

Package ‘finalsize’

April 9, 2024

Title Calculate the Final Size of an Epidemic

Version 0.2.1

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>](https://doi.org/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>

Depends R (>= 2.10)

Imports checkmate, Rcpp

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

LinkingTo Rcpp, RcppEigen

VignetteBuilder knitr

Config/Needs/website epiverse-trace/epiversetheme

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.2.3

NeedsCompilation yes

Author Pratik Gupte [aut, cre, cph] ([<https://orcid.org/0000-0001-5294-7819>](https://orcid.org/0000-0001-5294-7819)),
Edwin Van Leeuwen [aut, cph] ([<https://orcid.org/0000-0002-2383-5305>](https://orcid.org/0000-0002-2383-5305)),
Adam Kucharski [aut, cph] ([<https://orcid.org/0000-0001-8814-9421>](https://orcid.org/0000-0001-8814-9421)),
Rosalind Eggo [ctb] ([<https://orcid.org/0000-0002-0362-6717>](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 2024-04-09 16:40:06 UTC

R topics documented:

.final_size	2
final_size	2
polymod_uk	5
r0_conversions	6

Index **10**

.final_size	<i>Calculate the final size of an epidemic</i>
-------------	--

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	<i>Final size of an epidemic</i>
------------	----------------------------------

Description

`final_size` calculates the final size of an epidemic outbreak with a given R_0 , with options for specifying 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 = matrix(1),
  demography_vector = 1,
  susceptibility = matrix(1),
  p_susceptibility = matrix(1),
  solver = c("iterative", "newton"),
  control = list()
)
```

Arguments

<code>r0</code>	The basic reproductive number R_0 of the disease.
<code>contact_matrix</code>	Social contact matrix. Entry m_{ij} gives average number of contacts in group i reported by participants in group j . Defaults to the singleton matrix [1], representing a homogeneously mixing population.
<code>demography_vector</code>	Demography vector. Entry v_i gives proportion of total population in group i (model will normalise if needed). Defaults to 1, representing a population where demographic structure is not important (or not known), and where all individuals are assumed to belong to the same demographic group.
<code>susceptibility</code>	A matrix giving the susceptibility of individuals in demographic group i and risk group k . Defaults to the singleton matrix [1], representing a population where all individuals are fully susceptible to infection.
<code>p_susceptibility</code>	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 <i>must sum to 1.0</i> . Defaults to the singleton matrix [1], representing a population where all individuals are fully susceptible.
<code>solver</code>	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 <code>control</code> argument.
<code>control</code>	A list of named solver options, see <i>Solver options</i> .

Details**Specifying heterogeneous mixing and susceptibility:**

`final_size()` allows for heterogeneous population mixing and susceptibility, and this is described in the dedicated vignettes:

1. Heterogeneous population mixing: See vignette on "Modelling heterogeneous social contacts" (`vignette("varying_contacts", package = "finalsize")`);
2. Heterogeneous susceptibility to infection: See vignette on "Modelling heterogeneous susceptibility" (`vignette("varying_susceptibility", package = "finalsize")`).

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.

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*.

Examples

```
## For a given R_0
r0 <- 2.0
final_size(r0)

## For a population with multiple demographic groups
# load example POLYMOD data included in the package
data(polymod_uk)
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, and the final size in each group is influenced only by
# differences in social contacts
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
)

## Using manually specified solver 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
)
```

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

```
r0_conversions
```

```
Converts between epidemiological parameters related to R_0
```

Description

Converts between R_0 and the transmission rate λ , or calculates the effective reproduction number R_{eff} for a population, while accounting for population characteristics including demographic heterogeneity in social contacts, heterogeneity in the demographic distribution, and heterogeneity in susceptibility to infection.

Uses the R_0 (R_0), contact matrix (C), population (N), and infectious period (γ) to calculate the transmission rate using the following equation.

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

where $\text{EV}(C)$ 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
)

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

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

```

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 <i>must sum to 1.0</i> .
infectious_period	Duration of the infectious period in days. Default value is 1.8 days.
r0	The basic reproductive number R_0 of the infection.

Details

Given 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)

- `r_eff()` calculates the effective reproductive number;

- lamda_to_r0() calculates the R_0 from the transmission rate as

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

- r0_to_lambda() calculates the transmission rate as

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

Note that this is also called the 'force of infection' and is different from the effective transmission rate often denoted β .

Here, $EV(C)$ 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.

Value

Returns a single number for the calculated value.

Examples

```
#### Prepare data ####
# 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, risk varies across groups
susceptibility <- matrix(
  data = c(0.5, 0.7, 1.0), nrow = n_demo_grps, ncol = n_risk_grps
)

# risk does not vary within groups
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)

#### Effective R ####
r0 <- 2.0
r_eff(
  r0 = r0,
  contact_matrix = contact_matrix,
```



```
demography_vector = demography_vector,  
susceptibility = susceptibility,  
p_susceptibility = p_susceptibility  
)  
  
#### Transmission rate to R0 ####  
lambda_to_r0(  
  lambda, contact_matrix, demography_vector,  
  susceptibility, p_susceptibility,  
  infectious_period  
)  
  
#### R0 to Transmission rate ####  
r0 <- 1.5  
r0_to_lambda(  
  r0, contact_matrix, demography_vector,  
  susceptibility, p_susceptibility,  
  infectious_period  
)
```

Index

- * **datasets**
 - polymod_uk, 5
- * **epidemic**
 - final_size, 2
- * **model**
 - final_size, 2
 - .final_size, 2
- final_size, 2
- lambda_to_r0(r0_conversions), 6
- polymod_uk, 5
- r0_conversions, 6
- r0_to_lambda(r0_conversions), 6
- r_eff(r0_conversions), 6