Package 'MANTIS'

November 19, 2015

Type Package

Title Multilocus ANTIgenic Simulator

Version 3.0

Author Jose Lourenco and Paul Wikramaratna

Maintainer Jose Lourenco <jose.lourenco@zoo.ox.ac.uk>

Description an R package that simulates multilocus models of pathogen evolution.

License GPL-3

Imports Rcpp (>= 0.11.2)

LinkingTo Rcpp

RoxygenNote 5.0.1

NeedsCompilation yes

R topics documented:

`MANTIS-package` *Multilocus ANTIgenic Simulator*

Description

This package can be used to simulate the population dynamics of antigenically diverse pathogens according to the multilocus antigenic framework by Gupta et al. (see reference 1). Its functions allow the user to generate host-pathogen time-series and analyse critical dynamic patterns within; both can be exported into comma-separated values files (CSV) and PDF plots.

MANTIS is developed by the Evolutionary Ecology of Infectious Disease (EEID) research group at the Department of Zoology of the University of Oxford.

Summary of the framework: Transmission follows some basic assumptions of a classic susceptible-infectious-recovered (SIR) epidemiological framework. Hosts are segregated into 3 pertinent classes, `Z`, `W`, and `Y`. For each possible pathogen strain `i` in the system, individuals with cross-reactive immune responses are represented by `Wi`, infectious by `Yi` and with specific responses by `Zi` (for the corresponding ordinary-differential equations the reader should refer to reference 2). A pathogen strain is defined by the genetic loci that encode relevant antigens, against which immune responses

act, rather than by its entire genome. For example, the notation `{3,4,10}`, would indicate a system with three loci `i,k,j` and `N_i=3`, `N_k=4` and `N_j=10` alleles, respectively (for any given structure, the total number of possible strains is `N_i * N_k * ... * N_n`. Strains that do not share alleles do not interfere with each others' success as mediated by host immune responses. The degree to which infection with a given strain limits the ability of another strain (sharing alleles) to infect the same host is defined by a cross-immunity parameter `gamma`. If `gamma=0` the strains do not induce cross-protective responses. If `gamma=1` there is complete cross-protection. Otherwise, immunity to a given strain only reduces the probability of transmission of another by a factor `1-gamma`.

Methodology: These simple but realistic biological assumptions can be expressed by a system of ordinary-differential equations (ODE) representing the changes in the abundances of each of the strains over time in a genetically and demographically homogeneous host population. Here, a range-kutta algorithm with fixed-step size is used to solve an ODE system, thus numerically simulating the above described epidemiological framework.

This version: MANTIS V3.0 (cryptonym: "red egg")

Referencing: Please reference the MANTIS research article: 'MANTIS: an R package that simulates multilocus models of pathogen evolution. BMC Bioinformatics. 2015 May 28;16:176. doi: 10.1186/s12859-015-0598-9.'

Installation (source): Open an R session. Make sure the session is open in the same folder as the MANTIS package file. Run `install.packages("XXX",repos=NULL,type="src")`, where XXX is the package file name. Note: MANTIS requires the preinstallation of Rcpp (see `http://cran.r-project.org/web/packages/Rcpp/index.html`); and, installing source packages on machines with the Windows operating system may require the preinstallation of Rtools (see `http://cran.r-project.org/bin/windows/Rtools/`).

Installation (repository): MANTIS is not available in official R repositories.

Getting Started: For a general example, see the 'examples' section on this document.

Updates: Updates are planned to be submitted to official repositories and will also be available at the Evolutionary Ecology of Infectious Disease (EEID) group's webpage (see "See Also" for links)

References

(1) Chaos, persistence, and evolution of strain structure in antigenically diverse infectious agents. Gupta S, Ferguson N, Anderson R. Science. 1998 May 8;280(5365):912-5.

(2) The antigenic evolution of influenza: drift or thrift? Wikramaratna PS, Sandeman M, Recker M, Gupta S. Philos Trans R Soc Lond B Biol Sci. 2013 Feb 4;368(1614):20120200.

(3) Transient cross-reactive immune responses can orchestrate antigenic variation in malaria. Recker

M, Nee S, Bull PC, Kinyanjui S, Marsh K, Newbold C, Gupta S. Nature. 2004 Jun 3;429(6991):555-8.

(4) Role of stochastic processes in maintaining discrete strain structure in antigenically diverse pathogen populations. Buckee CO, Recker M, Watkins ER, Gupta S. Proc Natl Acad Sci U S A. 2011 Sep 13;108(37):15504-9.

(5) The generation of influenza outbreaks by a network of host immune responses against a limited set of antigenic types. Recker M, Pybus OG, Nee S, Gupta S. Proc Natl Acad Sci U S A. 2007 104, 7711-7716.

