Package 'MGDrivE2'

May 7, 2026

Type Package

Title Mosquito Gene Drive Explorer 2

Version 2.1.1

Maintainer Sean L. Wu <slwood89@gmail.com>

URL https://marshalllab.github.io/MGDrivE/docs_v2/index.html, https://www.marshalllab.com/

BugReports https://github.com/MarshallLab/MGDrivE/issues

Description A simulation modeling framework which significantly extends capabilities from the 'MGDrivE' simulation package via a new mathematical and computational framework based on stochastic Petri nets.
For more information about 'MGDrivE', see our publication: Sánchez et al. (2019) <doi:10.1111/2041-210X.13318>
Some of the notable capabilities of 'MGDrivE2' include: incorporation of human populations, epidemiological dynamics, time-varying parameters, and a continuous-time simulation framework with various sampling algorithms for both deterministic and stochastic interpretations. 'MGDrivE2' relies on the genetic inheritance structures provided in package 'MGDrivE', so we suggest installing that package initially.

License GPL-3

Encoding UTF-8

ByteCompile true

LazyData true

Depends R (>= 3.1.0)

Imports Matrix, deSolve, statmod

Suggests MGDrivE, knitr, rmarkdown, markdown, ggplot2

VignetteBuilder knitr

RoxygenNote 7.3.2

NeedsCompilation no

Author Sean L. Wu [aut, cre],
Jared B. Bennett [aut],
Héctor Manuel Sánchez Castellanos [ctb],

Tomás M. León [ctb],
Andrew J. Dolgert [ctb],
John M. Marshall [aut],
Agastya Mondal [aut]

Repository CRAN

Date/Publication 2025-09-14 11:30:24 UTC

Contents

add_interventions . . . 3
batch_migration . . . 4
calc_move_rate . . . 5
convert_prevalence_to_eir . . . 6
equilibrium_Imperial_decoupled . . . 7
equilibrium_Imperial_decoupled_human . . . 7
equilibrium_lifeycle . . . 8
equilibrium_SEI_decoupled_human . . . 10
equilibrium_SEI_decoupled_mosy . . . 10
equilibrium_SEI_Imperial . . . 13
equilibrium_SEI_SEIR . . . 15
equilibrium_SEI_SIS . . . 18
get_shape . . . 20
human_Imperial_ODE . . . 21
imperial_model_param_list_create . . . 22
make_Q_Imperial . . . 26
make_Q_SEI . . . 27
movement_prob2rate . . . 28
mu_ts . . . 28
sim_trajectory_base_CSV . . . 29
sim_trajectory_base_CSV_decoupled . . . 30
sim_trajectory_base_R . . . 31
sim_trajectory_base_R_decoupled_Imperial . . . 32
sim_trajectory_base_R_decoupled_SIS . . . 33
sim_trajectory_CSV . . . 34
sim_trajectory_CSV_decoupled . . . 36
sim_trajectory_R . . . 38
sim_trajectory_R_decoupled . . . 39
solve_muAqua . . . 41
split_aggregate_CSV . . . 42
split_aggregate_CSV_decoupled . . . 45
spn_hazards . . . 47
spn_hazards_decoupled . . . 48
spn_Post . . . 49
spn_Pre . . . 50
spn_P_epiSEIR_network . . . 51
spn_P_epiSEIR_node . . . 51
spn_P_epiSIS_network . . . 52

spn_P_epiSIS_node . . . 53
spn_P_epi_decoupled_node . . . 53
spn_P_lifecycle_network . . . 54
spn_P_lifecycle_node . . . 55
spn_S . . . 55
spn_T_epiSEIR_network . . . 56
spn_T_epiSEIR_node . . . 57
spn_T_epiSIS_network . . . 58
spn_T_epiSIS_node . . . 59
spn_T_epi_decoupled_node . . . 60
spn_T_lifecycle_network . . . 61
spn_T_lifecycle_node . . . 62
step_CLE . . . 63
step_DM . . . 64
step_ODE . . . 64
step_ODE_decoupled . . . 65
step_PTS . . . 66
step_PTS_decoupled . . . 67
summarize_eggs_geno . . . 68
summarize_eggs_stage . . . 68
summarize_females . . . 69
summarize_females_epi . . . 70
summarize_humans_epiImperial . . . 70
summarize_humans_epiSEIR . . . 71
summarize_humans_epiSIS . . . 72
summarize_larvae_geno . . . 72
summarize_larvae_stage . . . 73
summarize_males . . . 74
summarize_pupae_geno . . . 74
summarize_pupae_stage . . . 75
summarize_stats_CSV . . . 76
summarize_stats_CSV_decoupled . . . 77
track_hinf . . . 79

