Package 'anovir'

October 24, 2020

Type Package

Title Analysis of Virulence

Version 0.1.0

Maintainer Philip Agnew <philip.agnew@ird.fr>

Description Epidemiological population dynamics models traditionally define
a pathogen's virulence as the increase in the per capita rate of mortality
of infected hosts due to infection. This package provides functions
allowing virulence to be estimated by maximum likelihood techniques. The
approach is based on the analysis of relative survival comparing survival
in matching cohorts of infected vs. uninfected hosts (Agnew 2019)
<doi:10.1101/530709>.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Depends bbmle, R (>= 2.10), stats4

Suggests knitr, rmarkdown, stats, testthat

Collate 'anovir-package.R' 'private_functions.R' 'nll_recovery.R'
'nll_recovery_II.R' 'data_descriptions.R' 'main.R'
'play_file.R' 'nll_functions.R' 'nll_frailty_correlated.R'

VignetteBuilder knitr

URL https://www.biorxiv.org/content/10.1101/530709v1

NeedsCompilation no

Author Philip Agnew [aut, cre],
Jimmy Lopez [aut]

Repository CRAN

Date/Publication 2020-10-24 08:50:05 UTC

R topics documented:

---

anovir-package   *anovir: Analysis of Virulence*

Description

Epidemiological population dynamics models traditionally define a pathogen's virulence as the increase in the per capita rate of mortality of infected hosts due to infection. The ANOVIR package provides functions allowing virulence to be estimated by maximum likelihood techniques. The approach is based on the analysis of relative survival applied to the comparison of survival in matching cohorts of experimentally-infected vs. uninfected hosts (Agnew 2019).

Details

The analysis of relative survival is a statistical method for estimating excess mortality. Excess mortality occurs when a target population experiences greater mortality than would be expected for a given period of time. Here excess mortality is estimated in the context of emprical studies where survival in populations of experimentally infected hosts is compared to that in matching populations of uninfected, or control, hosts. In this context the relative survival approach assumes the rate of mortality observed in the infected treatment arises as the sum of two independent and mutually exclusive sources of mortality, (i) the 'natural' or background rate of mortality, and (ii) an addition rate of mortality due to infection.

The background rate of mortality is the expected rate of mortality hosts in the infected treatment would have experienced had they not been exposed to infection; it is estimated from mortality observed in the matching uninfected control treatment. When there is background mortality, the rate of mortality of infected hosts due to infection is not directly observed; however it can be estimated from the difference in mortality observed for an infected treatment and that observed in a matching control treatment.

The two sources of mortality assumed in the relative survival approach, and the additive effect of their rates for infected hosts, is also found in the population dynamics models on which most epidemiological theory is based. The additional rate of mortality of infected hosts due to infection is generally how these models define pathogen virulence.

Acknowledgements

PA is grateful to the following people;

- Yannis MICHALAKIS for affording me the time and space at work to develop this project

- Simon BLANFORD and Matthew THOMAS for generously providing the data from their 2012 study and allowing it to be made publically available

- Célia TOURAINE at the Institut du Cancer de Montpellier (ICM) for reading the original manuscript and validating the general approach of analysing relative survival as a means to estimate virulence

- Eric ELGUERO for useful input during discussions on models for recovery from infection and estimating confidence intervals

- François ROUSSET for helpful technical advice concerning R

This work was funded by basic research funds from the French research agencies of the Centre National de la Recherche Scientifique (CNRS) and the Institut de Recherche pour le Développement (IRD).

Author(s)

Philip AGNEW & Jimmy LOPEZ

References

Agnew P (2019) Estimating virulence from relative survival. bioRxiv: 530709 doi

See Also

Vignettes for examples of how to use and modify the functions in this package to estimate pathogen virulence

---

av_long_infected   *Average longevity: estimate for infected hosts*

Description

Calculates expected longevity of infected hosts due to background mortality and mortality due to infection

Usage

```
av_long_infected(a1, b1, a2, b2, d1 = "", d2 = "")
```

Arguments

a1, b1	numeric: location & scale parameters for background mortality, respectively
a2, b2	numeric: location & scale parameters for mortality due to infection, respectively
d1, d2	character: probability distributions to describe background mortality and mortality due to infection, respectively

Details

The expected average longevity is calculated as the integral from zero to infinity for the product of the cumulative survival functions for background mortality and mortality due to infection, given values of a1, b1, d1, a2, b2, d2

Value

a vector

See Also

av_long_uninfected

Examples

```
av_long_infected(
  a1 = 3.0, b1 = 0.6, d1 = "Weibull",
  a2 = 2.5, b2 = 0.5, d2 = "Frechet") # 12.156
```

---

av_long_uninfected | *Average longevity: estimate for uninfected hosts*

Description

Calculates expected average longevity due only to background mortality

Usage

```
av_long_uninfected(a1, b1, d1 = "")
```

Arguments

a1, b1   numeric: location & scale parameters for background mortality, respectively

d1   character: probability distribution chosen to describe data

Details

The expected average longevity is calculated as the integral from zero to infinity of the cumulative survival function for background mortality, given values of a1, b1, d1

Value

a vector

See Also

[av_long_infected](#)

Examples

```
av_long_uninfected(a1 = 3.0, b1 = 0.6, d1 = "Weibull") #17.947
```

---

check_data | *Checks data are correctly described for models*

Description

Function checking 'time', 'censor' and 'infected treatment' columns in data.frame are correctly specified for the likelihood functions in this package.

Usage

```
check_data(
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment
)
```

Arguments

data	Name of data.frame to be checked
time	Name of column with time of event data (death or censoring); requires time > 0 and numeric.
censor	Name of column containing censor status; '1' censored, '0' uncensored or died during experiment (needs to be numeric)
infected_treatment	
	Name of column for infection treatment status; '1' a treatment exposed to infection, '0' treatment not exposed to infection (needs to be numeric). NB Different likelihood models make different assumptions as to whether all the individuals in an infected treatment are infected or not (c.f. `nll_exposed_infected`)

Details

An error message is also given if data contain rows 'NA'; these need removing.

NB error messages triggered on 1st encounter with a fault. Correct and check revised data for other faults.

Value

Message, 'Checks completed' or error message

Examples

```
# view head of data.frame to be checked
  head(data_blanford, 3)

# specify data.frame and names of columns for;
  # time, censor status, infection status
  check_data(data = data_blanford,
             time = t,
             censor = censor,
             infected_treatment = inf)
```

```
# create data.frame 't_zero' with t = 0 in first row of data.frame
t_zero <- data_blanford
t_zero[1, 8] <- 0
head(t_zero, 3)

check_data(data = t_zero,
           time = t,
           censor = censor,
           infected_treatment = inf)

# correct '0' and make first row of column 'censor' = NA
  t_zero[1, 8] <- 1 ; t_zero[1, 5] <- NA ; head(t_zero, 3)

check_data(data = t_zero,
           time = t,
           censor = censor,
           infected_treatment = inf)

# remove row(s) with 'NA'
  t_zero_II <- na.omit(t_zero) # NB applies to whole data.frame
  check_data(data = t_zero_II,
           time = t,
           censor = censor,
           infected_treatment = inf)
```

---

conf_ints_virulence   *Approximate 95% confidence intervals for virulence*

Description

Function calculating the 95% confidence intervals for a hazard function based on the variance and covariance of its location and scale parameters.

