Package ‘EpiDynamics’

August 29, 2016

Type Package

Title Dynamic Models in Epidemiology

Depends R (>= 3.2.2)

Imports deSolve, reshape2, ggplot2, grid

Description Mathematical models of infectious diseases in humans and animals. Both, deterministic and stochastic models can be simulated and plotted.

License GPL (>= 2)

LazyLoad yes

URL https://github.com/oswaldosantos/EpiDynamics

Version 0.3.0

Date 2015-12-03

RoxygenNote 5.0.1

NeedsCompilation no

Author Oswaldo Santos Baquero [aut, cre], Fernando Silveira Marques [aut]

Maintainer Oswaldo Santos Baquero <oswaldosant@gmail.com>

Repository CRAN

Date/Publication 2015-12-03 23:28:14

R topics documented:

EpiDynamics-package .................................................. 2
MultiStrainPartialImmunity ........................................ 3
PlotMods ................................................................. 4
SEIR ................................................................. 5
SEIR4AgeClasses ...................................................... 6
SEIRnStages .......................................................... 8
SIR ............................................................... 9
sir2AgeClasses ...................................................... 11
SIR2Stages .......................................................... 12
EpiDynamics-package

Description

Mathematical models of infectious diseases in humans and animals. Both, deterministic and stochastic models can be simulated and plotted.

Details

Package: EpiDynamics
Type: Package
Version: 0.3.0
Date: 2015-12-03
Depends: R (>= 3.2.2)
Imports: deSolve, reshape2, ggplot2, grid
License: GPL (>= 2)
LazyLoad: yes
URL: http://oswaldosantos.github.io/EpiDynamics
Authors: Oswaldo Santos Baquero <oswaldosant@gmail.com>
          Fernando Silveira Marques <fernandosix@gmail.com>
Maintainer: Oswaldo Santos Baquero <oswaldosant@gmail.com>
MultiStrainPartialImmunity

Partial immunity model that cycles (P 4.2).

Description

Solves multi-strain where the strains are arranged in a circle and each strain offers partial immunity (in terms of reduced transmission) to its neighbours.

Usage

MultiStrainPartialImmunity(pars = NULL, init = NULL, time = NULL, ...)

Arguments

pars vector with 2 vectors and 2 values in the following order: beta, gamma, mu and a. The first element of each vector corresponds to the first strain, the second element to the second strain and so on. mu is the per capita death rate and alpha is the modified transmission rate due to partial immunity.

init vector with 3 vectors in the following order: S, P and L. The first element of each vector corresponds to the first strain, the second element to the second strain and so on. Si + Pi + Li = 1 but sum(Si) could be greater than 1.

time time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 4.2 from page 123 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first element, *$model, is the model function. The second, third and fourth elements are the vectors (*$pars, *$init, *$time, containing the pars, init and time arguments of the function. The fifth element *$results is a data.frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the proportion of susceptibles, infectious and recovered.

References

See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(beta = rep(40, 4), gamma = rep(9.98, 4),
               mu = 0.02, a = 0.4)

initials <- c(S = c(0.08, 0.1, 0.1, 0.1),
               P = c(0.4, 0.3, 0.3, 0.29),
               L = c(0.15, 0.02, 0.1, 0.01))

# Solve and plot.
multi.strain.pi <- MultiStrainPartialImmunity(pars = parameters,
                                              init = initials,
                                              time = 0:200)
PlotMods(multi.strain.pi, variables = c('L1', 'L2', 'L3', 'L4'), grid = FALSE)

PlotMods

Plot results of capm model functions

Description

Plot results of EpiDynamics' functions.

Usage

PlotMods(model.out = NULL, variables = NULL, x.label = NULL,
           y.label = NULL, legend.title = "variable", line.size = 1,
           text.size = 14, grid = TRUE, bifur = FALSE)

Arguments

model.out output of aEpiDynamics' function.
variables column index for the variables in model.out to be plotted.
x.label string with the name of x axis.
y.label string with the name of y axis.
legend.title string with the legend title.
line.size scalar to define the thick of the lines (points for bifurcations) to be plotted.
text.size scalar to define the size of axis texts and titles.
grid logical to indicate if each variable must be plotted in a separated panel.
bifur logical to indicate if model.out represent a bifurcation.
Examples

# Parameters and initial conditions.
parameters <- list(beta0 = 17 / 13, beta1 = 0.1, gamma = 1 / 13,
                  omega = 2 * pi / 365, mu = 1 / (50 * 365))

initials <- c(S = 1 / 17, I = 1e-4,
              R = 1 - 1 / 17 - 1e-4)

# Solve the system.
sir.sinusoidal.forcing <- SIRSinusoidalForcing(pars = parameters,
                                              init = initials,
                                              time = 0:(60 * 365))
PlotMods(sir.sinusoidal.forcing)

# Solve bifurcation dynamics for 20 years.
# If max(time) < 3650, bifurcation dynamics are solved for 3650 time-steps.
parameters2 <- list(beta0 = 17 / 13, beta1 = seq(0.001, 0.251, by = 0.001),
                     gamma = 1 / 13, omega = 2 * pi / 365, mu = 1 / (50 * 365))

# Uncomment the following lines:
# bifur <- SIRSinusoidalForcing(pars = parameters2,
#                               init = initials,
#                               time = 0:(20 * 365))
# PlotMods(bifur, bifur = TRUE)

---

SEIR

SEIR model (2.6).

Description

Solves a SEIR model with equal births and deaths.

Usage

SEIR(pars = NULL, init = NULL, time = NULL, ...)  

Arguments

pars  vector with 4 values: the per capita death rate (and the population level birth rate), the transmission rate, the movement form exposed to infectious and the recovery rate. The names of these values must be "mu", "beta", "sigma" and "gamma", respectively.

init  vector with 3 values: the initial proportion of proportion of susceptibles, exposed, infectious and recovered. The names of these values must be "S", "E", "I" and "R", respectively.

time  time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.
All parameters must be positive and S + E + I + R <= 1.
Details

This is the R version of program 2.6 from page 41 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first element, *$model*, is the model function. The second, third and fourth elements are the vectors (*$pars*, *$init*, *$time*, containing the pars, init and time arguments of the function. The fifth element *$results* is a data.frame with up to as many rows as elements in time. First column contains the time. Second to fifth column contain the proportion of susceptibles, exposed, infectious and recovered.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(mu = 1 / (70 * 365), beta = 520 / 365,
        sigma = 1 / 14, gamma = 1 / 7)
initials <- c(S = 0.1, E = 1e-04, I = 1e-04, R = 1 - 0.1 - 1e-4 - 1e-4)

# Solve and plot.
seir <- SEIR(pars = parameters, init = initials, time = 0:(60 * 365))
PlotMods(seir)

SEIR4AgeClasses

SEIR model with 4 age classes and yearly aging (P 3.4).