Index **80**

`add_interventions` *This set of functions modifies mosquito life history parameters in the presence of adult interventions - indoor residual spraying (IRS) and insecticide treated nets (ITN) This is based on the work of Le Menach et al (2007) and Griffin et al (2010). We vary three parameters in the presence of interventions: Egg laying rate (beta) Adult mortality (muF) Mosquito biting rate (av0)*

Description

This set of functions modifies mosquito life history parameters in the presence of adult interventions - indoor residual spraying (IRS) and insecticide treated nets (ITN) This is based on the work of Le Menach et al (2007) and Griffin et al (2010). We vary three parameters in the presence of interventions: Egg laying rate (beta) Adult mortality (muF) Mosquito biting rate (av0)

Usage

```
add_interventions(params, IRS_cov, LLIN_cov)
```

Arguments

params	a named list of parameters
IRS_cov	proportion of humans in the node receiving IRS
LLIN_cov	proportion of humans in the node receiving LLIN

Value

a vector of the equilibrium number of females in each SEI stage

---

batch_migration *Sample Batch Migration Events*

Description

Sample batch migration events for simulation given rates of occurance and probability of destination for each patch. Batch migration can be simulated for the aquatic life stages (eggs, larvae, pupae), adult females, and/or adult males. To simulate batch migration, each life stage needs all 3 of its arguments specified. If any arguments are left unspecified (NULL), batch migration for that life stage will not be sampled. The output of this function should be passed to sim_trajectory_R or sim_trajectory_CSV as the argument batch. Calls the internal function batch_migration_stage.

Usage

```
batch_migration(
  SPN_P,
  tmax,
  ELPrates = NULL,
  ELPmove = NULL,
  ELPprob = NULL,
  Frates = NULL,
  Fmove = NULL,
  Fprob = NULL,
  Mrates = NULL,
  Mmove = NULL,
  Mprob = NULL,
  stage = NULL
)
```

Arguments

SPN_P	places of the SPN
tmax	maximum time of the simulation
ELPrates	rate at which aquatic stage batch migration occurs for each node (nodes without mosquitoes should be set to NaN or NA)
ELPmove	movement matrix for destinations of aquatic stage batch migration events (diagonal will be set to zero and off-diagonal elements normalized)
ELPprob	probability for each individual to be chosen for aquatic stage batch migration events (must be same length as ELPrates)
Frates	rate at which adult female batch migration occurs for each node (nodes without mosquitoes should be set to NaN or NA)
Fmove	movement matrix for destinations of adult female batch migration events (diagonal will be set to zero and off-diagonal elements normalized)
Fprob	probability for each individual to be chosen for adult female batch migration events (must be same length as Frates)
Mrates	rate at which adult male batch migration occurs for each node (nodes without mosquitoes should be set to NaN or NA)
Mmove	movement matrix for destinations of adult male batch migration events (diagonal will be set to zero and off-diagonal elements normalized)
Mprob	probability for each individual to be chosen for adult male batch migration events (must be same length as Mrates)
stage	either NULL or "E", "L", or "P". If not NULL and migration for aquatic stages is specified by ELPrates, only the aquatic stage specified here will move

Value

vector of lists describing all batch migration events, segmented by life stage.

---

calc_move_rate *Calculate Outbound Movement Rate*

Description

Given P, the cumulative probability of moving before dying, and mu, the daily mortality rate, calculate the movement rate gamma to get P. The equation comes from integrating the competing risks and solving for gamma.

Usage

```
calc_move_rate(mu, P)
```

Arguments

mu	daily mortality rate
P	cumulative probability to move before dying

Value

numeric probability of movement

Examples

```
# parameters, see vignette MGDrivE2: One Node Lifecycle Dynamics
theta <- list(qE = 1/4, nE = 2, qL = 1/3, nL = 3, qP = 1/6, nP = 2,
              muE = 0.05, muL = 0.15, muP = 0.05, muF = 0.09, muM = 0.09,
              beta = 16, nu = 1/(4/24) )

# lets say a 70% chance to move over the entire lifespan
rMoveRate <- calc_move_rate(mu = theta$muF, P = 0.70)
```

---

convert_prevalence_to_eir

Generally, pathogen prevalence is a more accesible metric for users, but the Imperial equilibrium function requires an annual EIR. This function converts a given pathogen prevalence to an EIR

Description

Generally, pathogen prevalence is a more accesible metric for users, but the Imperial equilibrium function requires an annual EIR. This function converts a given pathogen prevalence to an EIR

Usage

```
convert_prevalence_to_eir(prevalence, age_vector, ft, params)
```

Arguments

prevalence	desired prevalence value
age_vector	age distribution of the population
ft	proportion treated
params	entomological and epidemiological parameters

Value

a vector of the equilibrium number of humans in each SIS stage

`equilibrium_Imperial_decoupled`

This function calculates the human and mosquito equilibrium values for the decoupled Imperial model. Currently this only works in one node.