Usage

```
conf_ints_virulence(
  a2 = a2,
  b2 = b2,
  var_a2 = var_a2,
  var_b2 = var_b2,
  cov_a2b2 = cov_a2b2,
  d2 = "",
  tmax = 21
)
```

Arguments

a2	numeric. Estimated value of location parameter describing mortality due to infection
b2	numeric. Estimated value of scale parameter describing mortality due to infection
var_a2	numeric. Estimated variance of location parameter describing mortality due to infection
var_b2	numeric. Estimated variance of scale parameter describing mortality due to infection
cov_a2b2	numeric. Estimated covariance of location and scale parameters above
d2	character. Probability distribution assumed to describe virulence; Weibull, Gumbel or Fréchet
tmax	maximum time virulence will be calculated for. Default value; tmax = 21

Details

The approach is based on the interval being estimated as a complementary log-log function of the hazard function, h(t), with the variance of virulence being estimated by the Delta method applied to log(h[t]).

Value

matrix containing estimates of virulence over time ± approx. 95% confidence intervals

Examples

```
# the values, variance and covariance of the location and scale parameters
# [a2,a2] describing mortality due to infection were estimated as;
# a2 = 2.5807642
# b2 = 0.1831328
# var_a2 = 0.0008196927
# var_b2 = 0.0010007282
# cov_a2b2 = -0.0003119921

 ci_matrix01 <- conf_ints_virulence(
   a2 = 2.5807642,
   b2 = 0.1831328,
   var_a2 = 0.0008196927,
   var_b2 = 0.0010007282,
   cov_a2b2 = -0.0003119921,
   d2 = "Weibull",
   tmax = 15)

 tail(ci_matrix01)

 plot(ci_matrix01[, 't'], ci_matrix01[, 'h2'],
   type = 'l', col = 'red',
   xlab = 'time', ylab = 'virulence (± 95% ci)')
```

```
lines(ci_matrix01[, 'lower_ci'], col = 'grey')
lines(ci_matrix01[, 'upper_ci'], col = 'grey')
```

---

data_blanford *Full data from Blanford et al (2012)*

Description

Complete data from the publication: Blanford S, Jenkins NE, Read AF, Thomas MB (2012) Evaluating the lethal and pre-lethal effects of a range of fungi against adult Anopheles stephensi mosquitoes. Malaria Journal. 11:365 doi

Usage

```
data_blanford
```

Format

An object of class data.frame with 1320 rows and 9 columns.

Details

block experimental block within experiment (1 - 5)

treatment experimental treatment

replicate cage replicate cage within treatment (1 - 4)

day time post-infection (days)

censor '1' censored, '0' died

d an indicator variable; '0' censored, '1' died

inf '0' uninfected treatement, '1' infected treatment

t time post-infection (days)

fq frequency of individuals

Source

Simon Blanford and Matthew Thomas generously provided and allowed the release of these data

Examples

```
head(data_blanford)
```

---

data_lorenz   *A subset of data from Lorenz & Koella (2011)*

Description

Data on the longevity of 256 adult female mosquitoes and the number of pathogen spores they harboured at the time of their death.

Usage

```
data_lorenz
```

Format

An object of class data.frame with 256 rows and 8 columns.

Details

These are the Lorenz & Koella data analysed in Agnew (2019) doi

Value

A dataframe

Infectious.dose Number of spores larvae were exposed to (spores/larva)

Food food treatment: '50' or '100'

Sex sex of mosquito: 'F' female

Spore.Count number of spores harboured by mosquito at time of death

t time of death to nearest half day

censored '1' censored, '0' died

d death indicator: '1' died during experiment, '0' right-censored

g infection treatment indicator: '0' uninfected, '1' infected

Source

Lorenz LM & Koella JC (2011) The microsporidian parasite Vavraia culicis as a potential late life-acting control agent of malaria. Evolutionary Applications 4: 783-790 doi

The full dataset is available at Dryad https://doi.org/10.5061/dryad.2s231

Examples

```
head(data_lorenz)
```

---

data_parker   *Full data from Parker et al (2014)*

Description

Complete data on the survival of adult female aphids exposed or unexposed to fungal infection.

Usage

```
data_parker
```

Format

An object of class `data.frame` with 328 rows and 11 columns.

Value

A dataframe

Genotype names of host genotypes

SD Spore Dose, concentration of fungal spores hosts were exposed to (spores/mm^2)

Fecundity total number of offspring produced by hosts over lifetime

Sporulation '1' visual signs of sporulation at host death, '0' no signs of sporulation

Status '0' censored, '1' died

Time time of death (days)

dose dose treatments on ordinal scale of 1-3, controls 0

censored '1' censored, '0' died

d death indicator: '1' died, '0' censored

t time of death (days)

g infection treatment indicator; '1' infected, '0' uninfected

Source

Parker BJ, Garcia JR, Gerardo NM (2014) Genetic variation in resistance and fecundity tolerance in a natural host-pathogen interaction. Evolution 68: 2421-2429 doi

The full dataset is available at Dryad https://doi.org/10.5061/dryad.24gq7

Examples

```
head(data_parker)
```

---

etd_infected   *Expected time of death: infected hosts*

Description

Time when infected hosts are expected to have died due to their cumulative exposure to background mortality and mortality due to infection.

Usage

```
etd_infected(a1, b1, a2, b2, d1 = "", d2 = "", tmax = 100)
```

Arguments

a1, b1	location & scale parameters for background mortality
a2, b2	location & scale parameters for mortality due to infection
d1, d2	character: name of probability distributions describing background mortality and mortality due to infection, respectively
tmax	numeric. Maximum value of t for which expected time of death is searched; defaults to 100. Minimum time is zero (0)

Details

It is the time t when, H1[t] + H2[t] = 1, and cumulative survival is, S1[t].S2[t] = exp(-1) = 0.367

Value

numeric

See Also

etd_uninfected

Examples

```
print(etd_infected(a1 = 2, b1 = 0.5, a2 = 30, b2 = 5,
                   d1 = "Weibull", d2 = "Gumbel")) # 7.34
print(etd_infected(a1 = 20, b1 = 5, a2 = 3, b2 = 0.6,
                   d1 = "Gumbel", d2 = "Frechet")) # 17.84
print(etd_infected(a1 = 3, b1 = 0.6, a2 = 2, b2 = 0.5,
                   d1 = "Frechet", d2 = "Weibull")) # 7.37
```

---

etd_uninfected   *Expected time of death: uninfected hosts*

Description

Time when uninfected hosts are expected to have died due to their cumulative exposure to background mortality.