Description

Solves a SEIR model with four different age-groups and yearly "movements" between the groups mimicking the school year.

Usage

SEIR4AgeClasses(pars = NULL, init = NULL, time = NULL, ...)

SEIR4AgeClasses

SEIR model with 4 age classes and yearly aging (P 3.4).
Arguments

pars list with: a matrix with the transmission rates, the rate at which individuals move from the exposed to the infectious classes, the recovery rate, a vector with death rates in each age group, and a vector with birth rates into the childhood class. The names of these elements must be "beta", "sigma", "gamma", "mu", and "nu", respectively, see example.

init vector with 16 values: initial proportions of the population that are susceptible, exposed, infectious and recovered in a particular age group. The vector must be named, see example. Requirements: S + E + I <= n for each age group and all values must be positive.

time time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 3.4 from page 87 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

All rates are specified in days. Moreover, a vector n with the proportion of each age group. All parameters must be positive.

Value

list of class SolveSEIR4ACYA. The first element, *$model, is the model function. The second, third and fourth elements are vectors (pars, init, time, respectively) containing the pars, init and time arguments of the function. The fifth element *$results is a data.frame with up to as many rows as elements in time. First column contains the time. The following columns contain the proportion of susceptibles, exposed, infectious and recovered.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- list(beta = matrix(c(2.089, 2.089, 2.089, 2.037,
                                  2.089, 9.336, 2.086, 2.037,
                                  2.086, 2.086, 2.086, 2.037,
                                  2.037, 2.037, 2.037, 2.037),
                                 nrow = 4, ncol = 4),
  sigma = 0.125, gamma = 0.2,
  mu = c(0, 0, 0, 1) / (55 * 365),
  nu = c(1 / (55 * 365), 0, 0, 0),
\[ n = \left( 6, 4, 10, 55 \right) / 75 \]

\[
\text{initials} \leftarrow c(S = c(0.05, 0.01, 0.01, 0.008),
E = c(0.0001, 0.0001, 0.0001, 0.0001),
I = c(0.0001, 0.0001, 0.0001, 0.0001),
R = c(0.0298, 0.04313333, 0.12313333, 0.72513333))
\]

# Solve and plot.
# Uncomment the following lines (running it takes more than a few seconds):
# seir.age.classes <- SEIR4AgeClasses(pars = parameters,
# init = initials,
# time = 0:36500)
# PlotMods(seir.age.classes,
# variables = c('I1', 'I2', 'I3', 'I4'), grid = F)

### SEIRnStages

**SEIR model with n stages (P 3.5).**

**Description**

Solves a SEIR model with multiple stages to create gamma-distributed exposed and infectious periods.

**Usage**

SEIRnStages(pars = NULL, init = NULL, time = NULL, ...)

**Arguments**

pars  
vector with 5 values: the transmission rate, the removal or recovery rate, the death rate (we assume that nu = mu), the number of stages in the infected period and the number of stages in the exposed period. The names of these elements must be "beta", "gamma", "mu", "n" and "m", respectively, see example. All rates are specified in days and all rates and parameters must be positive, moreover, m < n.

init  
vector with n + 1 values: initial proportions of the population that are susceptible and infected. The vector must be named, see example. Requirements: S + all Infected <= 1.

time  
time sequence for which output is wanted; the first value of times must be the initial time.

...  
further arguments passed to ode function.

**Details**

This is the R version of program 3.5 from page 94 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.
Value

list. The first element, *$model*, is the model function. The second, third and fourth elements are vectors (*$pars*, *$init*, *$time*, respectively) containing the pars, init and time arguments of the function. The fifth element *$results* is a data.frame with up to as many rows as elements in time. First column contains the time. The following columns contain the proportion of susceptibles and infected.

References


See Also

ode.

Examples

# Parameters and initial conditions.
N <- 13
parameters <- list(bound = 0.1 / 5, gamma = 1 / 18, mu = 1 / 55 * 365,
    n = n, m = 8)

initials <- c(S = 0.1, I = 0.00001 * rep(1, n) / n)

# Solve and plot.
# Uncomment the following lines (running it takes more than a few seconds):
# seir.n.stages <- SEIRnStages(pars = parameters,
#     init = initials,
#     time = seq(1, 30 * 365, 1))
# PlotMods(seir.n.stages, variables = 2)
# PlotMods(seir.n.stages, variables = 3:13, grid = F)

---

SIR

Simple SIR model (P 2.1).

Description

Solves a simple SIR model without births or deaths.

Usage

SIR(pars = NULL, init = NULL, time = NULL, ...)

---
Arguments

pars vector with 2 values: the transmission and recovery rates. The names of these values must be "beta", and "gamma", respectively.

init vector with 3 values: the initial proportion of susceptibles, infectious and recovered. The names of these values must be "S", "I" and "R", respectively.

time time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 2.1 from page 19 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

All parameters must be positive and S + I + R <= 1.

Value

list. The first element, *$model, is the model function. The second, third and fourth elements are the vectors *$pars, *$init and *$time, containing the pars, init and time arguments of the function. The fifth element *$results is a data.frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the proportion of susceptibles, infectious and recovered.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(beta = 1.4247, gamma = 0.14286)
initials <- c(S = 1 - 1e-06, I = 1e-06, R = 1 - (1 - 1e-06 - 1e-06))

# Solve and plot.
sir <- SIR(pars = parameters, init = initials, time = 0:70)
PlotMods(sir)
**sir2AgeClasses**

*SIR model with 2 age classes (P 3.3).*

**Description**

Solves a SIR model two different age-groups.

**Usage**

```r
sir2AgeClasses(pars = NULL, init = NULL, time = NULL, ...)
```

**Arguments**

- **pars** `vector` with 8 values: 4 transmission rates, 1 recovery rate, rate at which children mature, death rate in the childhood group and death rate in the adult group. The names of these values must be "betaCC","betaCA","betaAC", "betaAA", "gamma", "IC","muC" and "muA", respectively. The letters after the word "beta" denote transmission to any group from any group, e.g., "betaCA" represent transmission to children group from adult group. All parameters must be positive. Parameters "nC" na "nu" (proportion of the population that are in the childhood group and birth rate into the childhood class, respectively) are not defined explicitly, but calculated as: nC = muA/(IC+muA) and nu = (IC+nuA)nC. All rates are specified in years and all parameters must be positive.