Description

This function calculates the human and mosquito equilibrium values for the decoupled Imperial model. Currently this only works in one node.

Usage

```
equilibrium_Imperial_decoupled(age_vector, ft, eir, theta, cube, spn_P)
```

Arguments

`age_vector`	age structure of population (see vignette for example)
`ft`	proportion of population seeking treatment
`eir`	desired annual EIR
`theta`	parameters
`cube`	inheritance cube
`spn_P`	places of the stochastic petri net

Value

a matrix of the equilibrium number of humans in each Imperial stage by age, and immunity. mosquito equilibrium values, and full theta vector

`equilibrium_Imperial_decoupled_human`

This function calculates the human equilibrium values for the decoupled Imperial model. Requires the age structure of the population Currently this only works in one node.

Description

This function calculates the human equilibrium values for the decoupled Imperial model. Requires the age structure of the population Currently this only works in one node.

Usage

```
equilibrium_Imperial_decoupled_human(age_vector, ft, EIR, model_param_list)
```

Arguments

age_vector	age structure of population (see vignette for example)
ft	proportion of population seeking treatment
EIR	desired annual EIR
model_param_list	parameters for the

Value

a matrix of the equilibrium number of humans in each Imperial stage by age, and immunity

---

equilibrium_lifeycle *Calculate Equilibrium for Lifecycle Model (Logistic or Lotka-Volterra)*

Description

This function calculates deterministic equilibria for the mosquito lifecycle model.

Usage

```
equilibrium_lifeycle(
  params,
  NF,
  phi = 0.5,
  log_dd = TRUE,
  spn_P,
  pop_ratio_Aq = NULL,
  pop_ratio_F = NULL,
  pop_ratio_M = NULL,
  cube
)
```

Arguments

params	a named list of parameters (see details)
NF	vector of female mosquitoes at equilibrium, for every population in the environment
phi	sex ratio of mosquitoes at emergence
log_dd	Boolean: TRUE implies logistic density dependence, FALSE implies Lotka-Volterra model
spn_P	the set of places (P) (see details)
pop_ratio_Aq	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_F	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_M	May be empty; if not, a named vector or matrix. (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

Equilibrium can be calculated using one of two models: classic logistic dynamics or following the Lotka-Volterra competition model. This is determined by the parameter `log_dd`, and it changes elements of the return list: `K` is returned for logistic dynamics, or `gamma` is returned for Lotka-Volterra dynamics.

The places (`spn_P`) object is generated from one of the following: `spn_P_lifecycle_node`, `spn_P_lifecycle_network`, `spn_P_epiSIS_node`, `spn_P_epiSIS_network`, `spn_P_epiSEIR_node`, or `spn_P_epiSEIR_network`.

The initial population genotype ratios are set by supplying the `pop_ratio_Aq`, `pop_ratio_F`, and `pop_ratio_M` values. The default value is NULL, and the function will use the wild-type alleles provided in the `cube` object. However, one can supply several different objects to set the initial genotype ratios. All genotypes provided must exist in the `cube` (this is checked by the function). If a single, named vector is provided, then all patches will be initialized with the same ratios. If a matrix is provided, with the number of columns (and column names) giving the initial genotypes, and a row for each patch, each patch can be set to a different initial ratio. The three parameters do not need to match each other.

The `params` argument supplies all of the ecological parameters necessary to calculate equilibrium values. This is used to set the initial population distribution and during the simulation to maintain equilibrium. `params` must include the following named parameters:

- `qE`: inverse of mean duration of egg stage
- `nE`: shape parameter of Erlang-distributed egg stage
- `qL`: inverse of mean duration of larval stage
- `nL`: shape parameter of Erlang-distributed larval stage
- `qP`: inverse of mean duration of pupal stage
- `nP`: shape parameter of Erlang-distributed pupal stage
- `muE`: egg mortality
- `muL`: density-independent larvae mortality
- `muP`: pupae mortality
- `muF`: adult female mortality
- `muM`: adult male mortality
- `beta`: egg-laying rate, daily
- `nu`: mating rate of unmated females

The return list contains all of the `params` parameters, along with the density-dependent parameter, either `K` or `gamma`. These are the parameters necessary later in the simulations. This was done for compatibility with `equilibrium_SEI_SIS`, which requires several extra parameters not required further in the simulations.

For equilibrium with epidemiological parameters, see `equilibrium_SEI_SIS`. For equilibrium with latent humans (SEIR dynamics), see `equilibrium_SEI_SEIR`.

Value

a list with 3 elements: `init` a matrix of equilibrium values for every life-cycle stage, `params` a list of parameters for the simulation, `M0` a vector of initial conditions

`equilibrium_SEI_decoupled_human`

This function calculates the equilibrium values for the decoupled SIS human states. Currently this only works in one node.