Usage

```
etd_uninfected(a1, b1, d1 = "", tmax = 100)
```

Arguments

a1, b1	location & scale parameters for background mortality
d1	character: name of probability distribution describing background mortality
tmax	numeric. Maximum value of t for which expected time of death is searched; defaults to 100. Minimum time is zero (0)

Details

It is the time t when, H1[t] = 1, and cumulative survival is, S1[t] = exp(-1) = 0.367

Value

numeric

See Also

[etd_infected](#)

Examples

```
print(etd_uninfected(a1 = 2, b1 = 0.5, d1 = "Weibull")) # 7.38
print(etd_uninfected(a1 = 20, b1 = 5, d1 = "Gumbel")) # 20
print(etd_uninfected(a1 = 3, b1 = 0.6, d1 = "Frechet")) # 32.06
```

nll_basic   *Negative log-likelihood function: basic model*

Description

Function returning the negative log-likelihood (nll) for the 'basic' relative survival model, given the functions' parameters and the observed data.

Usage

```
nll_basic(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "Weibull",
  d2 = "Weibull"
)
```

Arguments

a1, b1	location and scale parameters for background mortality
a2, b2	location and scale parameters for mortality due to infection
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column identifying if event was death (0) or right-censoring (1)
infected_treatment	
	name of data frame column identifying if data are from an infected (1) or uninfected (0) treatment
d1, d2	names of probability distributions describing background mortality and mortality due to infection, respectively; both default to the Weibull distribution

Details

By deafult, this function takes arguments for location and scale parameters named; a1, b1, a2, b2. These parameters are components of survival functions describing patterns of background mortality and mortality due to infection. The particular form of these survival functions depends on the probability distributions chosen to describe each source of mortality; d1, d2. The function also takes arguments directing it to the data to be analysed and how they are labelled; data, time, censor, etc.

The nll returned by the function depends on the numerical values of the location and scale parameters, which determine how well the likelihood model describes the observed data. Maximum likelihood estimation functions, e.g., `mle2` of the package `bbmle`, can be used find values of the location and scale parameters minimising the model's nll. The resulting maximum likelihood estimates can be used to describe host mortality due to background mortality and mortality due to infection, including the pathogen's virulence.

The model assumes all the individuals in the infected population are infected. It is also assumes infections are homogeneous, i.e., each infection has the same influence on host survival. Consequently a single hazard function, with a single pair of values for its location and scale parameters, can be used to describe the pattern of mortality due to infection for the infected population as a whole.

Examples

```
# prepare subset of 'data_blanford'; treatments 'cont' and 'Bb06' of Block 3

  data01 <- subset(data_blanford,
    (data_blanford$block == 3) & (
      (data_blanford$treatment == 'cont') |
        (data_blanford$treatment == 'Bb06')) &
          (data_blanford$day > 0))

 head(data01, 4)

# step #1: 'prep function' linking 'nll_basic' to data
  # and identifying parameters to estimate
    m01_prep_function <- function(a1 = a1, b1 = b1, a2 = a2, b2 = b2){
      nll_basic(
        a1 = a1, b1 = b1, a2 = a2, b2 = b2,
        data = data01,
        time = t,
        censor = censor,
        infected_treatment = inf,
        d1 = 'Weibull', d2 = 'Weibull')
        }

# step #2: send 'prep_function' to 'mle2' for maximum likelihood estimation
  #  starting values specified as
    m01 <- mle2(m01_prep_function,
            start = list(a1 = 2, b1 = 0.5, a2 = 2.5, b2 = 0.25)
            )

    summary(m01)
```

---

nll_basic_logscale   *Negative log-likelihood function: basic model on logscale*

Description

Function returning the negative log-likelihood (nll) for the 'basic' relative survival model, given the functions' parameters and the observed data.

Usage

```
nll_basic_logscale(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "Weibull",
  d2 = "Weibull"
)
```

Arguments

a1, b1	location and scale parameters for background mortality on a logscale
a2, b2	location and scale parameters for mortality due to infection on a logscale
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column identifying if event was death (0) or right-censoring (1)
infected_treatment	
	name of data frame column identifying if data are from an infected (1) or uninfected (0) treatment
d1, d2	names of probability distributions describing background mortality and mortality due to infection, respectively; both default to the Weibull distribution

Details

By deafult, this function takes arguments for location and scale parameters named; a1, b1, a2, b2. These parameters are components of survival functions describing patterns of background mortality and mortality due to infection. The particular form of these survival functions depends on the probability distributions chosen to describe each source of mortality; d1, d2. The function also takes arguments directing it to the data to be analysed and how they are labelled; data, time, censor, etc.

The nll returned by the function depends on the numerical values of the location and scale parameters, which determine how well the likelihood model describes the observed data. Maximum likelihood estimation functions, e.g., mle2 of the package bbmle, can be used find values of the location and scale parameters minimising the model's nll. The resulting maximum likelihood estimates can be used to describe host mortality due to background mortality and mortality due to infection, including the pathogen's virulence.

The model assumes all the individuals in the infected population are infected. It is also assumes infections are homogeneous, i.e., each infection has the same influence on host survival. Consequently a single hazard function, with a single pair of values for its location and scale parameters, can be used to describe the pattern of mortality due to infection for the infected population as a whole.

Examples

```
# prepare subset of 'data_blanford'; treatments 'cont' and 'Bb06' of Block 3

  data01 <- subset(data_blanford,
    (data_blanford$block == 3) & (
      (data_blanford$treatment == 'cont') |
        (data_blanford$treatment == 'Bb06')) &
          (data_blanford$day > 0))

 head(data01, 4)

# step #1: 'prep function' linking 'nll_basic' to data
  # and identifying parameters to estimate
    m01_prep_function <- function(a1 = a1, b1 = b1, a2 = a2, b2 = b2){
      nll_basic_logscale(
        a1 = a1, b1 = b1, a2 = a2, b2 = b2,
        data = data01,
        time = t,
        censor = censor,
        infected_treatment = inf,
        d1 = 'Weibull', d2 = 'Weibull')
        }

# step #2: send 'prep_function' to mle2 for maximum likelihood estimation with
        #  starting values specified
    m01 <- mle2(m01_prep_function,
            start = list(a1 = 1, b1 = 1, a2 = 1, b2 = 1)
            )

    summary(m01)
```

---

nll_controls   *Negative log-likelihood function: control data only*

Description

Function returning negative log-likelihood (nll) for data in a control treatment.

Usage

```
nll_controls(
  a1 = a1,
```

```
  b1 = b1,
  data = data,
  time = time,
  censor = censor,
  d1 = "Weibull"
)
```

Arguments

a1, b1	location and scale parameters for background mortality
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column idenifying if event was death (0) or right-censoring (1)
d1	name of probability distribution describing background mortality. Choice of; 'Weibull', 'Gumbel', 'Fréchet'; defaults to the Weibull distribution

Details

This function returns the nll based on two parameters, the location and scale parameters used to describe background mortality.