- **init** `vector` with 4 values: the initial proportion of the population that are both susceptible and in the childhood group, the initial proportion of the population that are both infectious and in the childhood group, the initial proportion of the population that are both susceptible and in the adult group, and the initial proportion of the population that are both infectious and in the adult group. The names of these values must be "SC", "IC", "SA" and "IA", respectively. Requirements: SC + IC <= nC, and SA + IA <= nA = 1 - nC.

- **time** time sequence for which output is wanted; the first value of times must be the initial time.

- **...** further arguments passed to `ode` function.

**Details**

This is the R version of program 3.3 from page 79 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

**Value**

`list`. The first element, `$model`, is the model function. The second, third and fourth elements are the vectors `$pars`, `$init`, `$time`, containing the pars, init and time arguments of the function. The fifth element `$results` is a `data.frame` with up to as many rows as elements in time. First column contains the time. The following columns contain the proportion of susceptibles, infectious and recovered.
References


See Also

ode.

Examples

```
# Parameters and initial conditions.
parameters <- c(betaCC = 100, betaCA = 10, betaAC = 10, betaAA = 20,
gamma = 10, lC = 0.0066667, muC = 0.0, mua = 0.016667)
initials <- c(SC = 0.1, IC = 0.00001, SA = 0.1, IA = 0.0001)

# Solve and plot.
sir2AgeClasses <- sir2AgeClasses(pars = parameters, 
    init = initials, time = seq(0, 100, 0.01))
PlotMods(sir2AgeClasses, variables = c('IA', 'IC'), grid = FALSE)
```

---

**SIR2Stages**

*SIR model with 2 age classes (P.3.3).*

Description

Solves a SIR model two different age-groups.

Usage

```
SIR2Stages(pars = NULL, init = NULL, time = NULL, ...)
```

Arguments

- **pars** vector with 8 values: 4 transmission rates, 1 recovery rate, rate at which children mature, death rate in the childhood group and death rate in the adult group. The names of these values must be "betaCC","betaCA","betaAC", "betaAA", "gamma", "lC","muC" and "mua", respectively. The letters after the word "beta" denote transmission to any group from any group, e.g., "betaCA" represent transmission to children group from adult group. All parameters must be positive. Parameters "nC" na "nu" (proportion of the population that are in the childhood group and birth rate into the childhood class, respectively) are not defined explicitly but calculated as: nC = muA/(lC+muA) and nu = (lC+nuA)nC. All rates are specified in years and all parameters must be positive.

- **init** vector with 4 values: the initial proportion of the population that are both susceptible and in the childhood group, the initial proportion of the population that are both infectious and in the childhood group, the initial proportion of the population that are both infectious and in the childhood group, the initial proportion of the population that are both susceptible and in the adult group, and the initial proportion...
SIR2TypesImports

of the population that are both infectious and in the adult group. The names of these values must be "SC", "IC", "SA" and "IA", respectively. Requirements: SC + IC <= nC, and SA + IA <= nA = 1 - nC.

time time sequence for which output is wanted; the first value of times must be the initial time.
... further arguments passed to ode function.

Details

This is the R version of program 3.3 from page 79 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first element, *$model, is the model function. The second, third and fourth elements are the vectors (*$pars, *$init, *$time, containing the pars, init and time arguments of the function. The fifth element *$results is a data.frame with up to as many rows as elements in time. First column contains the time. The following columns contain the proportion of susceptibles, infectious and recovered.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(betaCC = 100, betaCA = 10, betaAC = 10, betaAA = 20,
gamma = 10, lc = 0.06666667, muC = 0.0, muA = 0.0166667)
initials <- c(SC = 0.1, IC = 0.0001, SA = 0.1, IA = 0.0001)

# Solve the system.
sir2stages <- SIR2Stages(pars = parameters,
init = initials, time = seq(0, 100, 0.01))

SIR2TypesImports SIR model with two types of imports (P 6.6).

Description

Solves a model with two types of stochastic imports and demographic stochasticit
Usage

SIR2TypesImports(pars = NULL, init = NULL, time = NULL, ...)

Arguments

pars vector with 5 parameters: transmission rate, recovery rate and per capita death rate. The names of these values must be "beta", "gamma", "mu", "epsilon" and "delta" respectively. All parameters must be positive. The birth rate is assumed to be constant and equal to mu * N, therefore preventing extinction of the host population.

init vector with 3 values: the initial population size that are susceptible, infectious and the total population size. The names of these values must be "X", "Y" and "N", respectively. All initial conditions must be positive and all refer to integer numbers.

time time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 6.6 from page 210 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first, second and third elements are the vectors ($pars, $init, $time, containing the pars, init and time arguments of the function. The fourth element $results is a data.frame with up to as many rows as elements in time. First column contains the time. The following columns contain the number of susceptibles, infectious, recovered and boolean for epsilon and delta imports.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(beta = 1, gamma = 0.1, mu = 5e-4,
              epsilon = 2e-5, delta = 0.01)
initials <- c(X = 5, Y = 2, N = 50)

# Solve and plot.
sir.2types.imports <- SIR2TypesImports(parameters, initials, 2 * 365)
PlotMods(sir.2types.imports)
SIR Additive Noise

SIR model with constant additive noise (P 6.1).

Description
Solves a SIR model with constant additive noise added to the transmission rate.

Usage
SIRAdditiveNoise(pars = NULL, init = NULL, time = NULL, step = 1, ...)

Arguments
pars vector with 5 values: the transmission rate, the recovery rate, the birth (death) rate, the amount of noise experienced in the transmission rate and the population size assumed to be constant. The names of these values must be "beta", "gamma", "mu", "noise", and "N" respectively.
init vector with 2 values: the initial number of susceptibles and infectious. The names of these values must be "X", and "Y", respectively. "X" and "Y" must be positive and are numbers not proportions.
time time sequence for which output is wanted; the first value of times must be the initial time.
step step size to set the integration step and to scale the noise term.
... further arguments passed to ode function.

Details
This is the R version of program 6.1 from page 194 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value
list. The first element, $model, is the model function. The second, third and fourth elements are the vectors $pars, $init and $time, containing the pars, init and time arguments of the function. The fifth element $results is a data.frame with up to as many rows as elements in time. First column contains the time. Second and third columns contain the number of susceptibles and recovered.