Description

This function calculates the equilibrium values for the decoupled SIS human states. Currently this only works in one node.

Usage

```
equilibrium_SEI_decoupled_human(params)
```

Arguments

`params`	a named list of parameters (see details)

Value

a vector of the equilibrium number of humans in each SIS stage

`equilibrium_SEI_decoupled_mosy`

Calculate Equilibrium for Decoupled Mosquito SEI model. Human states will be handled separately.

Description

Given prevalence of disease in humans (modeled as an SIS: Susceptible-Infected-Susceptible process with birth and death) and entomological parameters of transmission, this function calculates the quasi-stationary distribution of adult female mosquitoes across SEI (Susceptible-Exposed-Infectious) stages, allowing for Erlang distributed E stage.

Usage

```
equilibrium_SEI_decoupled_mosy(
  params,
  node_list = "b",
  NF = NULL,
  phi = 0.5,
  NH = NULL,
  log_dd = TRUE,
  spn_P,
  pop_ratio_Aq = NULL,
```

```
  pop_ratio_F = NULL,
  pop_ratio_M = NULL,
  pop_ratio_H = 1,
  cube
)
```

Arguments

params	a named list of parameters (see details)
node_list	a character vector specifying what type of nodes to create; (m = a node with only mosquitoes, h = a node with only humans, b = a node with both humans and mosquitoes)
NF	vector of female mosquitoes at equilibrium, for mosquito-only nodes
phi	sex ratio of mosquitoes at emergence
NH	vector of humans at equilibrium, for human-only nodes
log_dd	Boolean: TRUE implies logistic density dependence, FALSE implies Lotka-Volterra model
spn_P	the set of places (P) (see details)
pop_ratio_Aq	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_F	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_M	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_H	Prevalence in human-only nodes
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

This function handles 3 types of nodes: Human only, mosquito only, and nodes with both. These nodes are set using the node_list parameter. Mosquito-only node equilibrium calls equilibrium_lifeycle, which follows one of two models: classic logistic dynamics or the Lotka-Volterra competition model. This is determined by the parameter log_dd, and it changes elements of the return list: K is returned for logistic dynamics, or gamma is returned for Lotka-Volterra dynamics. This is parameterized with the NF parameter to define the adult female numbers. This parameter only needs to be supplied if there are mosquito-only nodes.

Human-only nodes don't require any equilibrium calculations. These nodes use the NH and pop_ratio_H to set adult human populations and infection rates in nodes. These two parameters only need to be supplied if there are human-only nodes.

For human and mosquito nodes, this function calls make_Q_SEI to construct the infinitesimal generator matrix which is used to solve for the quasi-stationary (stochastic) or equilibrium (deterministic) distribution of mosquitoes over stages. Parameters are provided by params.

For information on the method used to solve this distribution, see section "3.1.3 Nonsingularity of the Subintensity Matrix" of:

- Bladt, Mogens, and Bo Friis Nielsen. Matrix-exponential distributions in applied probability. Vol. 81. New York: Springer, 2017.

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The initial population genotype ratios are set by supplying the pop_ratio_Aq, pop_ratio_F, and pop_ratio_M values. The default value is NULL, and the function will use the wild-type alleles provided in the cube object. However, one can supply several different objects to set the initial genotype ratios. All genotypes provided must exist in the cube (this is checked by the function). If a single, named vector is provided, then all patches will be initialized with the same ratios. If a matrix is provided, with the number of columns (and column names) giving the initial genotypes, and a row for each patch, each patch can be set to a different initial ratio. The three parameters do not need to match each other.

The params argument supplies all of the ecological and epidemiological parameters necessary to calculate equilibrium values. This is used to set the initial population distribution and during the simulation to maintain equilibrium. This params must include the following named parameters, noted as being the same as lifecycle parameters, or new for the epidemiological equilibrium

- **(Lifecycle parameters)**
  - qE: inverse of mean duration of egg stage
  - nE: shape parameter of Erlang-distributed egg stage
  - qL: inverse of mean duration of larval stage
  - nL: shape parameter of Erlang-distributed larval stage
  - qP: inverse of mean duration of pupal stage
  - nP: shape parameter of Erlang-distributed pupal stage
  - muE: egg mortality
  - muL: density-independent larvae mortality
  - muP: pupae mortality
  - muF: adult female mortality
  - muM: adult male mortality
  - beta: egg-laying rate, daily
  - nu: mating rate of unmated females
- **(Epidemiological parameters)**
  - NH: number of humans, can be a vector
  - X: prevalence in humans, can be a vector
  - NFX: number of female mosquitoes, only required if any prevalence (X) is zero
  - b: mosquito to human transmission efficiency, can be a vector
  - c: human to mosquito transmission efficiency, can be a vector
  - r: rate of recovery in humans (1/duration of infectiousness)
  - muH: death rate of humans (1/avg lifespan)
  - f: rate of blood feeding
  - Q: human blood index
  - qEIP: related to scale parameter of Gamma distributed EIP (1/qEIP is mean length of EIP)
  - nEIP: shape parameter of Gamma distributed EIP

The return list contains all of the parameters necessary later in the simulations.