Value

numeric

See Also

[nll_basic](#)

Examples

```
# prepare a subset of the Blanford data for analysis
  data01 <- subset(data_blanford,
    (data_blanford$block == 3) &
      (data_blanford$treatment == 'cont') &
        (data_blanford$day > 0))

# check data frame for names of columns
    head(data01)

# step #1: 'prep function' linking 'nll_controls' to data
  # and identifying parameters to estimate
    m01_prep_function <- function(a1 = a1, b1 = b1){
      nll_controls(
        a1 = a1, b1 = b1,
        data = data01,
        time = t,
        censor = censor,
        d1 = 'Weibull'
```

```
      )}

# step #2: send 'prep_function' to mle2 for maximum likelihood estimation
  # specifying starting values
    m01 <- mle2(m01_prep_function,
             start = list(a1 = 2, b1 = 0.5)
             )

    summary(m01)
```

---

nll_exposed_infected   *Negative log-likelihood function: exposed-infected*

Description

Function returning negative log-likelihood (nll) for patterns of mortality in infected and control treatments, where the infected population harbours an unobserved proportion of hosts that were exposed to infection, but did not become infected.

Usage

```
nll_exposed_infected(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  p1 = p1,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "Weibull",
  d2 = "Weibull"
)
```

Arguments

a1, b1	location and scale parameters for background mortality
a2, b2	location and scale parameters for mortality due to infection
p1	unobserved proportion of hosts exposed to infection and infected; $0 <= p1 <= 1$
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column idenifying if event was death (0) or right-censoring (1)

infected_treatment

          name of data frame column identifying if data are from an infected (1) or unin-
          fected (0) treatment

d1, d2   names of probability distributions chosen to describe background mortality and
         mortality due to infection, respectively'; both default to the Weibull distribution

Details

This function returns the nll based on five parameters, the location and scale parameters for back-
ground mortality and mortality due to infection, respectively, plus a parameter for the proportion of
hosts that became infected when exposed to infection.

Value

numeric

See Also

[nll_two_inf_subpops_obs](#) [nll_two_inf_subpops_unobs](#)

Examples

```
# check column names in head of data frame with data to analyse
    head(data_parker)

# step #1: prepare nll function for analysis
    m01_prep_function <- function(a1 = a1, b1 = b1, a2 = a2, b2 = b2, p1 = p1){
      nll_exposed_infected(
        a1 = a1, b1 = b1, a2 = a2, b2 = b2, p1 = p1,
        data = data_parker,
        time = t,
        censor = censored,
        infected_treatment = g,
        d1 = "Frechet",
        d2 = "Weibull")
        }

# step #2: send 'prep_function' to mle2 for maximum likelihood estimation
    m01 <- mle2(m01_prep_function,
            start = list(a1 = 2.5, b1 = 1, a2 = 2, b2 = 0.5, p1 = 0.5)
            )

    summary(m01)

# model setting lower & upper bounds to parameter estimates
  # including 0 < p1 < 1
    m02 <- mle2(m01_prep_function,
            start = list(a1 = 2.5, b1 = 1.2, a2 = 1.9, b2 = 0.16, p1 = 0.48),
            method = "L-BFGS-B",
            lower = c(a1 = 0, b1 = 0, a2 = 0, b2 = 0, p1 = 0),
            upper = c(a1 = Inf, b1 = Inf, a2 = Inf, b2 = Inf, p1 = 1),
            )
```

```
    summary(m02)
```

---

nll_frailty   *Negative log-likelihood function: frailty*

Description

Function calculating negative log-likelihood (nll) for the observed patterns of mortality in infected and uninfected treatments when it assumed there is unobserved variation in virulence.

Usage

```
nll_frailty(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  theta = theta,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "Weibull",
  d2 = "Weibull",
  d3 = ""
)
```

Arguments

a1, b1	location and scale parameters describing background mortality
a2, b2	location and scale parameters describing mortality due to infection
theta	parameter describing variance of unobserved variation in virulence
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column idenifying if event was death (0) or right-censoring (1)
infected_treatment	
	name of data frame column identifying if data are from an infected (1) or uninfected (0) treatment
d1, d2	names of probability distributions chosen to describe background mortality and mortality due to infection, respectively; both default to the Weibull distribution
d3	name of probability distribution chosen to describe unobserved frailty; choice of 'gamma' or 'inverse Gaussian'

Details

The function assumes the unobserved variation in the rate of mortality due to infection is continuously distributed and follows either the gamma distribution or the inverse Gaussian distribution, with mean = 1 and variance = theta.

The nll is based on five parameter functions for the location and scale parameters for background mortality and mortality due to infection, respectively, plus a parameter estimating the variance of the unobserved variation in virulence.

Value

numeric

Examples

```
### Example 1: unobserved variation in virulence described by gamma distribution

# step #1: parameterise nll function to be passed to 'mle2'
    m01_prep_function <- function(a1 = a1, b1 = b1, a2 = a2, b2 = b2, theta = theta){
      nll_frailty(
        a1 = a1, b1 = b1, a2 = a2, b2 = b2, theta = theta,
        data = data_lorenz,
        time = t,
        censor = censored,
        infected_treatment = g,
        d1 = "Gumbel",
        d2 = "Weibull",
        d3 = "Gamma"
        )}

# step #2: send 'prep_function' to 'mle2' for maximum likelihood estimation
    m01 <- mle2(
      m01_prep_function,
      start = list(a1 = 20, b1 = 5, a2 = 3, b2 = 0.1, theta = 2)
      )

    summary(m01)

### Example 2: unobserved variation in virulence described by inverse Gaussian distribution

    m02_prep_function <- function(a1 = a1, b1 = b1, a2 = a2, b2 = b2, theta = theta){
      nll_frailty(
        a1 = a1, b1 = b1, a2 = a2, b2 = b2, theta = theta,
        data = data_lorenz,
        time = t,
        censor = censored,
        infected_treatment = g,
        d1 = "Gumbel",
        d2 = "Weibull",
        d3 = "Inverse Gaussian"
        )}
```

```
    m02 <- mle2(
      m02_prep_function,
        start = list(a1 = 20, b1 = 5, a2 = 3, b2 = 0.1, theta = 2)
        )

    summary(m02)

# compare model estimates by AICc
    AICc(m01, m02, nobs = 256)
```

---

nll_frailty_correlated

Negative log-likelihood function: correlated frailty model

Description

Function calculating the negative log-likelihood (nll) for patterns of mortality in infected and unin-fected treatments when there is unobserved variation in background mortality and mortality due to infection, where these two sources of variation are positively correlated.