References

See Also
ode.
Examples

# Parameters and initial conditions.
parameters <- c(beta = 1, gamma = 1 / 10, mu = 1 / (50 * 365),
noise = 10, N = 1e6)
initials <- c(X = 1e5, Y = 500)

# Solve and plot.
sir.additive.noise <- SIRAdditiveNoise(pars = parameters, init = initials,
                                   time = 0:(2 * 365), step = 1)
PlotMods(sir.additive.noise)

SIRBirthDeath  SIR model with births and deaths (P 2.2).

Description
Solves a simple SIR model with equal births and deaths.

Usage
SIRBirthDeath(pars = NULL, init = NULL, time = NULL, ...)

Arguments
pars  vector with 3 values: the per capita death rate (equal to the population level
      birth rate), the transmission rate, and the recovery rate. The names of these
      values must be "mu", "beta", and "gamma", respectively.
init  vector with 3 values: the initial proportion of proportion of susceptibles, in-
     fectious and recovered. The names of these values must be "S", "I" and "R",
      respectively.
time time sequence for which output is wanted; the first value of times must be the
      initial time.
...  further arguments passed to ode function.

Details
This is the R version of program 2.2 from page 27 of "Modeling Infectious Disease in humans and
animals" by Keeling & Rohani.
All parameters must be positive and S + I + R <= 1.

Value
list. The first element, *$model, is the model function. The second, third and fourth elements
are the vectors *$pars, *$init, *$time, containing the pars, init and time arguments of the
function. The fifth element *$results is a data.frame with up to as many rows as elements in
time. First column contains the time. Second, third and fourth columns contain the proportion of
susceptibles, infectious and recovered.
References

Keeling, Matt J., and Pejman Rohani. Modeling infectious diseases in humans and animals. Prince-
ton University Press, 2008. Modeling Infectious Diseases in Humans and Animals

See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(mu = 1 / (70 * 365),
              beta = 520 / 365, gamma = 1 / 7)
initials <- c(S = 0.1, I = 1e-4, R = 1 - 0.1 - 1e-4)

# Solve and plot.
sir.birth.death <- SIRBirthDeath(pars = parameters, init = initials,
                                  time = 0:(60 * 365))
PlotMods(sir.birth.death)

SIRCarrierState

SIR model with carrier state (2.7).

Description

Solves a SIR model with carrier state.

Usage

SIRCarrierState(pars = NULL, init = NULL, time = NULL, ...)

Arguments

pars vector with 6 values: the per capita death, transmission, infectious-recovery and carrier-recovery rates, the proportion of reduction in transmission from carriers compared with infectious and the proportion of infectious that become carriers. The names of these values must be "mu", beta", "gamma", "Gamma", "epsilon" and "rho", respectively.

init vector with 4 values: the initial proportion of proportion of susceptibles, infectious, carriers and recovered. The names of these values must be "S", "I", "C" and "R", respectively.

time time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.
Details

This is the R version of program 2.7 from page 44 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

All parameters must be positive and \( S + I + C + R \leq 1 \).

Value

- **list**. The first element, \(*Dmodel\), is the model function. The second, third and fourth elements are the vectors \(*Dpars, *Dinit, *Dtime\), containing the pars, init and time arguments of the function. The fifth element \(*Dresults\) is a data.frame with up to as many rows as elements in time. First column contains the time. Second to fifth column contain the proportion of susceptibles, infectious, cariers and recovered.

References


See Also

- ode.

Examples

```r
# Parameters and initial conditions.
parameters <- c(mu = 1 / (50 * 365), beta = 0.2,
                gamma = 0.1, Gamma = 0.001,
                epsilon = 0.1, rho = 0.4)
initials <- c(S = 0.1, I = 1e-4, C = 1e-3, R = 1 - 0.1 - 1e-4 - 1e-3)

# Solve the system.
sir.carrier.state <- SIRCarrierState(pars = parameters,
                                      init = initials, time = 0:60)
PlotMods(sir.carrier.state)
```

SIRDemogStoch

- **SIR model with demographic stochasticity (P 6.4).**

Description

Solves a SIR model with demographic stochasticity

Usage

- `SIRDemogStoch(pars = NULL, init = NULL, time = NULL, ...)"
**Arguments**

`pars` vector with 3 parameters: transmission rate, recovery rate and per capita death rate. The names of these values must be "beta", "gamma" and "mu", respectively. All parameters must be positive and all rates are specified in days. The birth rate is assumed to be constant and equal to mu * N, therefore preventing extinction of the host population.

`init` vector with 3 values: the initial population size that are susceptible, infectious and the total population size. The names of these values must be "X", "Y" and "N", respectively. All initial conditions must be positive.

`time` time sequence for which output is wanted; the first value of times must be the initial time.

`...` further arguments passed to `ode` function.

**Details**

This is the R version of program 6.4 from page 203 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

**Value**

`list`. The first, second and third elements are the vectors `*pars, *init, *time`, containing the `pars, init` and `time` arguments of the function. The fourth element `*results` is a `data.frame` with up to as many rows as elements in `time`. First column contains the time. The following columns contain the proportion of susceptibles, infectious and recovered.

**References**


**See Also**

`ode`.

**Examples**

# Parameters and initial conditions.
parameters <- c(beta = 1, gamma = 1 / 10, mu = 5e-4)
initials <- c(X = 500, Y = 25, N = 5e3)

# Solve and plot.
sir.demog.stoch <- SIRDemogStoch(pars = parameters,
                                 init = initials, time = 2 * 365)
PlotMods(sir.demog.stoch)
SIRInducedMortality  

*SIR model with disease induced mortality: Density-dependent transmission (P 2.3).*

**Description**

Solves a SIR model with a probability of mortality, and unequal births and deaths.

**Usage**

`sirinducedmortality(pars = NULL, init = NULL, time = NULL, ...)`

**Arguments**

- `pars` vector with 5 values: the probability that an infected individual dies from the disease before recovering, the per capita death rate from natural causes, the population level birth rate, the transmission rate, and the recovery rate. The name of these values must be: "rho", "mu", "nu", "beta", and "gamma", respectively. All parameters must be positive.
- `init` vector with 3 values: the initial number of susceptibles, infectious and recovered. The names of these values must be "X", "Y" and "Z", respectively.
- `time` time sequence for which output is wanted; the first value of times must be the initial time.
- `...` further arguments passed to `ode` function.