For equilibrium without epidemiological parameters, see `equilibrium_lifeycle`. For equilibrium with latent humans (SEIR dynamics), see `equilibrium_SEI_SEIR`.

For examples of using this function, see: `vignette("lifecycle-node", package = "MGDrivE2")`

Value

a vector of the equilibrium number of females in each SEI stage

---

`equilibrium_SEI_Imperial` *Calculate Equilibrium for Mosquito SEI - Human Imperial Model*

Description

In decoupled sampling, human states are handled separately from mosquito states. The function `equilibrium_Imperial_decoupled_human` calculates the distribution of humans at equilibrium required for the Imperial model of malaria transmission. Here we use parameters from that model to calculate the equilibrium states of Susceptible-Exposed-Infectious (SEI) female mosquitoes

Usage

```
equilibrium_SEI_Imperial(
  params,
  node_list = "b",
  NF = NULL,
  phi = 0.5,
  NH = NULL,
  log_dd = TRUE,
  spn_P,
  pop_ratio_Aq = NULL,
  pop_ratio_F = NULL,
  pop_ratio_M = NULL,
  pop_ratio_H = 1,
  cube
)
```

Arguments

`params`	a named list of parameters (see details)
`node_list`	list of geospatial nodes
`NF`	number of female mosquitoes
`phi`	sex ratio of mosquitoes at emergence
`NH`	vector of humans at equilibrium

log_dd	Boolean: TRUE implies logistic density dependence, FALSE implies Lotka-Volterra model
spn_P	the set of places (P) (see details)
pop_ratio_Aq	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_F	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_M	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_H	Prevalence in human-only nodes
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

Imperial model sampling is currently only supported for one-node dynamics: a single node with mosquitoes parameterized by the distribution of human states. These nodes are set using the node_list parameter. Mosquito-only node equilibrium calls equilibrium_lifeycle, which follows one of two models: classic logistic dynamics or the Lotka-Volterra competition model. This is determined by the parameter log_dd, and it changes elements of the return list: K is returned for logistic dynamics, or gamma is returned for Lotka-Volterra dynamics. This is parameterized with the NF parameter to define the adult female numbers. This parameter only needs to be supplied if there are mosquito-only nodes.

For human and mosquito nodes, this function calculates the number of SEI mosquitoes in each state.

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The initial population genotype ratios are set by supplying the pop_ratio_Aq, pop_ratio_F, and pop_ratio_M values. The default value is NULL, and the function will use the wild-type alleles provided in the cube object. However, one can supply several different objects to set the initial genotype ratios. All genotypes provided must exist in the cube (this is checked by the function). If a single, named vector is provided, then all patches will be initialized with the same ratios. If a matrix is provided, with the number of columns (and column names) giving the initial genotypes, and a row for each patch, each patch can be set to a different initial ratio. The three parameters do not need to match each other.

The params argument supplies all of the ecological and epidemiological parameters necessary to calculate equilibrium values. This is used to set the initial population distribution and during the simulation to maintain equilibrium. This params must include the following named parameters, noted as being the same as lifecycle parameters, or new for the epidemiological equilibrium

- **(Lifecycle parameters)**
  - qE: inverse of mean duration of egg stage
  - nE: shape parameter of Erlang-distributed egg stage
  - qL: inverse of mean duration of larval stage
  - nL: shape parameter of Erlang-distributed larval stage
  - qP: inverse of mean duration of pupal stage
  - nP: shape parameter of Erlang-distributed pupal stage
  - muE: egg mortality
  - muL: density-independent larvae mortality

  - muP: pupae mortality
  - muF: adult female mortality, supplied from Imperial equilibrium function
  - muM: adult male mortality, supplied from Imperial equilibrium function
  - beta: egg-laying rate, daily
  - nu: mating rate of unmated females
- **(Epidemiological parameters)**
  - NH: number of humans, can be a vector
  - FOIv: force of infection on mosquitoes, supplied from Imperial equilibrium function
  - Iv_eq: per-capita proportion of infectious mosquitoes The return list contains all of the parameters necessary later in the simulations.