Usage

```
nll_frailty_correlated(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  theta01 = theta01,
  theta02 = theta02,
  rho = rho,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "Weibull",
  d2 = "Weibull"
)
```

Arguments

a1, b1	location and scale parameters describing background mortality
a2, b2	location and scale parameters describing mortality due to infection
theta01	parameter describing variance of unobserved variation in background rate of mortality
theta02	parameter describing variance of unobserved variation in rate of mortality due to infection

rho	parameter for strength of positive correlation between theta01 and theta02; rho > 0
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column idenifying if event was death (0) or right-censoring (1)
infected_treatment	
	name of data frame column identifying if data are from an infected (1) or uninfected (0) treatment
d1, d2	names of probability distributions chosen to describe background mortality and mortality due to infection, respectively; both default to the Weibull distribution

Details

This function assumes both sources of unobserved variation follow the Gamma distribution where both have a mean = 1.0, and variances 'theta01' and 'theta02', respectively. The nll function is calculated based on seven parameters; location and scale parameters for background mortality and mortality due to infection, respectively, the unobserved variance of in each source of mortality, and the strength of the positive correlation between them.

See Also

[nll_frailty](#)

Examples

```
# step #1: parameterise nll function to be passed to 'mle2'
   m01_prep_function <- function(
     a1 = a1, b1 = b1, a2 = a2, b2 = b2,
     theta01 = theta01, theta02 = theta02, rho = rho){
       nll_frailty_correlated(
         a1 = a1, b1 = b1, a2 = a2, b2 = b2,
         theta01 = theta01, theta02 = theta02, rho = rho,
         data = data_lorenz,
         time = t,
         censor = censored,
         infected_treatment = g,
         d1 = "Gumbel",
         d2 = "Gumbel"
         )}

# step #2: send 'prep_function' to 'mle2' for maximum likelihood estimation
#          lower bounds of estimates set to 1e-6
  m01 <- mle2(m01_prep_function,
    start = list(a1 = 20, b1 = 5, a2 = 20, b2 = 4,
                 theta01 = 1, theta02 = 1, rho = 1),
    method = "L-BFGS-B",
    lower = list(a1 = 1e-6, b1 = 1e-6, a2 = 1e-6, b2 = 1e-6,
                 theta01 = 1e-6, theta02 = 1e-6, rho = 1e-6)
```

```
                    )

   summary(m01)

# NB no std. errors estimated and estimate of 'theta01' at lower boundary

# rerun model with theta01 set at lower boundary
    m02 <- mle2(m01_prep_function,
      start = list(a1 = 20, b1 = 5, a2 = 20, b2 = 4,
                   theta01 = 1, theta02 = 1, rho = 1),
      fixed = list(theta01 = 1e-6),
      method = "L-BFGS-B",
      lower = list(a1 = 1e-6, b1 = 1e-6, a2 = 1e-6,
                   b2 = 1e-6, theta02 = 1e-6, rho = 1e-6)
      )

summary(m02)

# NB std. error of 'rho' crosses lower boundary

# rerun model with theta01 and rho set at lower limits
    m03 <- mle2(m01_prep_function,
      start = list(a1 = 20, b1 = 5, a2 = 20, b2 = 4,
                   theta01 = 1, theta02 = 1, rho = 1),
      fixed = list(theta01 = 1e-6, rho = 1e-6),
      method = "L-BFGS-B",
      lower = list(a1 = 1e-6, b1 = 1e-6, a2 = 1e-6,
                   b2 = 1e-6, theta02 = 1e-6)
      )

    summary(m03)

# result of m03 corresponds with estimates from 'nll_frailty' model,
  # i.e., where it is assumed there is no unobserved variation in the
  # rate of background mortality and where the gamma distribution describes
  # the unobserved variation in virulence

# step #1: parameterise nll function to be passed to 'mle2'
    m04_prep_function <- function(a1, b1, a2, b2, theta){
      nll_frailty(a1, b1, a2, b2, theta,
        data = data_lorenz,
        time = t,
        censor = censored,
        infected_treatment = g,
        d1 = "Gumbel",
        d2 = "Gumbel",
        d3 = "Gamma"
        )}

# step #2: send 'prep_function' to 'mle2' for maximum likelihood estimation
    m04 <- mle2(m04_prep_function,
      start = list(a1 = 20, b1 = 5, a2 = 20, b2 = 5, theta = 2)
      )
```

```
    summary(m04)

# compare estimated values m03 vs. m04
    coef(m03) ; coef(m04)
```

---

nll_frailty_logscale   *Negative log-likelihood function: frailty variables on logscale*

Description

This negative log-likelihood (nll) function is the same as 'nll_frailty', except it assumes the variables to estimate are on a logscale.

Usage

```
nll_frailty_logscale(
  log.a1 = log.a1,
  log.b1 = log.b1,
  log.a2 = log.a2,
  log.b2 = log.b2,
  log.theta = log.theta,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "Weibull",
  d2 = "Weibull",
  d3 = ""
)
```

Arguments

log.a1, log.b1	location and scale parameters for background mortality, on a logscale
log.a2, log.b2	location and scale parameters for mortality due to infection, on a logscale
log.theta	parameter describing variance of the unobserved variation in virulence, on a logscale
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column idenifying if event was death (0) or right-censoring (1)
infected_treatment	
	name of data frame column identifying if data are from an infected (1) or uninfected (0) treatment

d1, d2	names of probability distributions chosen to describe background mortality and mortality due to infection, respectively; both default to the Weibull distribution
d3	name of probability distribution chosen to describe unobserved frailty; choice of 'gamma' or 'inverse Gaussian'

Value

numeric

See Also

nll_frailty

Examples

```
### Example 1: unobserved variation in virulence with gamma distribution

# step #1: parameterise nll function to be passed to 'mle2'
    m01_prep_function <- function(
      log.a1 = log.a1, log.b1 = log.b1,
      log.a2 = log.a2, log.b2 = log.b2,
      log.theta = log.theta){
        nll_frailty_logscale(
          log.a1 = log.a1, log.b1 = log.b1,
          log.a2 = log.a2, log.b2 = log.b2,
          log.theta = log.theta,
          data = data_lorenz,
          time = t,
          censor = censored,
          infected_treatment = g,
          d1 = "Gumbel", d2 = "Weibull", d3 = "Gamma"
          )}

# step #2: send 'prep_function' to 'mle2' for maximum likelihood estimation
    m01 <- mle2(
      m01_prep_function,
      start = list(
        log.a1 = 3, log.b1 = 1.5, log.a2 = 0.7, log.b2 = -0.7, log.theta = 1
        )
      )

  summary(m01)

  exp(coef(m01))
```

---

nll_frailty_shared   *Negative log-likelihood function: frailty shared*

Description

Function calculating negative log-likelihood (nll) for patterns of mortality in infected and unin-
fected treatments where unobserved variation is assumed to act equally on background mortality
and mortality due to infection.

Usage

```
nll_frailty_shared(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  theta = theta,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "",
  d2 = ""
)
```

Arguments

a1, b1	location and scale parameters for background mortality
a2, b2	location and scale parameters for mortality due to infection
theta	parameter describing variance of unobserved variation acting on mortality rates
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column idenifying if event was death (0) or right-censoring (1)
infected_treatment	
	name of data frame column identifying if data are from an infected (1) or unin-fected (0) treatment
d1, d2	names of probability distributions chosen to describe background mortality and mortality due to infection, respectively; both default to the Weibull distribution

Details

This function assumes unobserved variation acting on both the background rate of mortality and the
rate of mortality due to infection is continuously distributed and follows the gamma distribution,
with mean = 1.0 and variance = theta. The function returns the nll based on five parameters; the
location and scale parameters for background mortality and mortality due to infection, plus the
parameter describing the variance of the unobserved variation.