**Details**

This is the R version of program 2.3 from page 35 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

**Value**

`list`. The first element, `$model`, is the model function. The second, third and fourth elements are the vectors `$pars`, `$init` and `$time`, containing the pars, init and time arguments of the function. The fifth element `$results` is a `data.frame` with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the number of susceptibles, infectious and recovered.

**References**


**See Also**

`ode`. 

SIRInducedMortality
Examples

```r
# Parameters and initial conditions.
parameters <- c(rho = 0.5, mu = 1 / (70 * 365), nu = 1 / (70 * 365),
               beta = 520 / 365.0, gamma = 1 / 7)
initials <- c(X = 0.2, Y = 1e-4, Z = 0)

# Solve and plot.
# Uncomment the following lines (running it takes more than a few seconds):
# sir.induced.mortality <- SIRInducedMortality(pars = parameters,
#                                              init = initials,
#                                              time = 0:1e5)
# PlotMods(sir.induced.mortality)
```

Description

Solves a SIR model with a probability of mortality, and unequal births and deaths

Usage

```r
SIRInducedMortality2(pars = NULL, init = NULL, time = NULL, ...)
```

Arguments

- **pars** vector with 5 values: the probability than an infected individual dies from the disease before recovering, the per capita death rate from natural causes, the population level birth rate, the transmission rate, and the recovery rate. The names of these values must be "rho", "mu", "nu", "beta", and "gamma", respectively. All parameters must be positive.

- **init** vector with 3 values: the initial number of susceptibles, infectious and recovered. The names of these values must be "X", "Y" and "Z", respectively.

- **time** time sequence for which output is wanted; the first value of times must be the initial time.

- **...** further arguments passed to ode function.

Details

This is the R version of program 2.4 from page 36 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.
**Value**

A list. The first element, `*Dmodel`, is the model function. The second, third and fourth elements are the vectors `*Dpars`, `*Dinit`, `*Dtime`, containing the `pars`, `init` and `time` arguments of the function. The fifth element `*Dresults` is a data frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the number of susceptibles, infectious and recovered.

**References**


**See Also**

`ode`.

**Examples**

```r
# Parameters and initial conditions.
parameters <- c(rho = 0.5, mu = 1 / (70 * 365.0), nu = 1 / (70 * 365.0),
               beta = 520 / 365.0, gamma = 1 / 7)
initials <- c(X = 0.2, Y = 1e-4, Z = 0)

# Solve and plot.
sir.induced.mortality2 <- SIRInducedMortality2(pars = parameters,
                                              init = initials,
                                              time = 0:1e4)
PlotMods(sir.induced.mortality2)
```

---

**SIRPartialImmunity**

*SIR model with partial immunity (P 4.1).*

**Description**

Solves a model with partial immunity.

**Usage**

`SIRPartialImmunity(pars = NULL, init = NULL, time = NULL, ...)`

**Arguments**

- `pars` vector with 9 values: the death, birth, transmission and recovery rates, the modified susceptibility to strain i for those individuals recovered from the other strain and the modified transmission rate of strain i from those individuals that have recovered from the other strain. The name of these values must be "mu", "v", "beta1", "beta2", "gamma1", "gamma2", "alpha1", "alpha2", "a1", "a2". The numbers 1 and 2 at the end of parameters names stand for strain 1 and strain 2.
init

vector with 8 values: In this formulation NAB refers to the proportion of the population who are A with respect to strain 1 and B with respect to strain 2. Thus, initial values must be named: "NSS", "NIS", "NRS", "NRI", "NSI", "NSR", "NIR" and "NRR". The sum of initial values must be <= 1.

time
time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 4.1 from page 118 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first element, *$model*, is the model function. The second, third and fourth elements are the vectors (*$pars*, *$init*, *$time*, containing the pars, init and time arguments of the function. The fifth element *$results* is a *data.frame* with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the proportion of susceptibles, infectious and recovered.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(mu = 1 / (70 * 365), beta1 = 260 / 365,
                beta2 = 520 / 365, gamma1 = 1 / 7,
                gamma2 = 1 / 7, alpha1 = 0.5,
                alpha2 = 0.4, a1 = 0.4, a2 = 0.5)

initials <- c(NSS = 0.1, NIS = 1e-4, NRS = 0.02, NRI = 0,
              NSI = 1e-4, NSR = 0.5, NIR = 0, NRR = 0.3798)

# Solve and plot.
sir.partial.immunity <- SIRPartialImmunity(pars = parameters,
                                           init = initials,
                                           time = 0:(100 * 365))
PlotMods(sir.partial.immunity, variables = c('NIS', 'NIR'), grid = FALSE)
SIRScaledAdditiveNoise

SIR model with Scaled additive noise (P 6.2).

Description

Solves a SIR model with scaled additive noise.

Usage

SIRScaledAdditiveNoise(pars = NULL, init = NULL, time = NULL, ...)

Arguments

pars  vector with 5 parameters: transmission rate, recovery rate, per capita death rate, the total population size and the number of steps that will change noise term. The names of these values must be "beta", "gamma", "mu", "N" and "step", respectively. All parameters must be positive and all rates are specified in days. The birth rate is assumed to be constant and equal to mu * N, therefore preventing extinction of the host population. Noise terms are generated as a function of the time step and its magnitude is a function of the rate of each process.

init  vector with 2 values: the initial population size that are susceptible and infectious. The names of these values must be "X" and "Y", respectively. All initial conditions must be positive.

time  time sequence for which output is wanted; the first value of times must be the initial time.

...  further arguments passed to ode function.

Details

This is the R version of program 6.2 from page 197 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first element, *$model, is the model function. The second, third and fourth elements are the vectors *$pars, *$init, *$time, containing the pars, init and time arguments of the function. The fifth element *$results is a data.frame with up to as many rows as elements in time. First column contains the time. The following columns contain the proportion of susceptibles, infectious and recovered.

References

SIR model with sinusoidal births (P 5.3).

Description

Solves a SIR model with sinusoidal forcing of the birth rate.

Usage

SIRsinusoidalBirth(pars = NULL, init = NULL, time = NULL, ...)
Value

list. The first element, *$model*, is the model function. The second element is a list with the the *$pars* argument. The third and fourth elements are the vectors (*$init*, *$time*, containing the init and time arguments of the function. The fifth element *$results* is a data.frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the proportion of susceptibles, infectious and recovered.