Value

a vector of the equilibrium number of females in each SEI stage

---

`equilibrium_SEI_SEIR` *Calculate Equilibrium for Mosquito SEI - Human SEIR Model*

Description

Given prevalence of disease in humans (modeled as an SEIR: Susceptible-Latent-Infected-Recovered process with birth and death) and entomological parameters of transmission, this function calculates the quasi-stationary distribution of adult female mosquitoes across SEI (Susceptible-Exposed-Infectious) stages, allowing for Erlang distributed E stage.

Usage

```
equilibrium_SEI_SEIR(
  params,
  node_list = "b",
  NF = NULL,
  phi = 0.5,
  NH = NULL,
  log_dd = TRUE,
  spn_P,
  pop_ratio_Aq = NULL,
  pop_ratio_F = NULL,
  pop_ratio_M = NULL,
  pop_ratio_H = c(1, 0, 0, 0),
  cube
)
```

Arguments

params	a named list of parameters (see details)
node_list	a character vector specifying what type of nodes to create; (m = a node with only mosquitoes, h = a node with only humans, b = a node with both humans and mosquitoes)
NF	vector of female mosquitoes at equilibrium, for mosquito-only nodes
phi	sex ratio of mosquitoes at emergence
NH	vector of humans at equilibrium, for human-only nodes
log_dd	Boolean: TRUE implies logistic density dependence, FALSE implies Lotka-Volterra model
spn_P	the set of places (P) (see details)
pop_ratio_Aq	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_F	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_M	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_H	Prevalence in human-only nodes, default is all susceptible
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

This function handles 3 types of nodes: Human only, mosquito only, and nodes with both. These nodes are set using the node_list parameter. Mosquito-only node equilibrium calls equilibrium_lifeycle, which follows one of two models: classic logistic dynamics or the Lotka-Volterra competition model. This is determined by the parameter log_dd, and it changes elements of the return list: K is returned for logistic dynamics, or gamma is returned for Lotka-Volterra dynamics. This is parameterized with the NF parameter to define the adult female numbers. This parameter only needs to be supplied if there are mosquito-only nodes.

Human-only nodes don't require any equilibrium calculations. These nodes use the NH and pop_ratio_H to set adult human populations and infection rates in nodes. These two parameters only need to be supplied if there are human-only nodes. pop_ratio_H needs to be a matrix with the number of rows equal to the number of human-only patches, and 4 columns. The columns are assumed to be fractions of the population in "S", "E", "I", or "R" states, and every row must sum to 1.

For human and mosquito nodes, this function calls make_Q_SEI to construct the infinitesimal generator matrix which is used to solve for the quasi-stationary (stochastic) or equilibrium (deterministic) distribution of mosquitoes over stages. Parameters are provided by params.

For information on the method used to solve this distribution, see section "3.1.3 Nonsingularity of the Subintensity Matrix" of:

- Bladt, Mogens, and Bo Friis Nielsen. Matrix-exponential distributions in applied probability. Vol. 81. New York: Springer, 2017.

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The initial population genotype ratios are set by supplying the pop_ratio_Aq, pop_ratio_F, and pop_ratio_M values. The default value is NULL, and the function will use the wild-type alleles

provided in the cube object. However, one can supply several different objects to set the initial genotype ratios. All genotypes provided must exist in the cube (this is checked by the function). If a single, named vector is provided, then all patches will be initialized with the same ratios. If a matrix is provided, with the number of columns (and column names) giving the initial genotypes, and a row for each patch, each patch can be set to a different initial ratio. The three parameters do not need to match each other.

The params argument supplies all of the ecological and epidemiological parameters necessary to calculate equilibrium values. This is used to set the initial population distribution and during the simulation to maintain equilibrium. This params must include the following named parameters, noted as being the same as lifecycle parameters, or new for the epidemiological equilibrium

- **(Lifecycle parameters)**
  - qE: inverse of mean duration of egg stage
  - nE: shape parameter of Erlang-distributed egg stage
  - qL: inverse of mean duration of larval stage
  - nL: shape parameter of Erlang-distributed larval stage
  - qP: inverse of mean duration of pupal stage
  - nP: shape parameter of Erlang-distributed pupal stage
  - muE: egg mortality
  - muL: density-independent larvae mortality
  - muP: pupae mortality
  - muF: adult female mortality
  - muM: adult male mortality
  - beta: egg-laying rate, daily
  - nu: mating rate of unmated females
- **(Epidemiological parameters)**
  - NH: number of humans, can be a vector
  - X: SEIR prevalence in humans, can be a vector of length 4 for 1 node, or a matrix for many nodes
  - NFX: number of female mosquitoes, only required if any prevalence (X) is zero
  - b: mosquito to human transmission efficiency, can be a vector
  - c: human to mosquito transmission efficiency, can be a vector
  - r: rate of recovery in humans (1/duration of infectiousness)
  - muH: death rate of humans (1/avg lifespan)
  - f: rate of blood feeding
  - Q: human blood index
  - qEIP: related to scale parameter of Gamma distributed EIP (1/qEIP is mean length of EIP)
  - nEIP: shape parameter of Gamma distributed EIP
  - delta: inverse duration of the latent stage (E)

The return list contains all of the parameters necessary later in the simulations.

