivity and specificity.

Usage

```
relR_samplesize_simsolve(
  R_a,
  R_b,
  p_a,
  N,
  alpha = 0.05,
  alternative = c("two_sided", "less", "greater"),
  power = 0.8,
  sensitivity = 1,
  specificity = 1,
  overdispersion = NULL,
  epsilon = 0.01,
  nsims = 1e+05,
  tolerance = 10
)
```

Arguments

`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.
`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.
`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`power`	Numeric. The desired power. Must be a value between 0 and 1. Default: 0.8.
`sensitivity`	Numeric. The sensitivity of the linkage criteria. Must be between 0 and 1. Default: 1.
`specificity`	Numeric. The specificity of the linkage criteria. Must be between 0 and 1. Default: 1.
`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.
`epsilon`	Numeric. Dictates the minimum value for `R_b = R_a + epsilon` attempted in the simulation. Default: 0.01.
`nsims`	Dictates the number of simulations for each power simulation. Default: 100000
`tolerance`	Dictates the tolerance for the binary search. Default: 10.

Value

Simulated sample size needed achieve desired type I and II error rates under assumptions. Will return NA and throw a warning if impossible.

`relR_samplesize_solve` *Calculate optimal sample size for detecting differential transmission with imperfect specificity*

Description

Function to solve for optimal sample size when the specificity isn't 1

Usage

```
relR_samplesize_solve(
  R_a,
  R_b,
  p_a,
  N,
  alpha = 0.05,
  alternative = c("two_sided", "less", "greater"),
  power = 0.8,
  sensitivity = 1,
  specificity = 1,
  overdispersion = NULL,
  allow_impossible_m = FALSE
)
```

Arguments

`R_a`	Numeric (Positive). The assumed R among the group in the denominator of the ratio. Input value must be greater than 0.
`R_b`	Numeric (Positive). The assumed R among the group in the numerator of the ratio. Input value must be greater than 0.
`p_a`	Numeric. The proportion of the population in group `a`. Must be between 0 and 1.
`N`	Numeric (Positive). The size of the infected pool. Only one of `rho` or `N` should be specified.
`alpha`	Numeric. The desired alpha level. Default: 0.05
`alternative`	Character. Specifies the alternative hypothesis. Must be: `two_sided` (Default), `less`, or `greater`
`power`	Numeric. The desired power. Must be a value between 0 and 1. Default: 0.8.
`sensitivity`	Numeric. The sensitivity of the linkage criteria. Must be between 0 and 1. Default: 1.
`specificity`	Numeric. The specificity of the linkage criteria. Must be between 0 and 1. Default: 1.
`overdispersion`	Numeric (Positive). An overdispersion parameter, set if the assumed distribution of the number of edges is negative binomial. If `NULL` the assumed distribution is Poisson (equivalent to an overdispersion parameter of infinity) Default: `NULL` Note that this is equivalent to setting the overdispersion parameter to `Inf`.
`allow_impossible_m`	Logical. Indicates whether a value for `m` can be returned that is greater than the input `N`. Default: `FALSE`.

Value

The sample size

`samplesize` *Calculate sample size*

Description

[Deprecated] This function calculates the sample size needed to obtain at least a defined false discovery rate given a final outbreak size N.

Usage

```
samplesize(eta, chi, N, R = NULL, phi, min_pairs = 1, assumption = "mtml")
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`chi`	scalar or vector giving the specificity of the linkage criteria
`N`	scalar or vector giving the final outbreak size
`R`	scalar or vector giving the effective reproductive number of the pathogen
`phi`	scalar or vector giving the desired true discovery rate (1-false discovery rate)
`min_pairs`	minimum number of linked pairs observed in the sample, defaults to 1 pair (2 samples); this is to ensure reasonable results are obtained
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption (`prob_trans_stsl()`). 2. `'mtsl'` for the multiple-transmission single-linkage assumption (`prob_trans_mtsl()`). 3. `'mtml'` for the multiple-transmission multiple-linkage assumption (`prob_trans_mtml()`).

Value

scalar or vector giving the sample size needed to meet the given conditions

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

Examples

```
samplesize(eta=0.99, chi=0.995, N=100, R=1, phi=0.75)
```

`sens_spec_calc` *Calculate sensitivity and specificity*

Description

[Deprecated] Function to calculate the sensitivity and specificity of a genetic distance cutoff given an underlying mutation rate and mean number of generations between cases

Usage

```
sens_spec_calc(
  cutoff,
  mut_rate,
  mean_gens_pdf,
  max_link_gens = 1,
  max_gens = NULL,
  max_dist = NULL
)
```

Arguments

`cutoff`	the maximum genetic distance at which to consider cases linked
`mut_rate`	mean number of mutations per generation, assumed to be Poisson distributed
`mean_gens_pdf`	the density distribution of the mean number of generations between cases; the index of this vector is assumed to be the discrete distance between cases
`max_link_gens`	the maximum generations of separation for linked pairs
`max_gens`	the maximum number of generations to consider, if `NULL` (default) value set to the highest number of generations in mean_gens_pdf with a non-zero probability
`max_dist`	the maximum distance to calculate, if `NULL` (default) value set to max_gens * 99.9th percentile of mut_rate Poisson distribution

Value

a data frame with the sensitivity and specificity for a particular genetic distance cutoff

Author(s)

Shirlee Wohl and Justin Lessler

See Also

Other mutrate_functions: `gen_dists()`, `get_optim_roc()`, `sens_spec_roc()`

Examples

```
# calculate the sensitivity and specificity for a specific genetic distance threshold of 2 mutations
sens_spec_calc(cutoff=2,
              mut_rate=1,
              mean_gens_pdf=c(0.02,0.08,0.15,0.75),
              max_link_gens=1)

# calculate the sensitivity and specificity for a a range of genetic distance thresholds
sens_spec_calc(cutoff=1:10,
              mut_rate=1,
              mean_gens_pdf=c(0.02,0.08,0.15,0.75),
              max_link_gens=1)
```

`sens_spec_roc` *Make ROC from sensitivity and specificity*

Description

[Deprecated] This is a wrapper function that takes output from the `sens_spec_calc()` function and constructs values for the Receiver Operating Characteristic (ROC) curve

Usage