See Also

Evolutionary Ecology of Infectious Disease (EEID) group
`www.eeid.ox.ac.uk`

EEID on Twitter, follow us, get in touch!
`www.twitter.com/EEID_oxford`

Department of Zoology of the University of Oxford
`www.zoo.ox.ac.uk`

Examples

```
library(MANTIS)

#define epitope structure, with 2 loci
#with 4 and 2 alleles each (8 strains)
epiStruc<- c(4,2)
#get number of strains in the system
nStrains<- extractNumberStrains(epiStruc)
#time at which the solver will stop
tMax<- 1000
#solver step
tInt<- 0.005
#record solution every tObsPer steps
tObsPer<- 100
#random initial conditions for infected
initCondY <- runif(nStrains, 0, 1)*1e-4

#parameter values here defined by year
#thus output time scale will be years
beta<-  292
```

```
sigma<-  (1/5)*365 #5 days infectious period
mu<-  1/50 #50 years of life-span

#running model with weak cross-immunity
gamma<- 0.1
simdata<- runMANTIS(epiStruc, tMax, tObsPer,
tInt, initCondY, beta, gamma, sigma, mu)

#plotting output with weak cross-immunity to the screen
plotY(simdata, xiObs=0.1, xfObs=0.2)

#plotting output with weak cross-immunity to a PDF file
plotY(simdata, fileout="weak.ts.pdf", xiObs=0.1, xfObs=0.2)

#plotting diversity to PDF files
plotYDiversity(simdata,fileout="diveristyY.pdf",
method="shannon",xiObs=0.5, xfObs=0.8)

#export diversity of Z class into a text file
exportZDiversity(simdata, fileout="diversityZ.csv")

#calculate single-strain dominance
epsilon<-  calcSingleStrainDominance(simdata)
cat(paste("Single-strain dominance is epsilon=", epsilon, "\n"))

#running model with intermediate cross-immunity
gamma<- 0.6
simdata<- runMANTIS(epiStruc, tMax, tObsPer,
tInt, initCondY, beta, gamma, sigma, mu)

#select strains 1 and 2
subSetStrains<- c(1,2)
#plot highlighting a subset of strains to file
plotY(simdata, fileout="highlighting.strains_1_2.pdf", colours=c("blue","grey"),
 markStrains=subSetStrains, xiObs=0.5, xfObs=0.53)

#get strains that share epitopes with strain 1, including 1
strainsShareWith1<- withSharedEpitopes(simdata,1)
#plot highlighting a subset of strains
plotY(simdata, colours=c("orange","black"), markStrains=strainsShareWith1,
xiObs=0.5, xfObs=0.53)
#plotting diversity to PDF files
plotYDiversity(simdata, method="shannon", xiObs=0.5, xfObs=0.53)

#exporting output into a CSV text file
exportTS(simdata, fileout="intermediate.all.strains.csv")

#exporting shared epitopes matrix into a CSV text file
exportSharedEpitopes(simdata, fileout="intermediate.shared.epitopes.csv")

#plot relative prevalence of the strains in time
plotYRelPrevalence(simdata, fileout="relative.prevalence.plot.pdf",
xiObs=0.5, xfObs=0.54, method="host")
```

```
#measure single-strain dominance from a range of cross-immunity values, gamma
gammas<- seq(0,1,length.out=10)

#measure within a small antigenic space
epiStruc<- c(2,2)
initCondY <- runif(extractNumberStrains(epiStruc), 0, 1)*1e-4
SSD_sAntiSpace<- measureSSDfromGammaRange(gammas, "host", epiStruc, tMax,
tObsPer, tInt, tStep, initCondY, beta, sigma, mu)

#measure within a medium sized antigenic space
epiStruc<- c(4,2)
initCondY <- runif(extractNumberStrains(epiStruc), 0, 1)*1e-4
SSD_mAntiSpace<- measureSSDfromGammaRange(gammas, "host", epiStruc, tMax,
tObsPer, tInt, tStep, initCondY, beta, sigma, mu)

#measure within a larger sized antigenic space
epiStruc<- c(6,2)
initCondY <- runif(extractNumberStrains(epiStruc), 0, 1)*1e-4
SSD_lAntiSpace<- measureSSDfromGammaRange(gammas, "host", epiStruc, tMax,
tObsPer, tInt, tStep, initCondY, beta, sigma, mu)

#plot SSD measures agains gamma
colours=c("blue3","gold","tomato3")
pdf("SSDs_various_gammas_systems.pdf", width=5, height=4, bg="white")
plot(gammas, SSD_sAntiSpace, t='b', col=colours[1], lwd=2, ylab="SSD")
lines(gammas, SSD_mAntiSpace, t='b', col=colours[2], lwd=2)
lines(gammas, SSD_lAntiSpace, t='b', col=colours[3], lwd=2)
legend("topleft", legend=c("small antigenic space","med. antigenic space"
,"big antigenic space"),col=colours, lwd=2, cex=0.8)
a<- dev.off()

###will define a bigger antigenic space to have a look at antigenic dynamics

#define epitope structure
epiStruc<- c(5,5,5)
#get number of strains in the system
nStrains<- extractNumberStrains(epiStruc)
#time at which the solver will stop
tMax<- 500
#solver step
tInt<- 0.005
#record solution every tObsPer steps
tObsPer<- 100
#random initial conditions for infected
initCondY <- runif(nStrains, 0, 1)*1e-9

#parameter values here defined by year
#thus output time scale will be years
beta<- 292
sigma<-  (1/5)*365 #5 days infectious period
mu<-  1/50 #50 years of life-span
```

```
#running model with weak cross-immunity
gamma<- 0.75
simdata<- runMANTIS(epiStruc, tMax, tObsPer,
tInt, initCondY, beta, gamma, sigma, mu)

#plotting antigenic dynamics
plotAntigenicSpaceDyn(simdata, extThreshold=1e-4, fileout="antigenicspace.pdf")
#export table with dominance output
exportDominance(simdata, extThreshold=1e-4)
#export table with novelty output
exportNovelty(simdata, extThreshold=1e-4)

#export allelic matrix
exportAllelicMatrix(simdata, fileout="allelic.matrix.csv")

#set an extinction threshold
extThreshold=1e-50

#export sequences of dominant strains in time
exportSeqOfDomVariant(simdata, extThreshold=extThreshold, fileout="sequences.dominant.tim

##plot  sequences of dominant strains in time
plotSeqsOfDomVariant(simdata, epiStruc=epiStruc,
extThreshold=extThreshold, fileout="allelesintime.pdf")

##testing differences in transmission phenotype
##(one strain transmits more than others)

#define epitope structure, with 2 loci
epiStruc<- c(2,2)
#get number of strains in the system
nStrains<- extractNumberStrains(epiStruc)
#time at which the solver will stop
tMax<- 1000
#solver step
tInt<- 0.005
#record solution every tObsPer steps
tObsPer<- 100
#random initial conditions for infected
initCondY <- runif(nStrains, 0, 1)*1e-4

#parameter values here defined by year
#thus output time scale will be years
beta<-  292
sigma<-  (1/5)*365 #5 days infectious period
mu<-  1/50 #50 years of life-span

beta<- rep(292, nStrains)
beta[2]<- 292*2

#running model with weak cross-immunity
gamma<- 0.1
simdata<- runMANTIS(epiStruc, tMax, tObsPer, tInt, initCondY, beta, gamma, sigma, mu)
```

```
#plotting output with weak cross-immunity to the screen
plotY(simdata, xiObs=0.2, xfObs=0.5)

##running a simulation in NSS with seasonality
##all transmission rates are affected equally

#define antigenic structure
epiStruc<- c(2,2)
#time at which the solver will stop
tMax<- 750
#solver step
tInt<- 0.01

#record solution every tObsPer steps
tObsPer<- 10

#parameter values here defined by year
#thus output time scale will be years
beta<- 292
sigma<- (1/5)*365 #5 days infectious period
mu<- 1/50 #50 years of life-span

#running model with weak cross-immunity
gamma<- 0.01
epsilon<- 0.25
nLocus<- length(epiStruc)
nStrains<- extractNumberStrains(epiStruc)
infInitCond <- runif(nStrains, 0, 1)*1e-4

simdata<- runMANTIS(epiStruc, tMax, tObsPer, tInt, infInitCond, beta, gamma, sigma, mu, e
plotPathogenPrevalence(simdata, xiObs=0.8, xfObs=0.83)

##running a simulation in which strain 2 evolves
##an advantageous transmission phenotype at t=650

#define epitope structure, with 2 loci
epiStruc<- c(3,2)
#get number of strains in the system
nStrains<- extractNumberStrains(epiStruc)
#maximum time for the simulation
tMax<- 1000
#time step for the simulation
tInt<- 0.005
#record solution every tObsPer steps
tObsPer<- 100

#random initial conditions for infected
initCondY <- rep(1e-4, nStrains)
#strain 1 set with a slightly higher prevalence
initCondY[1]<- initCondY[1]+ 0.00001

#default transmission for all strains
```

```
beta<- rep(292, nStrains)
#cross immunity for this simulation
gamma<- 0.98
#5 days infectious period
sigma<- (1/5)*365
#50 years of life-span
mu<- 1/50

#define here the extra parameters for this experiment
tBetaChange= 650 #time at which beta will be changed
changedStrain= 2 #which strain to change beta for
changedBeta= 292*10

#run MANTIS
simdata<- runInvasionWithOneBetaChange(epiStruc, tMax, tObsPer, tInt, initCondY,
 beta, gamma, sigma, mu, tBetaChange, changedBeta, changedStrain)

#plot entire simulation
plotY(simdata, xiObs=0.65, xfObs=0.99, ymax=0.001, addLegend=TRUE)
```

