Package 'finalsize'

May 10, 2023

Title Calculate the Final Size of an Epidemic

Version 0.2.0

Description Calculate the final size of a susceptible-infectious-recovered epidemic in a population with demographic variation in contact patterns and susceptibility to disease, as discussed in Miller (2012) <doi:10.1007/s11538-012-9749-6>.

License MIT + file LICENSE

URL https://github.com/epiverse-trace/finalsize,

https://epiverse-trace.github.io/finalsize/

BugReports https://github.com/epiverse-trace/finalsize/issues

Imports checkmate, Rcpp

LinkingTo Rcpp, RcppEigen

Suggests colorspace, covr, dplyr, knitr, bookdown, scales, socialmixr, testthat (>= 3.0.0), tibble, xml2, ggplot2, ggtext, rmarkdown, usethis, forcats, purrr, tidyr

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.2.3

VignetteBuilder knitr

Depends R (>= 2.10)

LazyData true

NeedsCompilation yes

Author Pratik Gupte [aut, cre, cph] (<https://orcid.org/0000-0001-5294-7819>),
Edwin Van Leeuwen [aut, cph] (<https://orcid.org/0000-0002-2383-5305>),
Adam Kucharski [aut, cph] (<https://orcid.org/0000-0001-8814-9421>),
Rosalind Eggo [ctb] (<https://orcid.org/0000-0002-0362-6717>),
Hugo Gruson [ctb] (<https://orcid.org/0000-0002-4094-1476>),
Thibaut Jombart [ctb] (<https://orcid.org/0000-0003-3796-2097>),
Andree Valle-Campos [ctb] (<https://orcid.org/0000-0002-7779-481X>),
Joshua W. Lambert [rev] (<https://orcid.org/0000-0001-5218-3046>)

Maintainer Pratik Gupte <pratik.gupte@lshtm.ac.uk>

Repository CRAN

Date/Publication 2023-05-09 23:10:02 UTC

R topics documented:

.final_size   *Calculate the final size of an epidemic*

Description

An internal function that interfaces between the R function `final_size()` and functions in the package header.

Usage

```
.final_size(parameters)
```

Arguments

parameters   A named list of parameters for the final size calculation. See the R function
              documentation for details and input checking.

---

final_size   *Final size of an epidemic*

Description

`final_size` calculates the final size of an epidemic outbreak in a population with heterogeneous mixing, and with heterogeneous susceptibility to infection such as that conferred by an immunisation programme.

Usage

```
final_size(
  r0,
  contact_matrix,
  demography_vector,
  susceptibility,
  p_susceptibility,
  solver = c("iterative", "newton"),
  control = list()
)
```

Arguments

r0	The basic reproductive number R_0 of the disease.
contact_matrix	Social contact matrix. Entry m_{ij} gives average number of contacts in group i reported by participants in group j
demography_vector	Demography vector. Entry v_i gives proportion of total population in group i (model will normalise if needed).
susceptibility	A matrix giving the susceptibility of individuals in demographic group i and risk group k.
p_susceptibility	A matrix giving the probability that an individual in demography group i is in risk (or susceptibility) group k. Each row represents the overall distribution of individuals in demographic group i across risk groups, and each row *must sum to 1.0*.
solver	Which solver to use. Options are "iterative" (default) or "newton", for the iterative solver, or the Newton solver, respectively. Special conditions apply when using the Newton solver, see the control argument.
control	A list of named solver options, see *Solver options*.

Value

A data.frame of the proportion of infected individuals, within each demography group and susceptibility group combination. If the demography groups and susceptibility groups are named, these names are added to relevant columns. If the groups are not named, synthetic names are added of the form demo_grp_<i>, for each demographic group i.

Solver options

The control argument accepts a list of solver options, with the iterative solver taking two extra arguments than the Newton solver. This is an optional argument, and default options are used within the solver functions if an argument is missing. Arguments provided override the solver defaults.

Common options:

1. iterations: The number of iterations over which to solve for the final size, unless the error is below the solver tolerance. Default = 10000.

2. `tolerance`: The solver tolerance; solving for final size ends when the error drops below this tolerance. Defaults to set `1e-6`. Larger tolerance values are likely to lead to inaccurate final size estimates.

Iterative solver options:

1. `step_rate`: The solver step rate. Defaults to 1.9 as a value found to work well.
2. `adapt_step`: Boolean, whether the solver step rate should be changed based on the solver error. Defaults to TRUE.