```
sens_spec_roc(
  cutoff,
  mut_rate,
  mean_gens_pdf,
  max_link_gens = 1,
  max_gens = NULL,
  max_dist = NULL
)
```

Arguments

`cutoff`	the maximum genetic distance at which to consider cases linked
`mut_rate`	mean number of mutations per generation, assumed to be Poisson distributed
`mean_gens_pdf`	the density distribution of the mean number of generations between cases; the index of this vector is assumed to be the discrete distance between cases
`max_link_gens`	the maximum generations of separation for linked pairs
`max_gens`	the maximum number of generations to consider, if `NULL` (default) value set to the highest number of generations in mean_gens_pdf with a non-zero probability
`max_dist`	the maximum distance to calculate, if `NULL` (default) value set to max_gens * 99.9th percentile of mut_rate Poisson distribution

Value

data frame with cutoff, sensitivity, and 1-specificity

Author(s)

Shirlee Wohl and Justin Lessler

See Also

Other mutrate_functions: `gen_dists()`, `get_optim_roc()`, `sens_spec_calc()`

Examples

```
# ebola-like pathogen
R <- 1.5
mut_rate <- 1

# use simulated generation distributions
data('genDistSim')
mean_gens_pdf <- as.numeric(genDistSim[genDistSim$R == R, -(1:2)])

# get theoretical genetic distance dist based on mutation rate and generation parameters
dists <- as.data.frame(gen_dists(mut_rate = mut_rate,
                                 mean_gens_pdf = mean_gens_pdf,
                                 max_link_gens = 1))

dists <- reshape2::melt(dists,
                        id.vars = 'dist',
                        variable.name = 'status',
                        value.name = 'prob')

# get sensitivity and specificity using the same paramters
roc_calc <- sens_spec_roc(cutoff = 1:(max(dists$dist)-1),
                          mut_rate = mut_rate,
                          mean_gens_pdf = mean_gens_pdf)
```

`translink_expected_links_obs` *Calculate expected number of transmission links in a sample*

Description

This function calculates the expected number of observed pairs in the sample that are linked by the linkage criteria. The function requires the sensitivity and specificity of the linkage criteria, and sample size M. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
translink_expected_links_obs(
  sensitivity,
  specificity,
  rho,
  M,
  R = NULL,
  assumption = "mtml"
)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`specificity`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen (default=NULL)
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption. 2. `'mtsl'` for the multiple-transmission single-linkage assumption. 3. `'mtml'` for the multiple-transmission multiple-linkage assumption.

Value

scalar or vector giving the expected number of observed links in the sample

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl` `translink_expected_links_obs_stsl()`, `translink_expected_links_true_mtml()`, `translink_expected_links_t` `translink_expected_links_true_stsl()`, `translink_expected_links_true()`, `translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`, `translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
# The simplest case: single-transmission, single-linkage, and perfect sensitivity
translink_expected_links_obs(sensitivity=1, specificity=0.9, rho=0.5, M=100, assumption='stsl')

# Multiple-transmission and imperfect sensitivity
translink_expected_links_obs(sensitivity=0.99, specificity=0.9, rho=1, M=50, R=1, assumption='mtsl')
```

```
# Small outbreak, larger sampling proportion
translink_expected_links_obs(sensitivity=0.99, specificity=0.95, rho=1, M=50,
R=1, assumption='mtml')

# Large outbreak, small sampling proportion
translink_expected_links_obs(sensitivity=0.99, specificity=0.95,
rho=0.05, M=1000, R=1, assumption='mtml')
```

`translink_expected_links_obs_mtml` *Calculate expected number of observed pairs assuming multiple-transmission and multiple-linkage*

Description

This function calculates the expected number of pairs observed in a sample of size `M`. The multiple-transmission and multiple-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to multiple cases j in the sampled population (M).
3. Linkage events are independent of one another (i.e, linkage of case i to case j has no bearing on linkage of case i to any other sample).

Usage

```
translink_expected_links_obs_mtml(specificity, sensitivity, rho, M, R)
```

Arguments

`specificity`	scalar or vector giving the specificity of the linkage criteria
`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the expected number of linked pairs observed in the sample

Author(s)

John Giles, Shirlee Wohl and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtsl()`, `translink_expected_links_obs_stsl`
`translink_expected_links_obs()`, `translink_expected_links_true_mtml()`, `translink_expected_links_true_m`
`translink_expected_links_true_stsl()`, `translink_expected_links_true()`, `translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`, `translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
# Perfect sensitivity and specificity
translink_expected_links_obs_mtml(sensitivity=1, specificity=1, rho=0.5, M=100, R=1)

translink_expected_links_obs_mtml(sensitivity=0.99, specificity=0.9, rho=1, M=50, R=1)

translink_expected_links_obs_mtml(sensitivity=0.99, specificity=0.9, rho=0.5, M=100, R=1)
```

`translink_expected_links_obs_mtsl`

Calculate expected number of observed pairs assuming multiple-transmission and single-linkage

Description

This function calculates the expected number of pairs observed in a sample of size `M`. The multiple-transmission and single-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
translink_expected_links_obs_mtsl(specificity, sensitivity, rho, M, R)
```

Arguments

`specificity`	scalar or vector giving the specificity of the linkage criteria
`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the expected number of linked pairs observed in the sample

Author(s)