`aggregateSimulations`

Aggregate two MANTIS simulations into one.

Description

Returns a new simulation.

Usage

```
aggregateSimulations(data1, data2)
```

Arguments

`data1`	a list as obtained from running the simulation. See `?runMANTIS` for details.
`data2`	a list as obtained from running the simulation. See `?runMANTIS` for details.

Value

a list as obtained from running the simulation but now composed of data1 followed by data2.

calcDiversity *Calculates diversity measures on strain's time-series.*

Description

A matrix with time-series is used to generate estimates of diversity in time. These estimates can be generated by two standard methods: `Shannon` and `Simpson` indexes. The Shannon Index is defined as `SH = -sum Yr_i log Yr_i`, where `Yr_i` is the proportional abundance of strain `i`; log is the exponential logarithm. The Simpson Index is based on `SI = sum Yr_i^2`, returning `1-SI`.

Usage

```
calcDiversity(tsSeries, xiObs = 0.5, xfObs = 1, method = "shannon")
```

Arguments

`tsSeries`	a matrix in which columns are the time-series to apply the diversity method. Such a matrix can be obtained from running a simulation with `runMANTIS`. For example, if `simdata=runMANTIS(...)`, then `Y=extractY(simdata)` can be used to obtain a matrix of infectious classes. See `?runMANTIS`, `?extractY`, `?extractW` and `?extractZ` for details.
`xiObs`	the starting proportion of time to use for diversity. For example `xiObs=0.5` will start from the middle of the time-series.
`xfObs`	the end proportion of time to use for diversity.
`method`	a character string of two possible values, `"shannon"` or `"simpson"`, see description for details.

Value

a numeric array for the diversity in time.

calcSingleStrainDominance

Calculate a measure of single-strain dominance (SSD) (`epsilon`).

Description

This function estimates a measure of SSD for a simulated time-series of infectious individuals (Y). Single-strain dominance can be quantified by `epsilon`, the measure obtained by comparing the relative prevalence of the two most common antigenic variants within single epidemics and then averaging across extended periods of time, or, more formally, averaging across each of the epidemics within the studied time window. The number and time point of each epidemic `Pi` is determined by addressing total pathogen prevalence and identifying local maxima. Here, two estimates of

total pathogen prevalence can be used via the argument `method`, see bellow. `epsilon` is expected to vary between 0 and 1, with high values indicative of strong SSD within epidemics, as often found, for instance, in the antigenic evolution and epidemiological behaviour of influenza A viruses. `epsilon` has been used in several research publications - see references (2) and (5) for formal definitions (found in the documentation or output of function `readingMANTIS`).

Usage

```
calcSingleStrainDominance(data, xiObs = 0.5, xfObs = 1, method = "host")
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`xiObs`	the starting proportion of time to use for diversity. For example `xiObs=0.5` will start from the middle of the time-series.
`xfObs`	the end proportion of time to use for diversity.
`method`	a character string to choose the method to be used. Methods are `"host"` and `"strain"` and relate to how total prevalence of the pathogen is estimated. When `method="host"` the function `extractPathogenPrevByStrain` is called and when `method="strain"` the function `extractPathogenPrevByHost` is called. See the help of each function for how they differ. Estimations are expected to be very similar using the two methods.

Value

epsilon

`exportAllelicMatrix` *Export the allelic matrix.*

Description

Creates a comma separated CSV text file in which each row correspondes to an existing strain. Each row `i` contains a set of integers which identify the strain's allele at each existing loci. See the definition of 'allelic matrix' in `?runMANTIS`.

Usage

```
exportAllelicMatrix(data, fileout = "default.filename.csv")
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`fileout`	the name of the output CSV file.

exportDominance *Export antigenic dominance dynamics.*

Description

Creates a comma separated CSV text file in which the columns are the solution of the simulation in terms of antigenic novelty in time. The file (`fileout`) contains the output of which variant (strain) is dominating and the percentage of the antinegic space that has been explored by dominance, in time - first column is time, second column is the variant (strain) number that is dominating and third column the percentage of space already explored, in time.

See `?extractDomVariant` for details on the concept of antigenic dominance.

Usage

```
exportDominance(data, extThreshold, fileout = "dominance.csv")
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`extThreshold`	the theoretical extinction threshold for the pathogen. See `?extractNoveltyVariant` for details.
`fileout`	the name of the output CSV file. It defaults to "dominance.csv".

exportNovelty *Export antigenic novelty dynamics.*

Description

Creates 2 comma separated CSV text files in which the columns are the solution of the simulation in terms of antigenic novelty in time. The first file (`fileout1`) contains the output of which novel variants (strains) have emerged above the `extThreshold` for the first time - first column is time and the second the variant (strain) number. The second file (`fileout2`) containts the output of how much of the possible antigenic space has been explored, through novelty, in time - first column is time, second column is percentage of all antigenic space explored.

See `?extractNoveltyVariant` for details on the concept of antigenic novelty.

Usage

```
exportNovelty(data, extThreshold, fileout1 = "novelty.new.variants.csv",
  fileout2 = "novelty.space.travelled.csv")
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.
extThreshold	the theoretical extinction threshold for the pathogen. See ?extractNoveltyVariant for details.
fileout1	the name of the first output CSV file. It defaults to "novelty.new.variants.csv".
fileout2	the name of the second output CSV file. It defaults to "novelty.space.travelled.csv".

---

exportSeqOfDomVariant

Export the sequences of dominant strains in time.

Description

Creates a comma separated CSV text file in which each row corresponds to a time step and information of the dominant strain at the time. The information is essentially the sequence of the strain. Each row i contains the time, the strain number and a set of integers which identify the strain's allele at each existing loci (the allelic matrix). See the definition of 'allelic matrix' in ?runMANTIS.