Examples

```
# load example POLYMOD data included in the package
data(polymod_uk)
r0 <- 2.0
contact_matrix <- polymod_uk$contact_matrix
demography_vector <- polymod_uk$demography_vector

# define the number of age and susceptibility groups
n_demo_grps <- length(demography_vector)
n_risk_grps <- 3

# In this example, all risk groups from all age groups are fully
# susceptible
susceptibility <- matrix(
  data = 1, nrow = n_demo_grps, ncol = n_risk_grps
)

p_susceptibility <- matrix(
  data = 1, nrow = n_demo_grps, ncol = n_risk_grps
)
# p_susceptibility rows must sum to 1.0
p_susceptibility <- p_susceptibility / rowSums(p_susceptibility)

# using default arguments for `solver` and `control`
final_size(
  r0 = r0,
  contact_matrix = contact_matrix,
  demography_vector = demography_vector,
  susceptibility = susceptibility,
  p_susceptibility = p_susceptibility
)

# using manually specified solver settings for the iterative solver
control <- list(
  iterations = 100,
  tolerance = 1e-3,
  step_rate = 1.9,
  adapt_step = TRUE
)

final_size(
  r0 = r0,
```

```
    contact_matrix = contact_matrix,
    demography_vector = demography_vector,
    susceptibility = susceptibility,
    p_susceptibility = p_susceptibility,
    solver = "iterative",
    control = control
  )

  # manual settings for the newton solver
  control <- list(
    iterations = 100,
    tolerance = 1e-3
  )

  final_size(
    r0 = r0,
    contact_matrix = contact_matrix,
    demography_vector = demography_vector,
    susceptibility = susceptibility,
    p_susceptibility = p_susceptibility,
    solver = "newton",
    control = control
  )
```

---

lambda_to_r0   *Calculate R_0 from transmission rate (λ)*

Description

Uses the transmission rate (λ), social contacts matrix (c), demography (N), the distribution P of each demographic group i into susceptibility groups S, and the infectious period (γ) to calculate the R_0 using the following equation.

$$R_0 = Max(EV(C)) \times \lambda\gamma$$

where $EV(C)$ denotes the eigenvalues of the matrix C which is calculated from the social contacts matrix scaled by the number of individuals in each demographic and susceptibility group in the population.

Usage

```
lambda_to_r0(
  lambda,
  contact_matrix,
  demography_vector,
  susceptibility,
  p_susceptibility,
  infectious_period = 1.8
)
```

Arguments

lambda	The transmission rate of the disease, also called the 'force of infection' (λ). This is different from the effective transmission rate (β).
contact_matrix	Social contact matrix. Entry m_{ij} gives average number of contacts in group i reported by participants in group j
demography_vector	
	Demography vector. Entry v_i gives proportion of total population in group i.
susceptibility	A matrix giving the susceptibility of individuals in demographic group i and risk group k.
p_susceptibility	
	A matrix giving the probability that an individual in demography group i is in risk (or susceptibility) group k. Each row represents the overall distribution of individuals in demographic group i across risk groups, and each row *must sum to 1.0*.
infectious_period	
	Duration of the infectious period in days. Default value is 1.8 days.

Value

Returns the R_0 for the infection in the population.

Examples

```
# Get example dataset and prepare contact matrix and demography
data(polymod_uk)
contact_matrix <- polymod_uk$contact_matrix
demography_vector <- polymod_uk$demography_vector

# define lambda
lambda <- 0.3

# define infectious period of 5 days
infectious_period <- 5
# define the number of age and susceptibility groups
n_demo_grps <- length(demography_vector)
n_risk_grps <- 3

# In this example, all risk groups from all age groups are fully
# susceptible
susceptibility <- matrix(
  data = 1, nrow = n_demo_grps, ncol = n_risk_grps
)

p_susceptibility <- matrix(
  data = 1, nrow = n_demo_grps, ncol = n_risk_grps
)
# p_susceptibility rows must sum to 1.0
p_susceptibility <- p_susceptibility / rowSums(p_susceptibility)

lambda_to_r0(
```

```
    lambda, contact_matrix, demography_vector,
    susceptibility, p_susceptibility,
    infectious_period
)
```

---

polymod_uk *Example POLYMOD social contact data for the U.K.*

Description

An example of social contact and demography data for use with `finalsize`, accessed from the POLYMOD social contacts dataset using the `socialmixr` package. Data are for the United Kingdom, and age limits are set at 0, 20, and 40 years, with `symmetric = TRUE`. Code to get these data is given in `data-raw/polymod_uk.R`.

Usage

```
polymod_uk
```

Format

`polymod_uk`:

A list with two named elements:

contact_matrix A contact matrix with mean contacts between age groups. This matrix is scaled by its largest real eigenvalue, and each row is scaled by the corresponding element in the `demography_vector`.

demography_vector A vector with the number of individuals in each of three age groups: 0 – 20, 20 – 40, 40+.