For equilibrium without epidemiological parameters, see equilibrium_lifeycle. For equilibrium without latent humans (SIS dynamics), see equilibrium_SEI_SIS.

Value

a vector of the equilibrium number of females in each SEI stage

---

`equilibrium_SEI_SIS` *Calculate Equilibrium for Mosquito SEI - Human SIS Model*

Description

Given prevalence of disease in humans (modeled as an SIS: Susceptible-Infected-Susceptible process with birth and death) and entomological parameters of transmission, this function calculates the quasi-stationary distribution of adult female mosquitoes across SEI (Susceptible-Exposed-Infectious) stages, allowing for Erlang distributed E stage.

Usage

```
equilibrium_SEI_SIS(
  params,
  node_list = "b",
  NF = NULL,
  phi = 0.5,
  NH = NULL,
  log_dd = TRUE,
  spn_P,
  pop_ratio_Aq = NULL,
  pop_ratio_F = NULL,
  pop_ratio_M = NULL,
  pop_ratio_H = 1,
  cube
)
```

Arguments

`params`	a named list of parameters (see details)
`node_list`	a character vector specifying what type of nodes to create; (m = a node with only mosquitoes, h = a node with only humans, b = a node with both humans and mosquitoes)
`NF`	vector of female mosquitoes at equilibrium, for mosquito-only nodes
`phi`	sex ratio of mosquitoes at emergence
`NH`	vector of humans at equilibrium, for human-only nodes
`log_dd`	Boolean: TRUE implies logistic density dependence, FALSE implies Lotka-Volterra model
`spn_P`	the set of places (P) (see details)
`pop_ratio_Aq`	May be empty; if not, a named vector or matrix. (see details)
`pop_ratio_F`	May be empty; if not, a named vector or matrix. (see details)

pop_ratio_M	May be empty; if not, a named vector or matrix. (see details)
pop_ratio_H	Prevalence in human-only nodes
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

This function handles 3 types of nodes: Human only, mosquito only, and nodes with both. These nodes are set using the node_list parameter. Mosquito-only node equilibrium calls equilibrium_lifeycle, which follows one of two models: classic logistic dynamics or the Lotka-Volterra competition model. This is determined by the parameter log_dd, and it changes elements of the return list: K is returned for logistic dynamics, or gamma is returned for Lotka-Volterra dynamics. This is parameterized with the NF parameter to define the adult female numbers. This parameter only needs to be supplied if there are mosquito-only nodes.

Human-only nodes don't require any equilibrium calculations. These nodes use the NH and pop_ratio_H to set adult human populations and infection rates in nodes. These two parameters only need to be supplied if there are human-only nodes.

For human and mosquito nodes, this function calls make_Q_SEI to construct the infinitesimal generator matrix which is used to solve for the quasi-stationary (stochastic) or equilibrium (deterministic) distribution of mosquitoes over stages. Parameters are provided by params.

For information on the method used to solve this distribution, see section "3.1.3 Nonsingularity of the Subintensity Matrix" of:

- Bladt, Mogens, and Bo Friis Nielsen. Matrix-exponential distributions in applied probability. Vol. 81. New York: Springer, 2017.

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The initial population genotype ratios are set by supplying the pop_ratio_Aq, pop_ratio_F, and pop_ratio_M values. The default value is NULL, and the function will use the wild-type alleles provided in the cube object. However, one can supply several different objects to set the initial genotype ratios. All genotypes provided must exist in the cube (this is checked by the function). If a single, named vector is provided, then all patches will be initialized with the same ratios. If a matrix is provided, with the number of columns (and column names) giving the initial genotypes, and a row for each patch, each patch can be set to a different initial ratio. The three parameters do not need to match each other.

The params argument supplies all of the ecological and epidemiological parameters necessary to calculate equilibrium values. This is used to set the initial population distribution and during the simulation to maintain equilibrium. This params must include the following named parameters, noted as being the same as lifecycle parameters, or new for the epidemiological equilibrium

- **(Lifecycle parameters)**
  - qE: inverse of mean duration of egg stage
  - nE: shape parameter of Erlang-distributed egg stage
  - qL: inverse of mean duration of larval stage
  - nL: shape parameter of Erlang-distributed larval stage
  - qP: inverse of mean duration of pupal stage
  - nP: shape parameter of Erlang-distributed pupal stage