Usage

```
exportSeqOfDomVariant(data, extThreshold, fileout = "default.filename.csv")
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.
extThreshold	the theoretical extinction threshold for the pathogen. See ?extractNoveltyVariant for details.
fileout	the name of the output CSV file.

---

exportSharedEpitopes

Export the epitope relationships between strains.

Description

Creates a comma separated CSV text file in which each row correspondes to an existing strain. Each row i contains a set of integers which identify which other strains share epitopes with strain i. See the definition of 'shared epitopes' in ?runMANTIS.

Usage

```
exportSharedEpitopes(data, fileout = "default.filename.csv")
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.
fileout	the name of the output CSV file.

---

exportTS *Export the simulated time-series.*

Description

Creates a comma separated CSV text file in which the columns are the solution of the simulation. First column is time and the remaining columns are, in order, classes Z, W and Y. See the definition of 'time series' in ?runMANTIS.

Usage

```
exportTS(data, fileout = "default.filename.csv")
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.
fileout	the name of the output CSV file.

---

exportWDiversity *Export the simulated time-series' diversity of the cross-reactive (W) classes.*

Description

Creates a comma separated CSV text file in which the columns are the solution of the simulation in diversity measures for cross-reactive (W) classes. First column is time and the second and third column are the default diversity measures. See ?calcDiversity for details on the measures.

Usage

```
exportWDiversity(data, xiObs = 0.5, xfObs = 1,
  fileout = "default.filename.csv")
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.
xiObs	the starting proportion of time to use for diversity. For example xiObs=0.5 will start from the middle of the time-series.
xfObs	the end proportion of time to use for diversity.
fileout	the name of the output CSV file.

exportYDiversity *Export the simulated time-series' diversity of the infectious (Y) classes.*

Description

Creates a comma separated CSV text file in which the columns are the solution of the simulation in diversity measures for infectious (Y) classes. First column is time and the second and third column are the default diversity measures. See ?calcDiversity for details on the measures.

Usage

```
exportYDiversity(data, xiObs = 0.5, xfObs = 1,
  fileout = "default.filename.csv")
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.
xiObs	the starting proportion of time to use for diversity. For example xiObs=0.5 will start from the middle of the time-series.
xfObs	the end proportion of time to use for diversity.
fileout	the name of the output CSV file.

exportYRelPrevalence *Export the infectious (Y) classes in terms of relative prevalence.*

Description

Creates a comma separated CSV text file in which the columns are the relative prevalence of each strain in the infectious (Y) classes. First column is time and the remaining columns are the relative prevalences (between 0 and 1) calculated by dividing each strain's prevalence by the total pathogen prevalence.

Usage

```
exportYRelPrevalence(data, fileout = "default.filename.csv",
  method = "host")
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`fileout`	the name of the output CSV file.
`method`	a character string to choose the method to be used for the estimation of the pathogen's total prevalence. Methods are `"host"` and `"strain"` and relate to how total prevalence of the pathogen is estimated. When `method="host"` the function `extractPathogenPrevByStrain` is called and when `method="strain"` the function `extractPathogenPrevByHost` is called. See the help of each function for how they differ. Estimations are expected to be very similar using the two methods.

---

exportZDiversity *Export the simulated time-series' diversity of the specific-immunity (Z) classes.*

Description

Creates a comma separated CSV text file in which the columns are the solution of the simulation in diversity measures for specific-immunity classes (Z). First column is time and the second and third column are the default diversity measures. See ?calcDiversity for details on the measures.

Usage

```
exportZDiversity(data, xiObs = 0.5, xfObs = 1,
  fileout = "default.filename.csv")
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`xiObs`	the starting proportion of time to use for diversity. For example `xiObs=0.5` will start from the middle of the time-series.
`xfObs`	the end proportion of time to use for diversity.
`fileout`	the name of the output CSV file.

---

extractAllelicMatrix *Extract the allelic matrix.*

Description

Returns the allelic matrix of a simulation. See `?runMANTIS` for the definition of allelic matrix.

Usage

```
extractAllelicMatrix(data)
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.

Value

a matrix which represents the allelic matrix.

---

extractDomVariant *Extract antigenic dominance dynamics.*

Description

Extracts time series containing the solution for antigenic dynamics based on dominance. Dominance is based on prevalence levels, with the variant (strain) presenting the highest prevalence considered to be dominant. This indirectly depends on the assumption of the theoretical host population size. See the definition of extThreshold below for details.

Usage

```
extractDomVariant(data, extThreshold)
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.
extThreshold	theoretical extinction threshold for the pathogen. Since MANTIS is based on deterministic and continuous numerical simulations, strains are always present in the population (their prevalences are always bigger than zero). This is particularly important for the concept of antigenic novelty (see ?extractNoveltyVariant) and therefore becomes necessary for the concept of antigenic dominance - dominance should only be analysed in the context of novelty, since dominant variants must have emerged (appeared for the first time) before they dominate.

Value

a data.frame in which the first column is time, the second column is the variant (strain) number that is dominating and the third column is the percent of antigenic space explored by dominance. Values NA are used for time steps in which all variants are below the extinction threshold and therefore dominance is not biologically relevant.

`extractFinalConditions` *Get the population state (condition) at the last time step of a simulation.*

Description

Returns the population state.

Usage

```
extractFinalConditions(data)
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.

Value

a numeric array with the population state, with the class order `Zi,Wi,Yi`.

`extractNoveltyVariant` *Extract antigenic novelty dynamics.*

Description

Extracts time series containing the solution for antigenic dynamics based on novelty. Novelty is defined as the first appearance of variants (strains) in the population. This crucially depends on the assumption of the theoretical host population size. See the definition of `extThreshold` below for details.

Usage

```
extractNoveltyVariant(data, extThreshold)
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`extThreshold`	theoretical extinction threshold for the pathogen. Since MANTIS is based on deterministic and continuous numerical simulations, strains are always present in the population (their prevalences are always bigger than zero). This is particularly important for the concept of antigenic novelty - by necessarily having to initialize all strains at prevalences bigger than zero, novelty is readily 100 percent at the initial time step. For this reason, the consideration of an extinction

threshold was added. This allows the user to consider theoretical (host) population sizes, assuming that prevalences under the threshold should equate to the virtual extinction of pathogen strains. Variants therefore emerge (appear for the first time) as novel types only when they breach the threshold.

Value

a list with two data.frames. The first data.frame contains the time series in which the first column is time and second column is the number of the variant (strain) that has emerged for the first time above the theoretical extinction threshold (`extThreshold`). The second data.frame contains the time series in which the first column is time and the second column is the percent of the antigenic space that has been explored by novelty.

`extractNumberStrains`

Get the number of possible strains.

Description

Returns the number of strains that will exist in a system with antigenic structure 'epiStruc'.

Usage

```
extractNumberStrains(epiStruc)
```

Arguments

`epiStruc`	the antigenic structure of the system. See `?runMANTIS` for details.

Value

an integer number of strains in the system as defined by 'epiStruc'

`extractPathogenPrevByHost`

Returns a time-series for the total prevalence of the simulated pathogen.

Description

The number of infectious individuals to each strain (Y) is used to estimate the total pathogen prevalence, i.e. as the fraction of the population that is infected with at least one strain. In effect, each infectious class is considered independently and each infected individual is counted once, regardless of how many times they may be infected. This value is therefore bounded between 0 and 1.

Usage

```
extractPathogenPrevByHost(data)
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.