John Giles, Shirlee Wohl and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_stsl`
`translink_expected_links_obs()`, `translink_expected_links_true_mtml()`, `translink_expected_links_true_m`
`translink_expected_links_true_stsl()`, `translink_expected_links_true()`, `translink_fdr()`,
`translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`,
`translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
# Perfect sensitivity and specificity
translink_expected_links_obs_mtsl(sensitivity=1, specificity=1, rho=0.5, M=100, R=1)

translink_expected_links_obs_mtsl(sensitivity=0.99, specificity=0.9, rho=1, M=50, R=1)

translink_expected_links_obs_mtsl(sensitivity=0.99, specificity=0.9, rho=0.5, M=100, R=1)
```

---

translink_expected_links_obs_stsl

Calculate expected number of observed pairs assuming single-transmission and single-linkage

Description

This function calculates the expected number of link pairs observed in a sample of size `M`. The single-transmission and single-linkage method assumes the following:

1. Each case i is linked by transmission to only one other case j in the population (N).
2. Each case i is linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
translink_expected_links_obs_stsl(sensitivity, specificity, rho, M)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`specificity`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled

Value

scalar or vector giving the expected number of linked pairs observed in the sample

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl`
`translink_expected_links_obs()`, `translink_expected_links_true_mtml()`, `translink_expected_links_true_m`
`translink_expected_links_true_stsl()`, `translink_expected_links_true()`, `translink_fdr()`,
`translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`,
`translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
# perfect sensitivity and specificity
translink_expected_links_obs_stsl(sensitivity=1, specificity=1, rho=0.5, M=100)

translink_expected_links_obs_stsl(sensitivity=0.99, specificity=0.9, rho=1, M=50)

translink_expected_links_obs_stsl(sensitivity=0.99, specificity=0.9, rho=0.5, M=100)
```

---

translink_expected_links_true

Calculate expected number of true transmission pairs

Description

This function calculates the expected number true transmission pairs in a sample of size `M`. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
translink_expected_links_true(
  sensitivity,
  rho,
  M,
  R = NULL,
  assumption = "mtml"
)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen (default=NULL)
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption. 2. `'mtsl'` for the multiple-transmission single-linkage assumption. 3. `'mtml'` for the multiple-transmission multiple-linkage assumption.

Value

scalar or vector giving the expected number of true transmission pairs in the sample

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl` `translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mt` `translink_expected_links_true_mtsl()`, `translink_expected_links_true_stsl()`, `translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`, `translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
translink_expected_links_true(sensitivity=0.99, rho=0.75, M=100, R=1)
```

---

translink_expected_links_true_mtml

Calculate expected number of true transmission pairs assuming multiple-transmission and multiple-linkage

Description

This function calculates the expected number of true transmission pairs in a sample of size `M`. The multiple-transmission and multiple-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to multiple cases j in the sampled population (M).
3. Linkage events are independent of one another (i.e, linkage of case i to case j has no bearing on linkage of case i to any other sample).

Usage

```
translink_expected_links_true_mtml(sensitivity, rho, M, R)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the expected number of true transmission pairs in the sample

Author(s)

John Giles, Shirlee Wohl and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl`
`translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mt`
`translink_expected_links_true_stsl()`, `translink_expected_links_true()`, `translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`, `translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
translink_expected_links_true_mtml(sensitivity=0.95, rho=0.2, M=1000, R=1)
```

`translink_expected_links_true_mtsl`

Calculate expected number of true transmission pairs assuming multiple-transmission and single-linkage

Description

This function calculates the expected number true transmission pairs in a sample of size `M`. The multiple-transmission and single-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
translink_expected_links_true_mtsl(sensitivity, rho, M, R)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the expected number of true transmission pairs in the sample

Author(s)

John Giles, Shirlee Wohl and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl` `translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mt` `translink_expected_links_true_stsl()`, `translink_expected_links_true()`, `translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`, `translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
translink_expected_links_true_mtsl(sensitivity=0.95, rho=0.2, M=200, R=1)
```

---

translink_expected_links_true_stsl

Calculate expected number of true transmission pairs assuming single-transmission and single-linkage

Description

This function calculates the expected number of true transmission pairs in a sample of size `M`. The single-transmission and single-linkage method assumes the following:

1. Each case i is linked by transmission to only one other case j in the population (N).
2. Each case i is linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
translink_expected_links_true_stsl(sensitivity, rho, M)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled

Value

scalar or vector giving the expected number of true transmission pairs in the sample

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl`
`translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mtr`
`translink_expected_links_true_mtsl()`, `translink_expected_links_true()`, `translink_fdr()`,
`translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`,
`translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
translink_expected_links_true_stsl(sensitivity=0.95, rho=0.2, M=200)
```

`translink_fdr` *Calculate false discovery rate of identifying transmission pairs in a sample*

Description

This function calculates the false discovery rate (proportion of linked pairs that are false positives) in a sample given the sensitivity and specificity of the linkage criteria, and sample size M. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
translink_fdr(sensitivity, specificity, rho, M, R = NULL, assumption = "mtml")
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`specificity`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen (default=NULL)
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption. 2. `'mtsl'` for the multiple-transmission single-linkage assumption. 3. `'mtml'` for the multiple-transmission multiple-linkage assumption.

Value

scalar or vector giving the true discovery rate

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl`
`translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mt`
`translink_expected_links_true_mtsl()`, `translink_expected_links_true_stsl()`, `translink_expected_links_`
`translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`,
`translink_prob_transmit()`, `translink_samplesize()`, `translink_tdr()`

Examples

```
# The simplest case: single-transmission, single-linkage, and perfect sensitivity
translink_fdr(sensitivity=1, specificity=0.9, rho=0.5, M=100, assumption='stsl')

# Multiple-transmission and imperfect sensitivity
translink_fdr(sensitivity=0.99, specificity=0.9, rho=1, M=50, R=1, assumption='mtsl')

# Small outbreak, larger sampling proportion
translink_fdr(sensitivity=0.99, specificity=0.95, rho=1, M=50, R=1, assumption='mtml')