Value

numeric

See Also

nll_frailty

Examples

```
# step #1: prepare nll function for analysis
  m01_prep_function <- function(a1 = a1, b1 = b1, a2 = a2, b2 = b2, theta = theta){
    nll_frailty_shared(a1 = a1, b1 = b1, a2 = a2, b2 = b2, theta = theta,
      data = data_lorenz,
      time = t,
      censor = censored,
      infected_treatment = g,
      d1 = "Gumbel", d2 = "Gumbel"
      )}

# step #2: send 'prep_function' to mle2 for maximum likelihood estimation,
  # specifying starting values
  m01 <- mle2(m01_prep_function,
            start = list(a1 = 23, b1 = 5, a2 = 10, b2 = 1, theta = 1),
            method = "Nelder-Mead",
            control = list(maxit = 5000)
            )

  summary(m01)
```

---

nll_proportional_virulence

Negative log-likelihood function: nll proportional virulence

Description

Function assuming hazard functions describing virulence are proportional among infected treatments.

Usage

```
nll_proportional_virulence(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  theta = theta,
```

```
    data = data,
    time = time,
    censor = censor,
    infected_treatment = infected_treatment,
    reference_virulence = reference_virulence,
    d1 = "Weibull",
    d2 = "Weibull"
)
```

Arguments

`a1, b1`	location and scale parameters describing background mortality
`a2, b2`	location and scale parameters describing mortality due to infection
`theta`	a constant describing the proportional relationship of virulence; theta > 0
`data`	name of data frame containing survival data
`time`	name of data frame column identifying time of event; time > 0
`censor`	name of data frame column identifying if event was death (0) or right-censoring (1)
`infected_treatment`	
	name of data frame column identifying if data are from an infected (1) or uninfected (0) treatment
`reference_virulence`	
	name of data frame column identifying the infected treatment to use as a reference when estimating virulence; value of 1 the reference treatment and 2 for treatment to be compared
`d1, d2`	names of probability distributions describing background mortality and mortality due to infection, respectively; both default to the Weibull distribution

Details

The proportional hazards assumption requires, h1(t) / h2(t) = c, where h1(t) and h2(t) are two hazard functions at time t, and c is a constant. In this function the hazard functions for the increased mortality due to infection are assumed to be proportional for different infection treatments within the same experiment.

In the default form, one infected treatment is denoted '1' and used as the reference treatment for virulence against which the virulence in a second infected population, '2', is scaled. The default function can be extended to compare multiple treatments against the reference (see vignette: Worked examples II)

Examples

```
data01 <- data_lorenz

# create column 'reference_virulence' with values of 1 and 2 when
  # Infectious.dose = 5000 and 160000, respectively, otherwise 0

data01$reference_virulence <- ifelse(data01$g == 0, 0,
```

```
  ifelse(data01$g == 1, ifelse(data01$Infectious.dose == 5000, 1,
    ifelse(data01$Infectious.dose == 160000, 2, 0)), 0)
 )

m01_prep_function <- function(
  a1 = a1, b1 = b1, a2 = a2, b2 = b2, theta = theta){
    nll_proportional_virulence(
      a1 = a1, b1 = b1, a2 = a2, b2 = b2, theta = theta,
      data = data01,
      time = t,
      censor = censored,
      infected_treatment = g,
      reference_virulence = reference_virulence,
      d1 = 'Gumbel', d2 = 'Weibull')
      }

m01 <- mle2(m01_prep_function,
  start = list(a1 = 23, b1 = 5, a2 = 4, b2 = 0.2, theta = 1)
  )

summary(m01)

# virulence in the high dose treatment is estimated as
 # ~ 6x greater than in the low dose treatment
```

---

nll_recovery   *Negative log-likelihood function: recovery model*

Description

Function returning negative log-likelihood (nll) for patterns of survival in infected and uninfected treatments, when infected hosts can recover from infection.

Usage

```
nll_recovery(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  a3 = a3,
  b3 = b3,
  data = data,
  d1 = "",
  d2 = "",
  d3 = ""
)
```

Arguments

a1, b1	location & scale parameters for background mortality
a2, b2	location & scale parameters for mortality due to infection
a3, b3	location & scale parameters for how long infection 'survives'
data	a data.frame with the data
d1, d2, d3	probability distributions for background mortality, mortality due to infection & how long infection 'survives' ("Weibull", "Gumbel", "Frechet")

Details

This model assumes all the hosts in an infected treatment are all initially infected, and they can all potentially recover from infection. Recovered hosts are assumed to only experience background mortality equivalent to that experienced by matching uninfected or control individuals; no assumptions are made as to whether they are still infected or infectious. It is also assumed that the timing of recovery from infection is not directly observed, but an individual's infected/recovery status can be determined after they have died or been censored.

The probability that an infection 'survives' over time, i.e., the host does not recover from infection, is assumed to follow a probability distribution which acts independently of the probability distributions determining background mortality or mortality due to infection.

This function only estimates location and scale parameters as constants, it is not designed to estimate them as functions.

Value

numeric

Warning

requires the data to be specified in a specific format; see vignette 'data format' for details

Examples

```
# NB the data to analyse needs to be in a data frame of a specific form
  head(recovery_data)

# step #1: prepare nll function for analysis
  m01_prep_function <- function(a1, b1, a2, b2, a3, b3){
    nll_recovery(a1, b1, a2, b2, a3, b3,
                 data = recovery_data,
                 d1 = "Weibull", d2 = "Weibull", d3 = "Weibull"
                 )}

# step #2: send 'prep_function' to mle2 for maximum likelihood estimation,
  # specifying starting values
  m01 <- mle2(m01_prep_function,
    start = list(a1 = 2, b1 = 0.5, a2 = 2, b2 = 0.5, a3 = 2, b3 = 0.5)
    )
```

```
    summary(m01)

  # values used to simulate data were for the Weibull distribution;
    # a1 = 2.8, b1 = 0.5, a2 = 2.2, b2 = 0.35, a3 = 2.35, b3 = 0.35
```

---

nll_recovery_II   *Negative log-likelihood function: recovery model, no background mortality*

Description

Function returning negative log-likelihood (nll) for patterns of survival in infected and uninfected treatments, when infected hosts can recover from infection and there is no background mortality.