Value

a numeric array with the time-series of the total prevalence for the simulated pathogen.

---

extractPathogenPrevByStrain

Returns a time-series for the total prevalence of the simulated pathogen.

Description

The number of infectious individuals to each strain (Y) is used to calculate the total pathogen prevalence, of the pathogen: in effect, each infected individual is counted as many times as they are infected. It is therefore possible for this value to exceed 1 in cases when the prevalence of any individual strain is very high.

Usage

```
extractPathogenPrevByStrain(data)
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.

Value

a numeric array with the time-series of the total prevalence for the simulated pathogen.

`extractSeqOfDomVariant` *Extract the sequences of dominant strains in time.*

Description

Returns a matrix with the sequences of the dominant strains in time. This matrix is essentially the allelic matrix joint with a column for time and a column for the number of the strain that is dominating at each time step. See `?runMANTIS` for the definition of allelic matrix.

Usage

```
extractSeqOfDomVariant(data, extThreshold = 1e-04)
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`extThreshold`	the theoretical extinction threshold for the pathogen. See `?extractNoveltyVariant` for details.

Value

a matrix which represents the sequences of the dominant strains in time.

`extractW` *Returns the time-series for the cross-reactive (W) classes.*

Description

The list returned from `runMANTIS` is used to obtain only the time-series of the W classes. See `?runMANTIS` for what it returns.

Usage

```
extractW(data)
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.

Value

a matrix in which each column is an cross-reactive class `Wi` for strain `i`.

extractY *Returns the time-series for the infectious (Y) classes.*

Description

The list returned from `runMANTIS` is used to obtain only the time-series of the Y classes. See `?runMANTIS` for what it returns.

Usage

```
extractY(data)
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.

Value

a matrix in which each column is an infectious class `Yi` for strain `i`.

extractYRelPrevalences *Returns the relative prevalence time-series for the infectious (Y) classes.*

Description

The list returned from `runMANTIS` is used to obtain only the relative prevalence time-series of the Y classes. See `?runMANTIS` for what it returns.

Usage

```
extractYRelPrevalences(data, method = "host")
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`method`	a character string to choose the method to be used for the estimation of the pathogen's total prevalence. Methods are `"host"` and `"strain"` and relate to how total prevalence of the pathogen is estimated. When `method="host"` the function `extractPathogenPrevByStrain` is called and when `method="strain"` the function `extractPathogenPrevByHost` is called. See the help of each function for how they differ. Estimations are expected to be very similar using the two methods.

Value

a matrix in which each column is an infectious class `Yi` for strain `i`. The series are calculated by dividing each strain's prevalence by the total pathogen prevalence (as obtained by `extractPathogenPrevByStrain`)

`extractZ` *Returns the time-series for the specific-immunity (Z) classes.*

Description

The list returned from `runMANTIS` is used to obtain only the time-series of the Z classes. See `?runMANTIS` for what it returns.

Usage

```
extractZ(data)
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.

Value

a matrix in which each column is an specific-immunity class `Zi` for strain `i`.

`helloMANTIS` *Print a welcome message.*

Description

Print a welcome message from the Multilocus ANTIgenic Simulator (MANTIS) team.

Usage

```
helloMANTIS()
```

`measureDiversityfromGammaRange` *Measures mean strain diversity for a set of different simulations with varying gamma.*

Description

Various simulations can be run sequencially, for the same epidemiological parameters, while varying the cross-immunity parameter `gamma`; for each run, mean strain diversity is calculated.

Usage

```
measureDiversityfromGammaRange(gammas, method, epiStruc, tMax, tObsPer, tInt,
  tStep, initCondY, beta, sigma, mu, xiObs = 0.6, xfObs = 1)
```

Arguments

`gammas`	a numeric array with values for the `gamma` parameter.
`method`	character string for the method to be used in the diversity measure. See `?calcDiversity` for details.
`epiStruc`	see `?runMANTIS` for details.
`tMax`	see `?runMANTIS` for details.
`tObsPer`	see `?runMANTIS` for details.
`tInt`	see `?runMANTIS` for details.
`tStep`	see `?runMANTIS` for details.
`initCondY`	see `?runMANTIS` for details.
`beta`	see `?runMANTIS` for details.
`sigma`	see `?runMANTIS` for details.
`mu`	see `?runMANTIS` for details.
`xiObs`	the starting proportion of time to use for SSD. For example `xiObs=0.5` will start from the middle of the time-series.
`xfObs`	the end proportion of time to use for SSD.

Value

a numeric array with the measures for mean strain diversity in each simulation with different `gamma`.

`measureSSDfromGammaRange` *Measures single-strain dominance (SSD) for a set of different simulations.*

Description

Various simulations can be run sequencially, for the same epidemiological parameters, while varying the cross-immunity parameter `gamma`; for each run, SSD is calculated.

Usage

```
measureSSDfromGammaRange(gammas, method, epiStruc, tMax, tObsPer, tInt, tStep,
  initCondY, beta, sigma, mu, xiObs = 0.6, xfObs = 1)
```

Arguments

`gammas`	a numeric array with values for the `gamma` parameter.
`method`	character string for the method to be used in the SSD measure. See `?calcSingleStrainDomina` for details.
`epiStruc`	see `?runMANTIS` for details.
`tMax`	see `?runMANTIS` for details.
`tObsPer`	see `?runMANTIS` for details.
`tInt`	see `?runMANTIS` for details.
`tStep`	see `?runMANTIS` for details.
`initCondY`	see `?runMANTIS` for details.
`beta`	see `?runMANTIS` for details.
`sigma`	see `?runMANTIS` for details.
`mu`	see `?runMANTIS` for details.
`xiObs`	the starting proportion of time to use for SSD. For example `xiObs=0.5` will start from the middle of the time-series.
`xfObs`	the end proportion of time to use for SSD.

Value

a numeric array with the measures for single-strain dominance in each simulation with different `gamma`.