References

Keeling, Matt J., and Pejman Rohani. Modeling infectious diseases in humans and animals. Princeton University Press, 2008. Modeling Infectious Diseases in Humans and Animals. It is important to note that we wrote equations for the R population based on equations in the website and because of the dynamic S + I + R fluctuates around 1. Then, using R = 1 - S - I solves this inconsistency.

See Also

ode.

Examples

```r
# Parameters and initial conditions (bifurcation plot of infectious)
parameters <- list(beta = 17 / 13, alpha0 = 1 / (50 * 365),
                   alpha1 = 0.25, w = 2 * pi / 365,
                   mu = 1 / (50 * 365), gamma = 1 / 13)

parameters2 <- list(beta = 17 / 13, alpha0 = 1 / (50 * 365),
                      alpha1 = seq(0, 0.99, 0.01), w = 2 * pi / 365,
                      mu = 1 / (50 * 365), gamma = 1 / 13)

initials <- c(S = 1 / 17, I = 1e-4, R = 1 - (1 / 17 + 1e-4))

# Solve and plot.
sir.sinusoidal.birth <- SIRSinusoidalBirth(pars = parameters,
                                           init = initials,
                                           time = 0:(20 * 365))

PlotMods(sir.sinusoidal.birth)

# Bifurcations
# Uncomment the following lines (running it takes more than a few seconds):
# bifurcation <- SIRSinusoidalBirth(pars = parameters2,
#                                    init = initials,
#                                    time = 0:(20 * 365))
# PlotMods(bifur, bifur = TRUE)
```
SIR model with sinusoidal forcing (P 5.1).

Description

Solves a SIR model with sinusoidal forcing of the transmission rate.

Usage

SIRSinusoidalForcing(pars = NULL, init = NULL, time = NULL, ...)

Arguments

pars list with 4 values: the death rate, the mean transmission rate, a scalar (or a vector to create bifurcations) with the amplitude of sinusoidal forcing, the frequency of oscillations and the recovery rate. The names for these values must be: "mu", "beta0", "beta1", "omega" and "gamma", respectively.

init vector with 3 values: initial proportion of proportion of susceptibles, infectious and recovered. The names of these values must be "S", "I" and "R", respectively. All parameters must be positive and S + I <= 1.

time time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 5.1 from page 160 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

When beta1 is a vector in pars, it must be a sequence between 0 and 1.

Value

list. The first element, $model, is the model function. The second, third and fourth elements are the vectors ($pars, $init, $time, containing the pars, init and time arguments of the function. The fifth element $results is a data.frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the proportion of susceptibles, infectious and recovered.

References


See Also

ode.
Examples

# Parameters and initial conditions.
parameters <- list(beta0 = 17 / 13, beta1 = 0.1, gamma = 1 / 13,
                  omega = 2 * pi / 365, mu = 1 / (50 * 365))

initials <- c(S = 1 / 17, I = 1e-4,
              R = 1 - 1 / 17 - 1e-4)

# Solve and plot.
sir.sinusoidal.forcing <- SIRSinusoidalForcing(pars = parameters,
                                               init = initials,
                                               time = 0:(60 * 365))

PlotMods(sir.sinusoidal.forcing)

# Solve bifurcation dynamics for 20 years.
# If max(time) < 3650, bifurcation dynamics are solved for 3650 time-steps.
parameters2 <- list(beta0 = 17 / 13, beta1 = seq(0.001, 0.251, by = 0.001),
                    gamma = 1 / 13, omega = 2 * pi / 365, mu = 1 / (50 * 365))
# Uncomment the following lines (running it takes more than a few seconds):
# bifur <- SIRSinusoidalForcing(pars = parameters2,
#                              init = initials,
#                              time = 0:(20 * 365))
# PlotMods(bifur, bifur = TRUE)

SIR model with tau leap method (P 6.5).

Description

SIR model with demographic stochasticity approximated using the tau-leap method and assuming Poisson distributions.

Usage

SIR TauLeap(pars, init, endNtime)

Arguments

pars vector with 5 values: the transmission, recovery and death rates, the population size assumed to be constant and the time step. The names of these values must be "beta", "gamma", "mu", "N" and "tau" respectively.

init vector with 3 values: the initial number of susceptibles, infectious and recovered, respectively. The names of these values must be "X", "Y" and "Z" respectively.

end.time end time to be simulated.
**Details**

This is the R version of program 6.5 from page 204 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

**Value**

*list*. The first three elements are the vectors *$pars$, *$init$ and *$time$, containing the *pars*, *init* and *end.time* arguments of the function. The fourth element *$results$ is a *data.frame* with up to as many rows as time steps determined by the parameters *tau* and *end.time*. The first column contains the time steps. The second, third and fourth columns contain the number of susceptibles, infectious and recovered.

**References**


**Examples**

```r
# Parameters and initial conditions.
parameters <- c(beta = 1, gamma = 1 / 10, mu = 5e-4, N = 50, tau = 1)
initials <- c(X = 5, Y = 1, Z = 44)
end.time <- 2 * 365

# Solve and plot.
sir.demog.stoch <- SIR Tau Leap(pars = parameters, init = initials,
                                 end.time = end.time)
PlotMods(sir.demog.stoch)
```

**Description**

Solves a SIR model with corrected term-time forcing of the transmission rate.

**Usage**

```r
SIRTermTimeForcing(pars = NULL, init = NULL, term.times = terms,
cycles = 10, low.term.first = TRUE)
```

**Arguments**

pars *list* with 4 values: the mean transmission rate, a scalar (or a vector to create bifurcations) with the amplitude of sinusoidal forcing, the removal recovery rate, and the per capita death rate. The names for these values must be: "beta0", "beta1", "gamma", and "mu", respectively. All parameters must be positive and beta1 <= 1.
**SIRTermTimeForcing**

init vector with 3 values: the initial proportion of proportion of susceptibles, infectious and recovered. The names of these values must be "S", "I" and "R", respectively. $S + I + R <= 1$.

term.times vector indicating the term times (see details and example).
circles value indicating how many times term.times must be simulated (see details and example).

low.term.first logical. If TRUE (default), the first term-time is considered -1, the second 1, the third -1 and so on. When FALSE, the first term-time is 1, the second -1, and so on (see example).

... further arguments passed to ode function.

**Details**

This is the R version of program 5.2 from page 171 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

This model is based on the behaviour os measles and other child-hood diseases. Transmission rate is low during term == -1 (e.g. holyday term) and high during term == 1 (e.g. school term). We can define the year as the temporal unit of circles and each circle is composed by a term-time sequence (see example).