Usage

```
nll_recovery_II(
  a2 = a2,
  b2 = b2,
  a3 = a3,
  b3 = b3,
  data = data,
  d2 = "",
  d3 = ""
)
```

Arguments

a2, b2	location & scale parameters for mortality due to infection
a3, b3	location & scale parameters for how long infection 'survives'
data	a data.frame with the data
d2, d3	probability distributions for background mortality, mortality due to infection & how long infection 'survives' ("Weibull", "Gumbel", "Frechet")

Details

This model assumes all the hosts in an infected treatment are all initially infected, and they can all potentially recover from infection. Uninfected, infected and recovered hosts are assumed to not experience any background mortality during the period of the experiment. No assumptions are made as to whether recovered hosts are still infected or infectious. It is also assumed that the timing of recovery from infection is not directly observed, but and individual's infected/recovery status can be determined after they have died or been censored.

The probability that an infection 'survives' over time, i.e., the host does not recover from infection, is assumed to follow a probability distribution which acts independently of the probability distributions determining background mortality or mortality due to infection.

This function only estimates location and scale parameters as constants, it is not designed to estimate them as functions.

The nll function also requires the data to be specified in a specific format and requires columns detailing when and how many control individuals were right-censored, even though these individuals do not contribute to the nll estimated; see vignettes for details.

Value

numeric

Warning

requires the data to be specified in a specific format; see vignette 'data format' for details

Examples

```
# NB the data to analyse needs to be in a data frame of a specific form
    head(recovery_data_II)

# step #1: prepare nll function for analysis
    m01_prep_function <- function(a2, b2, a3, b3){
      nll_recovery_II(a2, b2, a3, b3,
                    data = recovery_data_II, # data_recovery_II,
                    d2 = "Weibull", d3 = "Weibull"
                  )}

# step #2: send 'prep_function' to mle2 for maximum likelihood estimation,
  # specifying starting values
    m01 <- mle2(m01_prep_function,
              start = list(a2 = 2, b2 = 0.5, a3 = 2, b3 = 0.5)
              )

    summary(m01)

# values used to simulate data were for the Weibull distribution;
  # a2 = 2.2, b2 = 0.35, a3 = 2.35, b3 = 0.35
```

---

nll_two_inf_subpops_obs

Negative log-likelihood function: two observed subpopulations of infected hosts

---

Description

Function returning negative log-likelihood (nll) for patterns of mortality in infected and uninfected treatments when an infected population harbours two identified, or 'observed', subpopulations of hosts experiencing different patterns of virulence, e.g. with/without visible signs of infection.

Usage

```
nll_two_inf_subpops_obs(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  a3 = a3,
  b3 = b3,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "Weibull",
  d2 = "Weibull",
  d3 = "Weibull",
  infsubpop = infsubpop
)
```

Arguments

a1, b1	location and scale parameters describing background mortality
a2, b2	location and scale parameters describing mortality due to infection in one sub-population
a3, b3	location and scale parameters describing mortality due to infection in the other subpopulation
data	name of data frame containing survival data
time	name of data frame column identifying time of event; time > 0
censor	name of data frame column idenifying if event was death (0) or right-censoring (1)
infected_treatment	
	name of data frame column identifying if data are from an infected (1) or uninfected (0) treatment
d1, d2, d3	names of probability distributions chosen to describe background mortality and mortality due to infection, respectively; each defaults to the Weibull distribution
infsubpop	name of data frame column identifying the two subpopulations of infected hosts; '1' or '2'

Details

The nll is based on six parameters, the location and scale parameters for background mortality, plus separate location and scale parameters for each of the two infected subpopulations.

It is assumed the patterns of mortality within each subpopulation act independently of one another.

See Also

[nll_exposed_infected](#) [nll_two_inf_subpops_unobs](#)

Examples

```
# example using data from Parker et al
    data01 <- data_parker

# create column 'infsubpop' in data01, fill with '0'
    data01$infsubpop <- 0

# infsubpop = '1' for individuals in infected treatments (g == 1)
  # with visible signs of sporulation (Sporulation = 1)
# infsubpop = '2' for individuals in infected treatments (g == 1)
  # with no visible signs of sporulation (Sporulation = 0)
    data01$infsubpop[data01$g == 1 & data01$Sporulation == 1] <- 1
    data01$infsubpop[data01$g == 1 & data01$Sporulation == 0] <- 2

    head(data01)

# step #1: parameterise nll function to be passed to 'mle2'
    m01_prep_function <- function(
      a1 = a1, b1 = b1, a2 = a2, b2 = b2, a3 = a3, b3 = b3){
        nll_two_inf_subpops_obs(
          a1 = a1, b1 = b1, a2 = a2, b2 = b2, a3 = a3, b3 = b3,
          data = data01,
          time = t,
          censor = censored,
          infected_treatment = g,
          d1 = "Frechet",
          d2 = "Weibull",
          d3 = "Weibull",
          infsubpop = infsubpop
        )}

# step #2: send 'prep_function' to 'mle2' for maximum likelihood estimation
    m01 <- mle2(
      m01_prep_function,
      start = list(a1 = 3, b1 = 1, a2 = 2, b2 = 0.5, a3 = 2, b3 = 0.5)
      )

    summary(m01)
```

---

nll_two_inf_subpops_unobs

Negative log-likelihood function: two unobserved subpopulations of infected hosts

---

Description

Function returning negative log-likelihood (nll) for patterns of mortality in infected and uninfected treatments when an infected population is assumed to harbour two distinct subpopulations of hosts

experiencing different virulence. The nll is based on seven parameters, the location and scale parameters for background mortality, separate location and scale parameters for each of the two infected subpopulations, and a parameter estimating the proportions of the two subpopulations

Usage

```
nll_two_inf_subpops_unobs(
  a1 = a1,
  b1 = b1,
  a2 = a2,
  b2 = b2,
  a3 = a3,
  b3 = b3,
  p1 = p1,
  data = data,
  time = time,
  censor = censor,
  infected_treatment = infected_treatment,
  d1 = "Weibull",
  d2 = "Weibull",
  d3 = "Weibull"
)
```

Arguments

`a1, b1`	location and scale parameters describing background mortality
`a2, b2`	location and scale parameters describing mortality due to infection in one subpopulation
`a3, b3`	location and scale parameters describing mortality due to infection in the other subpopulation
`p1`	parameter estimating the proportion of infected hosts in the first of the two subpopulations; 0 <= p1 <= 1
`data`	name of data frame containing survival data
`time`	name of data frame column identifying time of event; time > 0
`censor`	name of data frame column idenifying if event was death (0) or right-censoring (1)
`infected_treatment`	
	name of data frame column identifying if data are from an infected (1) or uninfected (0) treatment
`d1, d2, d3`	names of probability distributions chosen to describe background mortality and mortality due to infection in each subpopulation, respectively; defaults to the Weibull distribution

Details

p1 is the estimated proportion of hosts associated with the location and scale parameters a2, b2; 0 <= p1 <= 1.

It is assumed the patterns of mortality within each subpopulation act independently of one another.