# Large outbreak, small sampling proportion
translink_fdr(sensitivity=0.99, specificity=0.95, rho=0.5, M=1000, R=1, assumption='mtml')
```

`translink_prob_transmit` *Calculate probability of transmission*

Description

This function calculates the probability that two cases are linked by direct transmission given that they have been linked by phylogenetic criteria. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
translink_prob_transmit(
  sensitivity,
  specificity,
  rho,
  M,
  R,
  assumption = "mtml"
)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`specificity`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption. 2. `'mtsl'` for the multiple-transmission single-linkage assumption. 3. `'mtml'` for the multiple-transmission multiple-linkage assumption.

Value

scalar or vector giving the probability of transmission between two cases given linkage by phylogenetic criteria

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl`
`translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mtr`
`translink_expected_links_true_mtsl()`, `translink_expected_links_true_stsl()`, `translink_expected_links_`
`translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transm`
`translink_samplesize()`, `translink_tdr()`

Examples

```
translink_prob_transmit(sensitivity=0.99, specificity=0.9, rho=0.5, M=100, R=1)
```

`translink_prob_transmit_mtml`

Calculate probability of transmission assuming multiple-transmission and multiple-linkage

Description

This function calculates the probability that two cases are linked by direct transmission given that they have been linked by phylogenetic criteria. The multiple-transmission and multiple-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to multiple cases j in the sampled population (M).
3. Linkage events are independent of one another (i.e, linkage of case i to case j has no bearing on linkage of case i to any other sample).

Usage

```
translink_prob_transmit_mtml(sensitivity, specificity, rho, M, R)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`specificity`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the probability of transmission between two cases given linkage by phylogenetic criteria

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl`
`translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mt`
`translink_expected_links_true_mtsl()`, `translink_expected_links_true_stsl()`, `translink_expected_links_`
`translink_fdr()`, `translink_prob_transmit_mtsl()`, `translink_prob_transmit_stsl()`, `translink_prob_transmi`
`translink_samplesize()`, `translink_tdr()`

Examples

```
# Perfect sensitivity and specificity
translink_prob_transmit_mtml(sensitivity=1, specificity=1, rho=0.5, M=100, R=1)

translink_prob_transmit_mtml(sensitivity=0.99, specificity=0.9, rho=1, M=50, R=1)

translink_prob_transmit_mtml(sensitivity=0.99, specificity=0.9, rho=0.5, M=100, R=1)
```

`translink_prob_transmit_mtsl` *Calculate probability of transmission assuming multiple-transmission and single-linkage*

Description

This function calculates the probability that two cases are linked by direct transmission given that they have been linked by phylogenetic criteria. The multiple-transmission and single-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
translink_prob_transmit_mtsl(specificity, sensitivity, rho, M, R)
```

Arguments

`specificity`	scalar or vector giving the specificity of the linkage criteria
`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the probability of transmission between two cases given linkage by phylogenetic criteria

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml`(), `translink_expected_links_obs_mtsl` `translink_expected_links_obs_stsl`(), `translink_expected_links_obs`(), `translink_expected_links_true_mt` `translink_expected_links_true_mtsl`(), `translink_expected_links_true_stsl`(), `translink_expected_links_` `translink_fdr`(), `translink_prob_transmit_mtml`(), `translink_prob_transmit_stsl`(), `translink_prob_transm` `translink_samplesize`(), `translink_tdr`()

Examples

```
# Perfect sensitivity and specificity
translink_prob_transmit_mtsl(sensitivity=1, specificity=1, rho=0.5, M=100, R=1)

translink_prob_transmit_mtsl(sensitivity=0.99, specificity=0.9, rho=1, M=50, R=1)

translink_prob_transmit_mtsl(sensitivity=0.99, specificity=0.9, rho=0.5, M=100, R=1)
```

`translink_prob_transmit_stsl`

Calculate probability of transmission assuming single-transmission and single-linkage

Description

This function calculates the probability that two cases are linked by direct transmission given that they have been linked by phylogenetic criteria. The single-transmission and single-linkage method assumes the following:

1. Each case i is linked by transmission to only one other case j in the population (N).
2. Each case i is linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
translink_prob_transmit_stsl(sensitivity, specificity, rho, M)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`specificity`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled

Details

For perfect sensitivity, set `sensitivity = 1`.

Value

scalar or vector giving the probability of transmission between two cases given linkage by phylogenetic criteria

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl`
`translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mt`
`translink_expected_links_true_mtsl()`, `translink_expected_links_true_stsl()`, `translink_expected_links_`
`translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transm`
`translink_samplesize()`, `translink_tdr()`

Examples

```
# perfect sensitivity and specificity
translink_prob_transmit_stsl(sensitivity=1, specificity=1, rho=0.2, M=100)

# perfect sensitivity only
translink_prob_transmit_stsl(sensitivity=1, specificity=0.95, rho=0.2, M=100)

translink_prob_transmit_stsl(sensitivity=0.99, specificity=0.95, rho=0.9, M=50)

translink_prob_transmit_stsl(sensitivity=0.99, specificity=0.95, rho=0.05, M=100)
```

`translink_samplesize` *Calculate sample size needed to identify true transmission links*

Description

This function calculates the sample size needed to identify transmission links at a predefined false discovery rate, given a final outbreak size N.

Usage

```
translink_samplesize(
  sensitivity,
  specificity,
  N,
  R = NULL,
  tdr,
  min_pairs = 1,
  assumption = "mtml"
)
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`specificity`	scalar or vector giving the specificity of the linkage criteria
`N`	scalar or vector giving the final outbreak size
`R`	scalar or vector giving the effective reproductive number of the pathogen
`tdr`	scalar or vector giving the desired true discovery rate (1-false discovery rate)
`min_pairs`	minimum number of linked pairs observed in the sample, defaults to 1 pair (2 samples); this is to ensure reasonable results are obtained
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption. 2. `'mtsl'` for the multiple-transmission single-linkage assumption. 3. `'mtml'` for the multiple-transmission multiple-linkage assumption.