`plotAntigenicSpaceDyn` *Plot antigenic dynamics.*

Description

Makes 4 plots. Top-left subplot presents the simulated time series of the Y classes with the `extThreshold` highlighted with a dashed black line. Top-right subplot presents antigenic dominance dynamics (See `?extractDomVariant` for details). Bottom-left subplot presents antigenic novelty dynamics (See `?extractNoveltyVariant` for details). Bottom-right subplot presents the percentage of antigenic space explored by novelty and dominance in time. Can create a PDF file.

Usage

```
plotAntigenicSpaceDyn(data, extThreshold, fileout = NA, w = 7 * 1.5, h = 3
  * 2.5, colours = NULL, xiObs = 0, xfObs = 1,
  mainTopLeft = "zoomed-in time-series",
  mainTopRight = "antigenic dominance",
  mainBottomLeft = "antigenic novelty",
  mainBottomRight = "antigenic space explored in time",
  yylabTopLeft = "proportion infected", yylabTopRight = "variant",
  yylabBottomLeft = "variant", yylabBottomRight = "percentage",
  xxlab = "time")
```

Arguments

`data`	see `?plotY` for details.
`extThreshold`	see `?extractDomVariant` or `?extractNoveltyVariant` for details.
`fileout`	see `?plotY` for details.
`w`	see `?plotY` for details.
`h`	see `?plotY` for details.
`colours`	see `?plotY` for details.
`xiObs`	see `?plotY` for details.
`xfObs`	see `?plotY` for details.
`mainTopLeft`	title for top-left subplot.
`mainTopRight`	title for top-right subplot.
`mainBottomLeft`	title for bottom-left subplot.
`mainBottomRight`	title for bottom-right subplot.
`yylabTopLeft`	tile for top-left subplot y-axis.
`yylabTopRight`	tile for top-right subplot y-axis.

yylabBottomLeft
: tile for bottom-left subplot y-axis..

yylabBottomRight
: tile for bottom-right subplot y-axis.

xxlab
: see ?plotY for details.

---

plotPathogenPrevalence *Plot total prevalence of the pathogen.*

Description

Makes 2 plots, one with the full time-series, another zooming into a portion of the simulation. Can export to a PDF file.

Usage

```
plotPathogenPrevalence(data, fileout = NA, w = 7, h = 3, colour = NULL,
  xiObs = 0.9, xfObs = 1, ymax = NULL, mainLeft = "full time-series",
  mainRight = "zoomed-in time-series",
  yylab = "pathogen prevalence (proportion)", xxlab = "time",
  method = "host")
```

Arguments

data
: see ?plotY for details.

fileout
: see ?plotY for details.

w
: see ?plotY for details.

h
: see ?plotY for details.

colour
: see ?plotY for details.

xiObs
: see ?plotY for details.

xfObs
: see ?plotY for details.

ymax
: see ?plotY for details.

mainLeft
: see ?plotY for details.

mainRight
: see ?plotY for details.

yylab
: see ?plotY for details.

xxlab
: see ?plotY for details.

method
: a character string to choose the method to be used. Methods are "host" and "strain" and relate to how total prevalence of the pathogen is estimated. When method="host" the function extractPathogenPrevByStrain is called and when method="strain" the function extractPathogenPrevByHost is called. See the help of each function for how they differ. Estimations are expected to be very similar using the two methods.

`plotSeqsOfDomVariant` *Plot the sequences of the dominanting strains in time.*

Description

Makes 2 plots. Left subplot presents the simulated time series of the Y classes with the `extThreshold` highlighted with a dashed black line. Right subplot presents the loci and alleles of the dominating strains in time. In this plot, each locus is in a row and each column presents the alleles by colour (each possible allele per locus has an unique colour). The number of available colours equals the maximum of available alleles in all loci. Can create a PDF file.

Usage

```
plotSeqsOfDomVariant(data, extThreshold, epiStruc, fileout = NA, w = 7 *
  1.5, h = 1.5 * 2.5, colours = NULL, xiObs = 0, xfObs = 1,
  mainLeft = "full time-series",
  mainRight = "allelic info of dominant variants",
  yylabLeft = "proportion infected", yylabRight = "allele at loci L",
  xxlab = "time")
```

Arguments

`data`	see `?plotY` for details.
`extThreshold`	see `?extractDomVariant` or `?extractNoveltyVariant` for details.
`epiStruc`	desired antigenic structure for the simulation. See `?runMANTIS` for details.
`fileout`	see `?plotY` for details.
`w`	see `?plotY` for details.
`h`	see `?plotY` for details.
`colours`	see `?plotY` for details.
`xiObs`	see `?plotY` for details.
`xfObs`	see `?plotY` for details.
`mainLeft`	title for left subplot.
`mainRight`	title for right subplot.
`yylabLeft`	tile for top-left subplot y-axis.
`yylabRight`	tile for top-right subplot y-axis.
`xxlab`	see `?plotY` for details.

`plotW` *Plot the cross-reactive (W) time-series of all strains in the system.*

Description

Makes 2 plots, one with the full time-series, another zooming into a portion of the simulation. Can export to a PDF file.

Usage

```
plotW(data, fileout = NA, w = 7, h = 3, colours = NULL, xiObs = 0.9,
  xfObs = 1, markStrains = NULL, ymax = NULL,
  mainLeft = "full time-series", mainRight = "zoomed-in time-series",
  yylab = "proportion cross-reactive", xxlab = "time")
```

Arguments

`data`	see `?plotY` for details.
`fileout`	see `?plotY` for details.
`w`	see `?plotY` for details.
`h`	see `?plotY` for details.
`colours`	see `?plotY` for details.
`xiObs`	see `?plotY` for details.
`xfObs`	see `?plotY` for details.
`markStrains`	see `?plotY` for details.
`ymax`	see `?plotY` for details.
`mainLeft`	see `?plotY` for details.
`mainRight`	see `?plotY` for details.
`yylab`	see `?plotY` for details.
`xxlab`	see `?plotY` for details.

`plotWDiversity` *Plot diversity measures for the cross-reactive (W) classes.*

Description

Makes 2 plots, one with the full time-series, another zooming into a portion of the simulation. Can create a PDF file.

Usage

```
plotWDiversity(data, fileout = NA, w = 7, h = 3, colour = NULL,
  xiObs = 0.9, xfObs = 1, ymax = NULL, mainLeft = "full time-series",
  mainRight = "zoomed-in time-series", yylab = "diversity",
  xxlab = "time", method = "shannon")
```

Arguments

data	see ?plotY for details.
fileout	see ?plotY for details.
w	see ?plotY for details.
h	see ?plotY for details.
colour	see ?plotY for details.
xiObs	see ?plotY for details.
xfObs	see ?plotY for details.
ymax	see ?plotY for details.
mainLeft	see ?plotY for details.
mainRight	see ?plotY for details.
yylab	see ?plotY for details.
xxlab	see ?plotY for details.
method	character string for the method to use for diversity. See ?calcDiversity.