See Also

nll_exposed_infected nll_two_inf_subpops_obs

Examples

```
# example using data from Parker et al
    data01 <- data_parker

# step #1: parameterise nll function to be passed to 'mle2'
    m01_prep_function <- function(
      a1 = a1, b1 = b1, a2 = a2, b2 = b2, a3 = a3, b3 = b3, p1 = p1){
        nll_two_inf_subpops_unobs(
          a1 = a1, b1 = b1, a2 = a2, b2 = b2, a3 = a3, b3 = b3, p1 = p1,
          data = data01,
          time = t,
          censor = censored,
          infected_treatment = g,
          d1 = "Frechet",
          d2 = "Weibull",
          d3 = "Weibull"
          )}

# step #2: send 'prep_function' to 'mle2' for maximum likelihood estimation
    m01 <- mle2(
      m01_prep_function,
      start = list(a1 = 2, b1 = 1,
                   a2 = 2, b2 = 0.3,
                   a3 = 2, b3 = 0.7,
                   p1 = 0.5)
      )

    summary(m01)
```

---

recovery_data   *Simulated recovery data*

Description

Simulated data allowing for recovery from infection, where recovered individuals experience the same background mortality as uninfected individuals.

Usage

```
recovery_data
```

Format

An object of class data.frame with 120 rows and 10 columns.

Value

A dataframe

columns 1-2 indicator variables identifying individuals in the control treatment that died (control.d = 1) or were censored (control.c = 1)

columns 3-4 indicator variables identifying individuals in the infected treatment that died (infected.d = 1) or were censored (infected.c = 1) while still infected

columns 5-6 indicator variables indenifying individuals in the infected treatment that died (recovered.d = 1) or were censored (recovered.c = 1) after recovering from infection

censor '0' not censored, '1' censored

d death indicator: '1' died, '0'

t time of the event (death or censoring)

fq frequence or number of individuals associated with the event of death or censoring at time t

Examples

```
head(recovery_data) ; tail(recovery_data)
```

---

recovery_data_II   *Simulated recovery data, with no background mortality*

Description

Simulated data allowing for recovery from infection, when there is no background mortality.

Usage

```
recovery_data_II
```

Format

An object of class `data.frame` with 120 rows and 10 columns.

Value

A dataframe

columns 1-2 indicator variables identifying individuals in the control treatment that died (control.d = 1) or were censored (control.c = 1)

columns 3-4 indicator variables identifying individuals in the infected treatment that died (infected.d = 1) or were censored (infected.c = 1) while still infected

columns 5-6 indicator variables indenifying individuals in the infected treatment that died (recovered.d = 1) or were censored (recovered.c = 1) after recovering from infection

censor '0' not censored, '1' censored

d death indicator: '1' died, '0'

t time of the event (death or censoring)

fq frequence or number of individuals associated with the event of death or censoring at time t

Examples

```
head(recovery_data_II) ; tail(recovery_data_II)
```

---

sim_data_nll_basic   *Function simulating survival data for nll_basic*

Description

Function simulating survival data corresponding with assumptions of nll_basic

Usage

```
sim_data_nll_basic(
  a1 = a1,
  b1 = b1,
  n1 = n1,
  a2 = a2,
  b2 = b2,
  n2 = n2,
  t_censor = 1000,
  d1 = "",
  d2 = ""
)
```

Arguments

a1, b1, a2, b2	location and scale parameters describing background mortality and mortality due to infection, respectively; values must be > 0
n1, n2	size of populations to simulate data; uninfected and infected, respectively; values must be > 0
t_censor	time when any individuals still alive are right-censored; defaults to 1000 if not specified, must be > 0
d1, d2	probability distribution(s) chosen to describe background mortality and mortality due to infection, respectively; choice of, Weibull, Gumbel, Fréchet

Details

To generate data, the function collects the input variables for the probability distributions chosen to describe the background mortality and mortality due to infection, along with values for their location and scale parameters, respectively. These are used to parameterise cumulative survival functions for the two sources of mortality. These functions are solved for time t, where the value of S(t) is drawn from a random uniform distribution between 0 and 1; this yields values of t corresponding with values of survival drawn at random from the cumulative survival curve.

Values of t are rounded up to the nearest integer. This introduces a bias as recorded times of death are later than actual times of death; this bias occurs in real datasets when sampling occurs at discrete intervals. This bias can be partially offset by taking the mid-point of sampling intervals as the time

of death. Mid-point times of death are also provided, assuming sampling occurs at each integer interval.

In the case of hosts from infected treatments, two potential times of death are calculated; that due to background mortality and that due to infection. The earlier of the two defines the time of death, providing it is not later than the value of t_censor.

The value of t_censor defines the time when any individuals remaining alive are right-censored. The act of censoring is assumed to occur after populations have been sampled for mortality, i.e., if mortality is recorded daily and the last day of an experiment is day 14, the populations are checked for mortality at the beginning of day 14 and any individuals alive after this are classed as censored on day 14. The timing of censoring is 'true' as individuals were known to be alive at the time they were censored. Hence times of censoring do not vary for 'time' or 'mid-time', whereas they do for individuals dying at an unknown time between two consecutive sampling times.

Warning

The lower tail of Gumbel distribution can yield negative estimates for times of death as the random variable replacing cumulative survival approaches one; S(t) -> 1.

Examples

```
set.seed(34)

sim_data <- sim_data_nll_basic(
  a1 = 2.5, b1 = 0.9, n1 = 100, d1 = "Weibull",
  a2 = 2.0, b2 = 0.5, n2 = 100, d2 = "Weibull",
  t_censor = 14)

head(sim_data, 10)

sim_data$time[sim_data$infected_treatment == 0]

sim_data$time[sim_data$infected_treatment == 1]

# plot histogram for ages at death in infected population
   hist(
     sim_data$time[(sim_data$infected_treatment == 1 & sim_data$censor == 0)],
     breaks = seq(0, 14, 1),
     xlim = c(0, 15),
     main = 'infected, died',
     xlab = 'time of death',
     xaxt = 'n')
   axis(side = 1, tick = TRUE, at = seq(0, 14, 1))

# estimate location and scale parameters of simulated data with nll_basic
   m01_prep_function <- function(a1, b1, a2, b2){
     nll_basic(a1, b1, a2, b2,
       data = sim_data,
       time = time,
       censor = censor,
       infected_treatment = infected_treatment,
       d1 = 'Weibull', d2 = 'Weibull'
```

```
        )}

    m01 <- mle2(m01_prep_function,
             start = list(a1 = 2, b1 = 1, a2 = 2, b2 = 1)
             )

    confint(m01)

# estimate using 'mid_time' instead of 'time'
    m02_prep_function <- function(a1, b1, a2, b2){
      nll_basic(a1, b1, a2, b2,
        data = sim_data,
        time = mid_time,
        censor = censor,
        infected_treatment = infected_treatment,
        d1 = 'Weibull', d2 = 'Weibull'
        )}

    m02 <- mle2(m02_prep_function,
             start = list(a1 = 2, b1 = 1, a2 = 2, b2 = 1)
             )

    confint(m02)

  # compare estimates
    AICc(m01, m02, nobs = 200)
    # for these data, m02 based on 'mid-time' provides a better
    # description of the data
```

Index