Value

scalar or vector giving the sample size needed to meet the given conditions

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl`
`translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mtm`
`translink_expected_links_true_mtsl()`, `translink_expected_links_true_stsl()`, `translink_expected_links_`
`translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transm`
`translink_prob_transmit()`, `translink_tdr()`

Examples

```
translink_samplesize(sensitivity=0.99, specificity=0.995, N=100, R=1, tdr=0.75)
```

`translink_tdr` *Calculate true discovery rate of identifying transmission pairs*

Description

This function calculates the true discovery rate (proportion of true transmission pairs) in a sample given the sensitivity and specificity of the linkage criteria, and sample size M. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
translink_tdr(sensitivity, specificity, rho, M, R = NULL, assumption = "mtml")
```

Arguments

`sensitivity`	scalar or vector giving the sensitivity of the linkage criteria
`specificity`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen (default=NULL)
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption. 2. `'mtsl'` for the multiple-transmission single-linkage assumption. 3. `'mtml'` for the multiple-transmission multiple-linkage assumption.

Value

scalar or vector giving the true discovery rate

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other transmission linkage functions: `translink_expected_links_obs_mtml()`, `translink_expected_links_obs_mtsl` `translink_expected_links_obs_stsl()`, `translink_expected_links_obs()`, `translink_expected_links_true_mt` `translink_expected_links_true_mtsl()`, `translink_expected_links_true_stsl()`, `translink_expected_links_` `translink_fdr()`, `translink_prob_transmit_mtml()`, `translink_prob_transmit_mtsl()`, `translink_prob_transm` `translink_prob_transmit()`, `translink_samplesize()`

Examples

```
# The simplest case: single-transmission, single-linkage, and perfect sensitivity
translink_tdr(sensitivity=1, specificity=0.9, rho=0.5, M=100, assumption='stsl')

# Multiple-transmission and imperfect sensitivity
translink_tdr(sensitivity=0.99, specificity=0.9, rho=1, M=50, R=1, assumption='mtsl')

# Small outbreak, larger sampling proportion
translink_tdr(sensitivity=0.99, specificity=0.95, rho=1, M=50, R=1, assumption='mtml')

# Large outbreak, small sampling proportion
translink_tdr(sensitivity=0.99, specificity=0.95, rho=0.5, M=1000, R=1, assumption='mtml')
```

`truediscoveryrate` *Calculate true discovery rate of a sample*

Description

[Deprecated] This function calculates the true discovery rate (proportion of true transmission pairs) in a sample given the sensitivity η and specificity χ of the linkage criteria, and sample size M. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
truediscoveryrate(eta, chi, rho, M, R = NULL, assumption = "mtml")
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`chi`	scalar or vector giving the specificity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen (default=NULL)
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are: 1. `'stsl'` for the single-transmission single-linkage assumption (`prob_trans_stsl()`). 2. `'mtsl'` for the multiple-transmission single-linkage assumption (`prob_trans_mtsl()`). 3. `'mtml'` for the multiple-transmission multiple-linkage assumption (`prob_trans_mtml()`).

Value

scalar or vector giving the true discovery rate

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other discovery_rate: `falsediscoveryrate()`

Examples

```
# The simplest case: single-transmission, single-linkage, and perfect sensitivity
truediscoveryrate(eta=1, chi=0.9, rho=0.5, M=100, assumption='stsl')

# Multiple-transmission and imperfect sensitivity
truediscoveryrate(eta=0.99, chi=0.9, rho=1, M=50, R=1, assumption='mtsl')

# Small outbreak, larger sampling proportion
truediscoveryrate(eta=0.99, chi=0.95, rho=1, M=50, R=1, assumption='mtml')

# Large outbreak, small sampling proportion
truediscoveryrate(eta=0.99, chi=0.95, rho=0.5, M=1000, R=1, assumption='mtml')
```

`true_pairs` *Calculate expected number of true transmission pairs*

Description

[Deprecated] This function calculates the expected number true transmission pairs in a sample of size `M`. Assumptions about transmission and linkage (single or multiple) can be specified.

Usage

```
true_pairs(eta, rho, M, R = NULL, assumption = "mtml")
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen (default=NULL)
`assumption`	a character vector indicating which assumptions about transmission and linkage criteria. Default = `'mtml'`. Accepted arguments are:

1. `'stsl'` for the single-transmission single-linkage assumption (`prob_trans_stsl()`).
2. `'mtsl'` for the multiple-transmission single-linkage assumption (`prob_trans_mtsl()`).
3. `'mtml'` for the multiple-transmission multiple-linkage assumption (`prob_trans_mtml()`).

Value

scalar or vector giving the expected number of true transmission pairs in the sample

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other true_pairs: `true_pairs_mtml()`, `true_pairs_mtsl()`, `true_pairs_stsl()`

Examples

```
true_pairs(eta=0.99, rho=0.75, M=100, R=1)
```

`true_pairs_mtml` *Expected number of true transmission pairs assuming multiple-transmission and multiple-linkage*

Description

[Deprecated] This function calculates the expected number of true transmission pairs in a sample of size `M`. The multiple-transmission and multiple-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to multiple cases j in the sampled population (M).
3. Linkage events are independent of one another (i.e, linkage of case i to case j has no bearing on linkage of case i to any other sample).

Usage

```
true_pairs_mtml(eta, rho, M, R)
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the expected number of true transmission pairs in the sample

Author(s)

John Giles, Shirlee Wohl and Justin Lessler

See Also

Other true_pairs: `true_pairs_mtsl()`, `true_pairs_stsl()`, `true_pairs()`

Examples

```
true_pairs_mtml(eta=0.95, rho=0.2, M=1000, R=1)
```

`true_pairs_mtsl` *Expected number of true transmission pairs assuming multiple-transmission and single-linkage*

Description

[Deprecated] This function calculates the expected number true transmission pairs in a sample of size `M`. The multiple-transmission and single-linkage method assumes the following:

1. Each case i is, on average, the infector of `R` cases in the population (N)
2. Each case i is allowed to be linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
true_pairs_mtsl(eta, rho, M, R)
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled
`R`	scalar or vector giving the effective reproductive number of the pathogen

Value

scalar or vector giving the expected number of true transmission pairs in the sample

Author(s)

John Giles, Shirlee Wohl and Justin Lessler

See Also

Other true_pairs: `true_pairs_mtml()`, `true_pairs_stsl()`, `true_pairs()`

Examples