---

plotY *Plot the infectious (Y) time-series of all strains in the system.*

Description

Make 2 plots, opne with the full time-series, another zooming into a portion of the simulation. Can create a PDF file.

Usage

```
plotY(data, fileout = NA, w = 7, h = 3, colours = NULL, xiObs = 0.9,
  xfObs = 1, markStrains = NULL, ymax = NULL,
  mainLeft = "full time-series", mainRight = "zoomed-in time-series",
  yylab = "proportion infected", xxlab = "time", addLegend = FALSE)
```

Arguments

data	the list as obtained from running the simulation. See ?runMANTIS for details.
fileout	the desired filename for the resulting PDF plot, if not given (NA) the plot goes to screen
w	the width of the PDF plot.
h	the height of the PDF plot.
colours	the set of colours to be used for the existing strains. If length(colours)<nStrains, colours are reused (nStrains is the number of strains in data). If not defined by the user, the rainbow collection is used.
xiObs	the starting proportion of time used to zoom into the time-series. For example xiObs=0.5 will start zooming from the middle of the time-series.
xfObs	the end proportion of time used to zoom into the time-series.
markStrains	list of strain IDs (numbers), used to highlight the given strains in the plot. The highlight will consist of an increase in the thickness of the lines and, when length(colours) is 2, the first colour is used for these strains and the second colour for the remaining.
ymax	numeric value for the maximum to be represented in the Y axis.
mainLeft	main title for the left plot.
mainRight	main title for the right plot.
yylab	title for the y-axis.
xxlab	title for the x-axis.
addLegend	boolean flag whether to add a legend to the time series plot.

---

plotYDiversity *Plot diversity measures for the infectious (Y) classes.*

Description

Makes two plots, one with the full time-series, another zooming into a portion of the simulation. Can create a PDF file.

Usage

```
plotYDiversity(data, fileout = NA, w = 7, h = 3, colour = NULL,
  xiObs = 0.9, xfObs = 1, ymax = NULL, mainLeft = "full time-series",
  mainRight = "zoomed-in time-series", yylab = "diversity",
  xxlab = "time", method = "shannon")
```

Arguments

data	see ?plotY for details.
fileout	see ?plotY for details.
w	see ?plotY for details.
h	see ?plotY for details.
colour	see ?plotY for details.
xiObs	see ?plotY for details.
xfObs	see ?plotY for details.
ymax	see ?plotY for details.
mainLeft	see ?plotY for details.
mainRight	see ?plotY for details.
yylab	see ?plotY for details.
xxlab	see ?plotY for details.
method	character string for the method to use for diversity. See ?calcDiversity.

---

plotYRelPrevalence *Plot the relative-prevalence of the infectious (Y) time-series for all strains in the system.*

Description

Makes 2 plots, one with the full time-series, another zooming into a portion of the simulation. Can create a PDF file.

Usage

```
plotYRelPrevalence(data, fileout = NA, w = 7, h = 3, colours = NULL,
  xiObs = 0.9, xfObs = 1, markStrains = NULL, ymax = NULL,
  mainLeft = "full time-series", mainRight = "zoomed-in time-series",
  yylab = "proportion infected (relative prevalence)", xxlab = "time",
  method = "host")
```

Arguments

data	see ?plotY for details.
fileout	see ?plotY for details.
w	see ?plotY for details.
h	see ?plotY for details.
colours	see ?plotY for details.
xiObs	see ?plotY for details.
xfObs	see ?plotY for details.

markStrains	see ?plotY for details.
ymax	see ?plotY for details.
mainLeft	see ?plotY for details.
mainRight	see ?plotY for details.
yylab	see ?plotY for details.
xxlab	see ?plotY for details.
method	a character string to choose the method to be used for the estimation of the pathogen's total prevalence. Methods are "host" and "strain" and relate to how total prevalence of the pathogen is estimated. When method="host" the function extractPathogenPrevByStrain is called and when method="strain" the function extractPathogenPrevByHost is called. See the help of each function for how they differ. Estimations are expected to be very similar using the two methods.

---

plotZ *Plot the specific-immunity (Z) time-series of all strains in the system.*

Description

Makes two plots, one with the full time-series, another zooming into a portion of the simulation. Can create a PDF file.

Usage

```
plotZ(data, fileout = NA, w = 7, h = 3, colours = NULL, xiObs = 0.9,
  xfObs = 1, markStrains = NULL, ymax = NULL,
  mainLeft = "full time-series", mainRight = "zoomed-in time-series",
  yylab = "proportion with specific-immunity", xxlab = "time")
```

Arguments

data	see ?plotY for details.
fileout	see ?plotY for details.
w	see ?plotY for details.
h	see ?plotY for details.
colours	see ?plotY for details.
xiObs	see ?plotY for details.
xfObs	see ?plotY for details.
markStrains	see ?plotY for details.
ymax	see ?plotY for details.
mainLeft	see ?plotY for details.
mainRight	see ?plotY for details.
yylab	see ?plotY for details.
xxlab	see ?plotY for details.

`plotZDiversity` *Plot diversity measures for the specific-immunity (Z) classes.*

Description

Makes 2 plots, one with the full time-series, another zooming into a portion of the simulation. Can create a PDF file.

Usage

```
plotZDiversity(data, fileout = NA, w = 7, h = 3, colour = NULL,
  xiObs = 0.9, xfObs = 1, ymax = NULL, mainLeft = "full time-series",
  mainRight = "zoomed-in time-series", yylab = "diversity",
  xxlab = "time", method = "shannon")
```

Arguments

`data`	see `?plotY` for details.
`fileout`	see `?plotY` for details.
`w`	see `?plotY` for details.
`h`	see `?plotY` for details.
`colour`	see `?plotY` for details.
`xiObs`	see `?plotY` for details.
`xfObs`	see `?plotY` for details.
`ymax`	see `?plotY` for details.
`mainLeft`	see `?plotY` for details.
`mainRight`	see `?plotY` for details.
`yylab`	see `?plotY` for details.
`xxlab`	see `?plotY` for details.
`method`	character string for the method to use for diversity. See `?calcDiversity`.

`readingMANTIS` *Print a list of reading suggestions by the Multilocus ANTIgenic Simulator (MANTIS) team.*

Description

This function prints a list of the recommended literature on the epidemiological framework behind MANTIS.

Usage

```
readingMANTIS()
```

runInvasionWithOneBetaChange

Run the Multilocus ANTIgenic Simulator (MANTIS) under changes in the beta parameter.

Description

Function that runs the multilocus model assuming that a change in the beta of one or more strains will happen at a given point in time. Here, only the parameters not included in a normal call to `runMANTIS` are described. For full details on a normal run of MANTIS see `?runMANTIS` for details.