**Value**

list. The first element, *$model$, is the model function. The second, third and fourth elements are vectors ({$pars$, *$init$, *$time$, respectively) containing the pars, init and time arguments of the function. The fifth element *$results$ is a data.frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the proportion of susceptibles, infectious and recovered.

**References**


**See Also**

ode.

**Examples**

```r
# Parameters and initial conditions.
initials <- c(S = 1/17, I = 1e-4, R = 1 - 1/17 - 1e-4)
parameters <- list(beta0 = 17 / 13, beta1 = 0.25,
                   gamma = 1 / 13, mu = 1 / (50 * 365))

# Term-times and cycles
# In a year-unit cicle, holidays happen for example
# between days 1 and 6, 101 and 115, 201 and 251,
# 301 and 307 and 308 and 365.
# Setting low.term.first == TRUE (default) we define the
```
SIRVector

SIR model for mosquito vectors (P 4.4).

Description
Solves a simple SIR model for mosquito vectors.

Usage
SIRVector(pars = NULL, init = NULL, time = NULL, ...)
time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 4.2 from page 123 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first element, *$model, is the model function. The second, third and fourth elements are vectors (*$pars, *$init, *$time, respectively) containing the pars, init and time arguments of the function. The fifth element *$results is a data.frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the proportion of susceptibles, infectious and recovered.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(muH = 5.5e-5, muM = 0.143,
               vH = 5.5e-2, vM = 1.443e3,
               betaHM = 0.5, betaMH = 0.8,
               gamma = 0.033, r = 0.5 / 1e3)

initials <- c(XH = 1e3, XM = 1e4, YH = 1, YM = 1)

# Solve and plot.
sir.vector <- SIRVector(pars = parameters,
                         init = initials,
                         time = 0:1000)

PlotMods(sir.vector)
SIS

**Simple SIS model (P 2.5).**

**Description**

Solves a simple SIS model without births or deaths.

**Usage**

```r
SIS(pars = NULL, init = NULL, time = NULL, ...)```

**Arguments**

- `pars` vector with 2 values: the transmission and recovery rates. The names of these values must be "beta", and "gamma", respectively.
- `init` vector with 2 values: the initial proportion of proportion of susceptibles and infectious. The names of these values must be "S" and "I", respectively.
- `time` time sequence for which output is wanted; the first value of times must be the initial time.
- `...` further arguments passed to `ode` function.

**Details**

This is the R version of program 2.5 from page 39 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

All parameters must be positive and S + I <= 1.

**Value**

*list*. The first element, `*model`, is the model function. The second, third and fourth elements are the vectors (`*pars`, `*init`, `*time`, containing the `pars`, `init` and `time` arguments of the function. The fifth element `*results` is a `data.frame` with up to as many rows as elements in time. First column contains the time. Second and third columns contain the proportion of susceptibles and infectious.

**References**


**See Also**

- `ode`.
Examples

```r
# Parameters and initial conditions.
parameters <- c(beta = 1.4247, gamma = 0.14286)
initials <- c(S = 1 - 1e-06, I = 1e-06)

# Solve and plot.
sis <- SIS(pars = parameters, init = initials, time = 0:70)
PlotMods(sis)
```

**SIS2RiskGroups**

*SIS model with 2 risk groups (P 3.1).*

Description

Solves a SIS model with high-risk (H) and low-risk (L).

Usage