Source

[doi:10.1371/journal.pmed.0050074](); obtained using `socialmixr::polymod`. See further methods in `data-raw/polymod_uk.R`.

---

r0_to_lambda *Calculate transmission rate (λ) from R_0*

Description

Uses the R0 ($R0$), contact matrix (C), population (N), and infectious period (γ) to calculate the transmission rate using the following equation.

$$\lambda = R_0 / (Max(EV(C))\gamma)$$

where $EV(C)$ denotes the eigenvalues of the matrix C which is calculated from the social contacts matrix scaled by the number of individuals in each demographic and susceptibility group in the population.

Usage

```
r0_to_lambda(
  r0,
  contact_matrix,
  demography_vector,
  susceptibility,
  p_susceptibility,
  infectious_period = 1.8
)
```

Arguments

r0   The basic reproductive number R_0 of the infection.

contact_matrix   Social contact matrix. Entry m_{ij} gives average number of contacts in group i reported by participants in group j

demography_vector

                  Demography vector. Entry v_i gives proportion of total population in group i.

susceptibility   A matrix giving the susceptibility of individuals in demographic group i and risk group k.

p_susceptibility

                  A matrix giving the probability that an individual in demography group i is in risk (or susceptibility) group k. Each row represents the overall distribution of individuals in demographic group i across risk groups, and each row *must sum to 1.0*.

infectious_period

                  Duration of the infectious period in days. Default value is 1.8 days.

Value

Returns the transmission rate of the infection, also called the 'force of infection' (λ). This is different from the effective transmission rate (β).

Examples

```
# Get example dataset and prepare contact matrix and demography
data(polymod_uk)
contact_matrix <- polymod_uk$contact_matrix
demography_vector <- polymod_uk$demography_vector

# define R0 similar to pandemic influenza
r0 <- 1.5
# define infectious period of 5 days
infectious_period <- 5
# define the number of age and susceptibility groups
n_demo_grps <- length(demography_vector)
n_risk_grps <- 3

# In this example, all risk groups from all age groups are fully
# susceptible
```

```
susceptibility <- matrix(
  data = 1, nrow = n_demo_grps, ncol = n_risk_grps
)

p_susceptibility <- matrix(
  data = 1, nrow = n_demo_grps, ncol = n_risk_grps
)
# p_susceptibility rows must sum to 1.0
p_susceptibility <- p_susceptibility / rowSums(p_susceptibility)

r0_to_lambda(
  r0, contact_matrix, demography_vector,
  susceptibility, p_susceptibility,
  infectious_period
)
```

---

r_eff *Calculate R_eff in a heterogeneous population*

Description

r_eff calculates the effective reproductive number R_eff in a population with heterogeneous mixing, and with heterogeneous susceptibility to infection such as due to immunisation.

Usage

```
r_eff(r0, contact_matrix, demography_vector, susceptibility, p_susceptibility)
```

Arguments

r0   The basic reproductive number R_0 of the infection.

contact_matrix Social contact matrix. Entry m_{ij} gives average number of contacts in group i reported by participants in group j

demography_vector
                    Demography vector. Entry v_i gives proportion of total population in group i (model will normalise if needed).

susceptibility A matrix giving the susceptibility of individuals in demographic group i and risk group k.

p_susceptibility
                    A matrix giving the probability that an individual in demography group i is in risk (or susceptibility) group k. Each row represents the overall distribution of individuals in demographic group i across risk groups, and each row *must sum to 1.0*.

Value

A single number of the effective reproductive number of the infection in the population.

Examples

```
# load example POLYMOD data included in the package
data(polymod_uk)
r0 <- 2.0
contact_matrix <- polymod_uk$contact_matrix
demography_vector <- polymod_uk$demography_vector

# define the number of age and susceptibility groups
n_demo_grps <- length(demography_vector)
n_risk_grps <- 3

# In this example, all risk groups from all age groups are fully
# susceptible
susceptibility <- matrix(
  data = 1, nrow = n_demo_grps, ncol = n_risk_grps
)

p_susceptibility <- matrix(
  data = 1, nrow = n_demo_grps, ncol = n_risk_grps
)
# p_susceptibility rows must sum to 1.0
p_susceptibility <- p_susceptibility / rowSums(p_susceptibility)

# calculate R_effective
r_eff(
  r0 = r0,
  contact_matrix = contact_matrix,
  demography_vector = demography_vector,
  susceptibility = susceptibility,
  p_susceptibility = p_susceptibility
)
```

Index