Usage

```
runInvasionWithOneBetaChange(epiStruc, tMax, tObsPer, tInt, initCondY, beta,
  gamma, sigma, mu, tBetaChange, changedBeta, changedStrain)
```

Arguments

`tBetaChange`	a numeric value for the time step at which betas should be changed in value, as given by `changedBetas`.
`changedBeta`	is the numeric value of beta to be attributed to `changedStrain`. See `?runMANTIS` for details on the beta parameter.
`changedStrain`	is the numeric id (numeber) of the strain to change the neta as in `changedBeta`. See `?runMANTIS` for details on the beta parameter.

Value

a numeric array with the population state, with the class order `Zi,Wi,Yi`.

runMANTIS

Run the Multilocus ANTIgenic Simulator (MANTIS)

Description

Function that runs the multilocus model. A deterministic ordinary-differential equations system (ODE) is solved. For more details on the conceptual framework, including variables and biological parameters, a summary can be found in the manual. References to the literature can also be found by running the function `readingMANTIS()`.

Usage

```
runMANTIS(epiStruc, tMax, tObsPer, tStep, initCondY, beta, gamma, sigma, mu,
  epsilon = 0, initCondZ = NA, initCondW = NA)
```

Arguments

epiStruc	desired antigenic structure for the simulation. It should be an array of integers specifying the number of alleles per locus. Its length determines the number of loci and each integer the number of alleles per locus. The minimum number of loci allowed is 2 (`length(epiStruc)>=2`) and therefore the minimum number of strains in the system is also 2. If the user wants to simulate 2 strains, the second locus can be set to have only one allele, that is `epiStruct=c(2,1)`.
tMax	the time at which the solver will stop solving the ODE system.
tObsPer	an integer specifying the periodicity at which the solution is recorded. It may be used to reduce the size of the output variable considerably.
tStep	time step for the fixed-step ODE solver. If the dynamics of the system appear inconsistent, this should be reduced.
initCondY	initial conditions for the infectious classes (`Yi`). This is a collection of numbers of length equal to the number of strains in the system. Its sum needs to be smaller than 1, as initial conditions represent proportions.
beta	numeric value or collection of numbers. If a single value is given, all strains will share beta=value. If a collection of numbers is given, it is required that the length of the collection be equal to the number of strains; in this case each strain indexes the collection and can have a different value for beta.
gamma	number representing cross-protection given by challenge of strains that share alleles. This needs to be between 0 and 1 (inclusive), for which 0 confers no protection and 1 full protection.
sigma	number representing the recovery rate (also known as the loss of infection rate). `1/sigma` is the infectious period.
mu	number representing the death rate of all individuals in the system. `1/mu` is the mean life-span.
epsilon	number representing the strength of seasonality changes in `beta`, defaults to zero. `epsilon=0` equates to no seasonality changes. If `epsilon>0`, `beta` suffers from yearly seasonal changes. Seasonality is modelled via a sinusoidal signal, defined as `beta*(1.0+epsilon*pow(sin(PI*t),6))`. This means that `beta` is the minimum (out of season) and `beta*(1+epsilon)` the peak of the season. NOTE: seasonality expects model parameters to be defined per year, as modelled oscillations are per year. Seasonality parameters can be tested using the function `testSeasonality`, see ?testSeasonality.
initCondZ	initial conditions for the infectious classes (`Zi`). This is a collection of numbers of length equal to the number of strains in the system. Its sum needs to be smaller than 1, as initial conditions represent proportions. This is optional and will be assumed zero if not given.
initCondW	initial conditions for the infectious classes (`Wi`). This is a collection of numbers of length equal to the number of strains in the system. Its sum needs to be smaller than 1, as initial conditions represent proportions. This is optional and will be assumed zero if not given.

Value

A 'list' of size 4. The first element is 'time'; the second element is the resulting 'time series'; the third element contains the information about 'shared epitopes' between strains; and the fourth element contains the 'allelic matrix' which describes the allelic combination that defines each strain in the system.

'time' is an array of numbers representing the time steps of the simulation's solution, its length is determined by `tMax/tStep`.

'time series' is a matrix with `tMax/tStep` rows and `3*nStrains` columns (nStrains is the number of strains). Each column of 'time series' represents an epidemiological class (equation) in the system. Columns `1:nStrains` are the classes `Zi` for each `strain i`, representing individuals with specific-immunity to `strain i`. Columns `(nStrains+1):(nStrains*2)` are the classes `Wi` for each `strain i`, representing cross-reactive individuals to `strain i`. Columns `(nStrains*2+1):(nStrains*3)` are the classes `Yi` for each `strain i`, representing infectious individuals with `strain i`. For more details on the epidemiological framework, including variables `Z, W, Y` and biological parameters, see references in the output of function `readingMANTIS()` .

'shared epitopes' is a matrix with N rows and M columns. N equals the number of strains and M the number of possible strains to which each strain can share an allele. Each row `i` contains a set of integer numbers identifying which other strains share an allele with strain `i`. For example, for an antigenic structure `epiStruct=c(2,2)`, giving 4 strains, 'shared epitopes' will have 4 rows in which the first will contain `"1 2 3"`, meaning that strain 1 shares alleles with strain 1, 2 and 3.

'allelic matrix' is a matrix with N rows and L columns. N equals the number of strains and L the number of loci. For simplification, alleles are defined by integer numbers from `1....Al`, with `Al` being the maximum diversity allowed at each loci `l`. Each row represents the combination of alleles that define each strain.

`testSeasonality` *Test Seasonality.*

Description

Can be used to test parameters for seasonal forcing in transmission. See ?runMANTIS for details on seasonality, beta and epsilon.

Usage

```
testSeasonality(epsilon, beta, time = seq(1, 50, 0.025))
```

Arguments

`epsilon`	the strenght of the seasonal signal.
`beta`	the (minimum) beta for transmission.
`time`	time vector for the test, defaults to 50 years.

`withSharedEpitopes` *Returns the set of strains sharing alleles with strain* `i`.

Description

The `'shared epitopes'` matrix is used to return a list of strain numbers that share at least one allele with strain `i` given as argument `strain`. The matrix is the third element of the list returned by `runMANTIS` after running a simulation. For details on the `'shared epitopes'` matrix see `?runMANTIS`.

Usage

```
withSharedEpitopes(data, strain = NULL)
```

Arguments

`data`	the list as obtained from running the simulation. See `?runMANTIS` for details.
`strain`	a strain number. Used to query the `'shared epitopes'` matrix to find the set of strains that shared alleles with `strain`. If `strain` is NULL, the entire `'shared epitopes'` matrix is returned.

Value

a numeric array with strain numbers.