```
true_pairs_mtsl(eta=0.95, rho=0.2, M=200, R=1)
```

`true_pairs_stsl` *Expected number of true transmission pairs assuming single-transmission and single-linkage*

Description

[Deprecated] This function calculates the expected number of true transmission pairs in a sample of size `M`. The single-transmission and single-linkage method assumes the following:

1. Each case i is linked by transmission to only one other case j in the population (N).
2. Each case i is linked by the linkage criteria to only one other case j in the sampled population (M).

Usage

```
true_pairs_stsl(eta, rho, M)
```

Arguments

`eta`	scalar or vector giving the sensitivity of the linkage criteria
`rho`	scalar or vector giving the proportion of the final outbreak size that is sampled
`M`	scalar or vector giving the number of cases sampled

Value

scalar or vector giving the expected number of true transmission pairs in the sample

Author(s)

John Giles, Shirlee Wohl, and Justin Lessler

See Also

Other true_pairs: `true_pairs_mtml()`, `true_pairs_mtsl()`, `true_pairs()`

Examples

```
true_pairs_stsl(eta=0.95, rho=0.2, M=200)
```

`varfreq_cdf_logistic` *Calculate cumulative observed variant prevalence at time t given logistic growth*

Description

This function calculates the cumulative observed variant prevalence after t time steps (e.g., days) given a logistic growth rate and initial variant prevalence.

Usage

```
varfreq_cdf_logistic(t, p0_v1, r_v1, c_ratio = 1)
```

Arguments

`t`	time step number (e.g., days) at which to calculate prevalence
`p0_v1`	initial variant prevalence (# introductions / infected population size)
`r_v1`	logistic growth rate
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)

Value

scalar giving the cdf of variant prevalence at time t

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other logistic growth functions: `varfreq_freq_logistic()`

Other variant frequency functions: `varfreq_expected_mbias()`, `varfreq_freq_logistic()`, `varfreq_obs_freq()`

Examples

```
varfreq_cdf_logistic(t = 30, p0_v1 = 1/10000, r_v1 = 0.1, c_ratio = 1)
```

`varfreq_expected_mbias` *Calculate multiplicative bias (observed / actual) in variant prevalence*

Description

This function calculates the multiplicative bias of the observed variant proportion relative to the actual variant proportion. This function assumes that variant 1 is the variant of concern. This function is specific to the two-variant system.

Usage

```
varfreq_expected_mbias(p_v1, c_ratio)
```

Arguments

`p_v1`	actual variant prevalence (proportion)
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2

Value

scalar giving the multiplicative bias of variant 1

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant frequency functions: `varfreq_cdf_logistic()`, `varfreq_freq_logistic()`, `varfreq_obs_freq()`

Examples

```
varfreq_expected_mbias(p_v1 = 0.1, c_ratio = 1.1)
```

`varfreq_freq_logistic` *Calculate observed variant prevalence at time t given logistic growth*

Description

This function calculates the observed variant prevalence after t time steps (e.g., days) given a logistic growth rate and initial variant prevalence.

Usage

```
varfreq_freq_logistic(t, p0_v1, r_v1, c_ratio = 1)
```

Arguments

`t`	time step number (e.g., days) at which to calculate prevalence
`p0_v1`	initial variant prevalence (# introductions / infected population size)
`r_v1`	logistic growth rate
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2; default = 1 (no bias)

Value

scalar giving the variant prevalence at time t

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other logistic growth functions: `varfreq_cdf_logistic()`

Other variant frequency functions: `varfreq_cdf_logistic()`, `varfreq_expected_mbias()`, `varfreq_obs_freq()`

Examples

```
varfreq_freq_logistic(t = 30, p0_v1 = 1/10000, r_v1 = 0.1, c_ratio = 1)
```

`varfreq_obs_freq` *Calculate observed variant prevalence*

Description

This function calculates the observed variant prevalence from the coefficient of detection ratio and the actual variant prevalence. This function assumes that variant 1 is the variant of concern. This function is specific to the two-variant system.

Usage

```
varfreq_obs_freq(p_v1, c_ratio)
```

Arguments

`p_v1`	actual variant prevalence (proportion)
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2

Value

scalar of observed prevalence of variant 1

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant frequency functions: `varfreq_cdf_logistic()`, `varfreq_expected_mbias()`, `varfreq_freq_logistic()`

Examples

```
varfreq_obs_freq(p_v1 = 0.1, c_ratio = 1.1)
```

`vartrack_cod_ratio` *Calculate the coefficient of detection ratio for two variants*

Description

This function calculates the coefficient of detection ratio C_{V_1}/C_{V_2} for two variants. This function assumes that variant 1 is the variant of concern. This function is specific to the two-variant system. Parameters not provided are assumed to be equivalent between the two variants.

Usage

```
vartrack_cod_ratio(
  phi_v1 = 1,
  phi_v2 = 1,
  gamma_v1 = 1,
  gamma_v2 = 1,
  psi_v1 = 1,
  psi_v2 = 1,
  tau_a = 1,
  tau_s = 1
)
```

Arguments

`phi_v1`	probability that a tested infection caused by variant 1 results in a positive test (sensitivity)
`phi_v2`	probability that a tested infection caused by variant 2 results in a positive test (sensitivity)
`gamma_v1`	probability that a detected infection caused by variant 1 meets some quality threshold
`gamma_v2`	probability that a detected infection caused by variant 2 meets some quality threshold
`psi_v1`	probability that an infection caused by variant 1 is asymptomatic
`psi_v2`	probability that an infection caused by variant 2 is asymptomatic
`tau_a`	probability of testing an asymptomatic infection (any variant); note that this parameter is not required if psi_v1==psi_v2
`tau_s`	probability of testing a symptomatic infection (any variant); note that this parameter is not required if psi_v1==psi_v2

Value

scalar giving the multiplicative bias of variant 1

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant tracking functions: `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xsect()`, `vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_co` `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
vartrack_cod_ratio(phi_v1=0.975, phi_v2=0.95, gamma_v1=0.8, gamma_v2=0.6)
```

`vartrack_prob_detect` *Calculate the probability of detecting a variant given a sample size*

Description

This function calculates the probability of detecting the presence of a variant given a sample size and sampling strategy.