```r
SIS2RiskGroups(pars = NULL, init = NULL, time = NULL, ...)
```

Arguments

- **pars** vector with 6 values: 4 transmission rates, 1 recovery rate and the proportion of the population that are in the high-risk group. The names of these values must be "betaHH", "betaHL", "betaLL", "betaLH", "gamma" and "nH", respectively. The letters after the word "beta" denote transmission to any group from any group, e.g., "betaHL" represent transmission to high-risk group from low-risk group. All parameters must be positive.

- **init** vector with 2 values: the initial proportion of infectious in the high-risk group and the initial proportion of infectious in the low-risk group. The names of these values must be "IH" and "IL", respectively.

- **time** time sequence for which output is wanted; the first value of times must be the initial time.

- **...** further arguments passed to ode function.

Details

This is the R version of program 3.1 from page 58 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

All parameters must be positive and \(nH \leq 1, IH(0) \leq nH, IL(0) \leq 1-nH.\)
Value

list. The first element, *Dmodel*, is the model function. The second, third and fourth elements are the vectors (*Dpars*, *Dinit*, *Dtime*, containing the pars, init and time arguments of the function. The fifth element *Dresults* is a data.frame with up to as many rows as elements in time. First column contains the time. The following columns contain the proportion of susceptibles and infectious.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- c(betaHH = 10.0, betaHL = 0.1, betaIH = 0.1, betaIL = 1.0, gamma = 1, nH = 0.2)
initials <- c(IH = 0.00001, IL = 0.001)

# Solve and plot.
sis2risk.groups <- SIS2RiskGroups(pars = parameters,
        init = initials, time = 0:15)
PlotMods(sis2risk.groups, variables = c('IL', 'IH'), grid = FALSE)

SIS model with demographic stochasticity (P 6.3).

Description

SIS model with event-driven or demographic stochasticity.

Usage

SISDemogStoch(pars, init, end.time)

Arguments

pars vector with 3 values: the transmission and recovery rates and the population size assumed to be constant. The names of these values must be "beta", "gamma", and "N" respectively.

init initial number of infectious.

date time to be simulated.
Details

This is the R version of program 6.3 from page 202 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first three elements are the vectors *$pars, *$init and *$time, containing the pars, init and end.time arguments of the function. The fourth element *$results is a data.frame with up to as many rows as time steps created during the stochastic simulations. The second column contains the number of infectious.

References


Examples

# Parameters and initial conditions.
parameters <- c(beta = 0.03, gamma = 1 / 100, N = 100)
initials <- 70

# Solve and plot.
xis.demog.stoch <- SISdemogStoch(pars = parameters,
                                init = initials, end.time = 10 * 365)
PlotMods(xis.demog.stoch)

SISinusoidalTransmBrith

Rabbit Hemorrhagic Disease model with sinusoidal transmission rate and per capita birth rate (P 5.4).

Description

Solves the Rabbit Hemorrhagic Disease, in which both transmission rate and birth rates can be seasonally forced.

Usage

SISinusoidalTransmBrith(pars = NULL, init = NULL, time = NULL, ...)
Arguments

pars list with: the mean transmission rate, the amplitude of sinusoidal forcing (transmission), the mean birth rate, the amplitude of sinusoidal forcing for the birth rate, the frequency of the oscillations, the recovery rate, the per capita death rate, the mortality rate due to infection, and the carrying capacity. The names of these values must be "beta0", "beta1", "alpha0", "alpha1", "w", "gamma", "mu", "m" and "K", respectively. All parameters must be positive, alpha1, beta1 <= 1.

init vector with 3 values: the initial numbers of susceptible hosts (rabbits), infectious hosts (rabbits) and total population size. The names of these values must be "X", "Y" and "N", respectively. All initial values must be positive and X(0) + Y(0) <= N(0).

time time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 5.4 from page 186 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

Value

list. The first element, *$model, is the model function. The second element is a list with the *$pars argument. The third and fourth elements are the vectors *$init and *$time, containing the init and time arguments of the function. The fifth element *$results is a data.frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the number of susceptibles, infectious and recovered.

References


See Also

ode.

Examples

# Parameters and initial conditions.
parameters <- list(beta0 = 0.936, beta1 = 0.1, alpha0 = 0.02, alpha1 = 0.1,
                   w = 2 * pi / 365, gamma = 0.025, mu = 0.01, m = 0.475,
                   K = 100000)
initials <- c(X = 0.5, Y = 0.01, N = 0.6)

# Solve and plot.
sis.rhdm <- SISinusoidalTransmBrith(pars = parameters,
                                     init = initials,
                                     time = 0:(60 * 365))
SIS model with multiple risk groups (P 3.2).

Description

Solves a SIS model with n different risk-groups.

Usage

SISnRiskGroups(pars = NULL, init = NULL, time = NULL, ...)

Arguments

pars list with: the number of risk groups, a (m x m) matrix with the transmission rates, a vector with the recovery rates (one for each risk group) and a vector with the proportions of the population that are in each risk group. The names of these elements must be "m", "beta", "gamma" and "n", respectively (see example). All rates are specified in years.

init vector with m * 2 values: the initial proportions of susceptibles and infectious in each risk group. The names of these values must be "S1",..., "Sm" and "I1",..., "Im", respectively (see example). All initial states must be positive and Si + Ii = ni, i= 1, ..., m.

time time sequence for which output is wanted; the first value of times must be the initial time.

... further arguments passed to ode function.

Details

This is the R version of program 3.2 from page 64 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

All parameters must be positive, and ni <= 1, sum(ni) = 1, i = 1, ..., m.

Value

list. The first element, *$model, is the model function. The second, third and fourth elements are the vectors (*$pars, *$init, *$time, containing the pars, init and time arguments of the function. The fifth element *$results is a data.frame with up to as many rows as elements in time. First column contains the time. The following columns contain the proportion of susceptibles and infectious of each risk group.

References

SISPairwiseApprox

See Also
ode.

Examples

# Parameters and initial conditions.
tmp <- matrix(c(0, 3, 10, 60, 100))
beta <- 0.0016 * tmp %*% t(tmp)
parameters <- list(m = 5, beta = beta,
  gamma = c(0.2, 0.2, 0.2, 0.2, 0.2),
  n = c(0.06, 0.31, 0.52, 0.08, 0.03))
initials <- c(I = c(0, 0, 0, 0, 1e-5))

# Solve and plot.
xis.n.riks.groups <- SISnRiskGroups(pars = parameters,
  init = initials,
  time = 0:30)
PlotMods(xis.n.riks.groups, grid = FALSE)

SISPairwiseApprox  Pairwise SIS approximation model (P 7.8).

Description
Solves a pairwise approximation to the SIS model on a random network of N individuals, each with n contacts.

Usage
SISPairwiseApprox(pars = NULL, init = NULL, time = NULL, ...)

Arguments
pars  vector with 4 values: the transmission rate across a contact, the recovery rate for infectious individuals, the number of connections per individual in the population and the number of individuals in the population. The names of these values must be "tau", "gamma", "n" and "N" respectively.

init  vector with 3 values: the initial number of susceptibles, infectious and susceptible-infectious pairs. The names of these values must be "X", "Y" and "XY", respectively.

time  time sequence for which output is wanted; the first value of times must be the initial time.

...  further arguments passed to ode function.
Details

This is the R version of program 7.8 from page 285 of "Modeling Infectious Disease in humans and animals" by Keeling & Rohani.

All parameters must be positive.

Value

list. The first element, *$model*, is the model function. The second, third and fourth elements are the vectors *$pars*, *$init* and *$time*, containing the pars, init and time arguments of the function. The fifth element *$results* is a data.frame with up to as many rows as elements in time. First column contains the time. Second, third and fourth columns contain the number of susceptibles, infectious and susceptible-infectious pairs.

References


See Also

ode.

Examples

# Parameters and initial conditions.
n <- 4; N <- 1e4; Y <- 1; X <- N - Y
parameters <- c(tau = 0.1, gamma = 0.05, n = n, N = N)
initials <- c(X = X, Y = Y, XY = n * Y * X / N)

# Solve and plot.
sis.pairwise.approx <- SISPairwiseApprox(pars = parameters,
init = initials, time = 0:100)
PlotMods(sis.pairwise.approx)
Index

*Topic package
  EpiDynamics-package, 2
  data.frame, 3, 6, 7, 9–11, 13–16, 18–20, 22–24, 26, 27, 29, 30, 32, 33, 35–38, 40
  SISinusoidalTransmBrth, 36
  SISnRiskGroups, 38
  SISPairwiseApprox, 39

EpiDynamics-package, 2

list, 3, 6, 7, 9–11, 13–16, 18–20, 22–27, 29, 30, 32, 33, 35–38, 40

MultiStrainPartialImmunity, 3

ode, 3–27, 30, 32–35, 37–40

PlotMods, 4

SEIR, 5
SEIR4AgeClasses, 6
SEIRnStages, 8
SIR, 9
sir2AgeClasses, 11
SIR2Stages, 12
SIR2TypesImports, 13
SIRAdditiveNoise, 15
SIRBirthDeath, 16
SIRCcarrierState, 17
SIRDemogStoch, 18
SIRInducedMortality, 20
SIRInducedMortality2, 21
SIRPartialImmunity, 22
SIRScaledAdditiveNoise, 24
SIRSinusoidalBirth, 25
SIRSinusoidalForcing, 27
SIRTauLeap, 28
SIRTermTimeForcing, 29
SIRVector, 31
SIS, 33
SIS2RiskGroups, 34
SISDemogStoch, 35