Usage

```
vartrack_prob_detect(
  n,
  t = NA,
  p_v1 = NA,
  omega,
  p0_v1 = NA,
  r_v1 = NA,
  c_ratio = 1,
  sampling_freq
)
```

Arguments

`n`	sample size (either of cross-section or per timestep)
`t`	time step number (e.g., days) at which variant should be detected by. Default = NA (either `'t'` or `'p_v1'` should be provided, not both)
`p_v1`	the desired prevalence to detect a variant by. Default = NA (either `'t'` or `'p_v1'` should be provided, not both)
`omega`	probability of sequencing (or other characterization) success
`p0_v1`	initial variant prevalence (# introductions / infected population size)
`r_v1`	logistic growth rate
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)
`sampling_freq`	the sampling frequency (must be either 'xsect' or 'cont')

Value

scalar of detection probability

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant detection functions: `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xsect()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xse` `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
# Cross-sectional sampling
vartrack_prob_detect(p_v1 = 0.02, n = 100, omega = 0.8, c_ratio = 1, sampling_freq = 'xsect')

# Periodic sampling
vartrack_prob_detect(n = 158, t = 30, omega = 0.8, p0_v1 = 1/10000,
r_v1 = 0.1, c_ratio = 1, sampling_freq = 'cont')
```

`vartrack_prob_detect_cont`

Calculate probability of detecting a variant given a per-timestep sample size assuming periodic sampling

---

Description

This function calculates the probability of detecting the presence of a variant given a sample size and either a desired maximum time until detection or a desired prevalence by which to detect the variant by. It assumes a periodic sampling strategy, where samples are collected at regular intervals (time steps).

Usage

```
vartrack_prob_detect_cont(
  n,
  t = NA,
  p_v1 = NA,
  omega,
  p0_v1,
  r_v1,
  c_ratio = 1
)
```

Arguments

`n`	per-timestep (e.g., per day) sample size
`t`	time step number (e.g., days) at which variant should be detected by. Default = NA (either `'t'` or `'p_v1'` should be provided, not both)

`p_v1`	the desired prevalence to detect a variant by. Default = NA (either `'t'` or `'p_v1'` should be provided, not both)
`omega`	probability of sequencing (or other characterization) success
`p0_v1`	initial variant prevalence (# introductions / infected population size)
`r_v1`	logistic growth rate
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)

Value

scalar of detection probability

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant detection functions: `vartrack_prob_detect_xsect()`, `vartrack_prob_detect()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_xsect()`, `vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
vartrack_prob_detect_cont(n = 158, t = 30, omega = 0.8, p0_v1 = 1/10000, r_v1 = 0.1, c_ratio = 1)
```

`vartrack_prob_detect_xsect`

Calculate probability of detecting a variant assuming cross-sectional sampling

---

Description

This function calculates the probability of detecting the presence of a variant given a sample size and assuming a single, cross-sectional sample of detected infections.

Usage

```
vartrack_prob_detect_xsect(p_v1, n, omega, c_ratio = 1)
```

Arguments

`p_v1`	variant prevalence (proportion)
`n`	sample size
`omega`	probability of sequencing (or other characterization) success
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)

Value

scalar of expected sample size

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant detection functions: `vartrack_prob_detect_cont()`, `vartrack_prob_detect()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
vartrack_prob_detect_xsect(p_v1 = 0.02, n = 100, omega = 0.8, c_ratio = 1)
```

`vartrack_prob_prev` *Calculate confidence in a variant estimate given a sample size*

Description

This function calculates the probability of accurately estimating variant prevalence given a sample size and desired precision in the variant prevalence estimate. Currently, only cross-sectional sampling is supported.

Usage

```
vartrack_prob_prev(p_v1, n, omega, precision, c_ratio = 1, sampling_freq)
```

Arguments

`p_v1`	variant prevalence (proportion)
`n`	sample size
`omega`	probability of sequencing (or other characterization) success
`precision`	desired precision in variant prevalence estimate
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)
`sampling_freq`	the sampling frequency (must be either 'xsect' in current implementation)

Value

scalar of expected sample size

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant prevalence estimation functions: `vartrack_prob_prev_xsect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xse` `vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
vartrack_prob_prev(p_v1 = 0.1, n = 200, omega = 0.8, precision = 0.1,
c_ratio = 1, sampling_freq = 'xsect')
```

---

vartrack_prob_prev_xsect

Calculate confidence in a variant estimate assuming cross-sectional sampling

Description

This function calculates the probability of accurately estimating variant prevalence given a given a sample size and desired precision in the variant prevalence estimate, and assuming a single, cross-sectional sample of detected infections.

Usage

```
vartrack_prob_prev_xsect(p_v1, n, omega, precision, c_ratio = 1)
```

Arguments

`p_v1`	variant prevalence (proportion)
`n`	sample size
`omega`	probability of sequencing (or other characterization) success
`precision`	desired precision in variant prevalence estimate
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)

Value

scalar of expected sample size

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant prevalence estimation functions: `vartrack_prob_prev()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xse` `vartrack_prob_detect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_de` `vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
vartrack_prob_prev_xsect(p_v1 = 0.1, n = 200, precision = 0.1, omega = 0.8, c_ratio = 1)
```

---

vartrack_samplesize_detect

Calculate sample size needed for variant detection given a desired probability of detection

Description

This function calculates the sample size needed for detecting the presence of a variant given a desired probability of detection and sampling strategy.

Usage

```
vartrack_samplesize_detect(
  prob,
  t = NA,
  p_v1 = NA,
  omega,
  p0_v1 = NA,
  r_v1 = NA,
  c_ratio = 1,
  sampling_freq
)
```

Arguments

`prob`	desired probability of detection
`t`	time step number (e.g., days) at which variant should be detected by. Default = NA (either `'t'` or `'p_v1'` should be provided, not both)
`p_v1`	the desired prevalence to detect a variant by. Default = NA (either `'t'` or `'p_v1'` should be provided, not both)
`omega`	probability of sequencing (or other characterization) success
`p0_v1`	initial variant prevalence (# introductions / infected population size)
`r_v1`	logistic growth rate
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)
`sampling_freq`	the sampling frequency (must be either 'xsect' or 'cont')

Value

scalar of expected sample size

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant detection functions: `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xsect()`, `vartrack_prob_detect()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect_xsect()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xse`
`vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_co`
`vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
# Cross-sectional sampling
vartrack_samplesize_detect(p_v1 = 0.1, prob = 0.95, omega = 0.8,
                           c_ratio = 1, sampling_freq = 'xsect')

# Periodic sampling
vartrack_samplesize_detect(prob = 0.95, t = 30, omega = 0.8, p0_v1 = 1/10000,
                           r_v1 = 0.1, c_ratio = 1, sampling_freq = 'cont')
```

`vartrack_samplesize_detect_cont` *Calculate sample size needed for variant detection assuming periodic sampling*

Description

This function calculates the sample size needed for detecting the presence of a variant given a desired probability of detection and either a desired maximum time until detection or a desired prevalence by which to detect the variant by. It assumes a periodic sampling strategy, where samples are collected at regular intervals (time steps).

Usage

```
vartrack_samplesize_detect_cont(
  prob,
  t = NA,
  p_v1 = NA,
  omega,
  p0_v1,
  r_v1,
  c_ratio = 1
)
```

Arguments

`prob`	desired probability of detection
`t`	time step number (e.g., days) at which variant should be detected by. Default = NA (either `'t'` or `'p_v1'` should be provided, not both)
`p_v1`	the desired prevalence to detect a variant by. Default = NA (either `'t'` or `'p_v1'` should be provided, not both)
`omega`	probability of sequencing (or other characterization) success
`p0_v1`	initial variant prevalence (# introductions / infected population size)
`r_v1`	logistic growth rate
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)

Value

scalar of expected sample size

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant detection functions: `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xsect()`, `vartrack_prob_detect()`, `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xse`
`vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_xs`
`vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
vartrack_samplesize_detect_cont(prob = 0.95, t = 30, omega = 0.8,
p0_v1 = 1/10000, r_v1 = 0.1, c_ratio = 1)
```

`vartrack_samplesize_detect_xsect`

Calculate sample size needed for variant detection assuming cross-sectional sampling

---

Description

This function calculates the sample size needed for detecting the presence of a variant given a desired probability of detection and assuming a single, cross-sectional sample of detected infections.

Usage

```
vartrack_samplesize_detect_xsect(p_v1, prob, omega, c_ratio = 1)
```

Arguments

`p_v1`	variant prevalence (proportion)
`prob`	desired probability of detection
`omega`	probability of sequencing (or other characterization) success
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)

Value

scalar of expected sample size

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant detection functions: `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xsect()`, `vartrack_prob_detect()`, `vartrack_samplesize_detect_cont()`, `vartrack_samplesize_detect()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xse`
`vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_co`
`vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`, `vartrack_samplesize_prev()`

Examples

```
vartrack_samplesize_detect_xsect(p_v1 = 0.1, prob = 0.95, omega = 0.8, c_ratio = 1)
```

`vartrack_samplesize_prev` *Calculate sample size needed for estimating variant prevalence given a desired confidence*

Description

This function calculates the sample size needed for estimating variant prevalence given a desired confidence and desired precision in the variant prevalence estimate. Currently, only cross-sectional sampling is supported.

Usage

```
vartrack_samplesize_prev(
  p_v1,
  prob,
  precision,
  omega,
  c_ratio = 1,
  sampling_freq
)
```

Arguments

`p_v1`	variant prevalence (proportion)
`prob`	desired confidence in variant prevalence estimate
`precision`	desired precision in variant prevalence estimate
`omega`	probability of sequencing (or other characterization) success
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)
`sampling_freq`	the sampling frequency (must be ’xsect’ in current implementation)

Value

scalar of sample size

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant prevalence estimation functions: `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_prev_xsect()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xse` `vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_co` `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`, `vartrack_samplesize_prev_xsect()`

Examples

```
vartrack_samplesize_prev(p_v1 = 0.1, prob = 0.95, precision = 0.25,
omega = 0.8, c_ratio = 1, sampling_freq = 'xsect')
```

`vartrack_samplesize_prev_xsect` *Calculate sample size needed for variant prevalence estimation under cross-sectional sampling*

Description

This function calculates the sample size needed for estimating variant prevalence given a desired confidence and desired precision in the variant prevalence estimate and assuming a single, cross-sectional sample of detected infections.

Usage

```
vartrack_samplesize_prev_xsect(p_v1, prob, precision, omega, c_ratio = 1)
```

Arguments

`p_v1`	variant prevalence (proportion)
`prob`	desired confidence in variant prevalence estimate
`precision`	desired precision in variant prevalence estimate
`omega`	probability of sequencing (or other characterization) success
`c_ratio`	coefficient of detection ratio, calculated as the ratio of the coefficients of variant 1 to variant 2. Default = 1 (no bias)

Value

scalar of sample size

Author(s)

Shirlee Wohl, Elizabeth C. Lee, Bethany L. DiPrete, and Justin Lessler

See Also

Other variant prevalence estimation functions: `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_prev()`

Other variant tracking functions: `vartrack_cod_ratio()`, `vartrack_prob_detect_cont()`, `vartrack_prob_detect_xse` `vartrack_prob_detect()`, `vartrack_prob_prev_xsect()`, `vartrack_prob_prev()`, `vartrack_samplesize_detect_co` `vartrack_samplesize_detect_xsect()`, `vartrack_samplesize_detect()`, `vartrack_samplesize_prev()`

Examples

```
vartrack_samplesize_prev_xsect(p_v1 = 0.1, prob = 0.95, precision = 0.25, omega = 0.8, c_ratio = 1)
```

Index