  - muE: egg mortality
  - muL: density-independent larvae mortality
  - muP: pupae mortality
  - muF: adult female mortality
  - muM: adult male mortality
  - beta: egg-laying rate, daily
  - nu: mating rate of unmated females
- **(Epidemiological parameters)**
  - NH: number of humans, can be a vector
  - X: prevalence in humans, can be a vector
  - NFX: number of female mosquitoes, only required if any prevalence (X) is zero
  - b: mosquito to human transmission efficiency, can be a vector
  - c: human to mosquito transmission efficiency, can be a vector
  - r: rate of recovery in humans (1/duration of infectiousness)
  - muH: death rate of humans (1/avg lifespan)
  - f: rate of blood feeding
  - Q: human blood index
  - qEIP: related to scale parameter of Gamma distributed EIP (1/qEIP is mean length of EIP)
  - nEIP: shape parameter of Gamma distributed EIP

The return list contains all of the parameters necessary later in the simulations.

For equilibrium without epidemiological parameters, see equilibrium_lifeycle. For equilibrium with latent humans (SEIR dynamics), see equilibrium_SEI_SEIR.

For examples of using this function, see: `vignette("lifecycle-node", package = "MGDrivE2")`

Value

a vector of the equilibrium number of females in each SEI stage

---

`get_shape` *Calculate Erlang shape parameter*

Description

Calculate Erlang shape parameter

Usage

```
get_shape(cv, q)
```

Arguments

cv	coefficient of variation (CV) between mean and standard deviation of dwell times, smaller values of CV correspond to distributions less dispersed around their mean and larger value to more dispersed distributions.
q	inverse of mean dwell time

Value

integer value representing the coefficient of variation in Erlang-distributed life stages.

---

human_Imperial_ODE *ODE describing the age-structured Imperial model used in decoupled sampling, which will pass in values of I_V and return the human states for usein the mosquito portion of the model*

Description

ODE describing the age-structured Imperial model used in decoupled sampling, which will pass in values of I_V and return the human states for usein the mosquito portion of the model

Usage

```
human_Imperial_ODE(t, state, parameters)
```

Arguments

t	starting time of simulation
state	distributon of disease states
parameters	parameter set

Value

matrix of disease states after integration

`imperial_model_param_list_create` *Model Parameter List Creation*

Description

`model_param_list_create` creates list of model parameters to be used within `equilibrium_init_create`

Usage

```
imperial_model_param_list_create(
  eta = 1/(21 * 365),
  rho = 0.85,
  a0 = 2920,
  sigma2 = 1.67,
  max_age = 100 * 365,
  rA = 1/195,
  rT = 0.2,
  rD = 0.2,
  rU = 1/110.299,
  rP = 1/15,
  dE = 12,
  delayGam = 12.5,
  cD = 0.0676909,
  cT = 0.322 * cD,
  cU = 0.006203,
  gamma1 = 1.82425,
  d1 = 0.160527,
  dID = 3650,
  ID0 = 1.577533,
  kD = 0.476614,
  uD = 9.44512,
  aD = 8001.99,
  fD0 = 0.007055,
  gammaD = 4.8183,
  alphaA = 0.75735,
  alphaU = 0.185624,
  b0 = 0.590076,
  b1 = 0.5,
  dB = 3650,
  IB0 = 43.8787,
  kB = 2.15506,
  uB = 7.19919,
  theta0 = 0.0749886,
  theta1 = 0.0001191,
  iv0 = 1.09629,
  kv = 2.00048,
```

```
  av = 2493.41,
  gammaV = 2.91282,
  fvS = 0.141195,
  pctMort = 0.215,
  phi0 = 0.791666,
  phi1 = 0.000737,
  dCA = 10950,
  IC0 = 18.02366,
  kC = 2.36949,
  uCA = 6.06349,
  PM = 0.774368,
  dCM = 67.6952,
  dVM = 76.8365,
  dVA = 30 * 365,
  PVM = 0.195768,
  uVA = 11.4321,
  tau1 = 0.69,
  tau2 = 2.31,
  muF = 0.132,
  nEIP = 3,
  qEIP = 1/10,
  Q0 = 0.92,
  DY = 365,
  thetaB = 0.89,
  thetaI = 0.97,
  r_llin = 0.56,
  s_llin = 0.03,
  r_irs = 0.6,
  s_irs = 0,
  qE = 1/3,
  nE = 2,
  qL = 1/7,
  nL = 3,
  qP = 1/1,
  nP = 2,
  muE = 0.05,
  muL = 0.15,
  muP = 0.05,
  muM = 0.132,
  eps = 58.9,
  nu = 1/(4/24),
  NH = 1000,
  ...
)
```

Arguments

eta	Death rate for expoential population distribtuion, i.e. 1/Mean Population Age. Default = 0.0001305

rho	Age-dependent biting parameter. Default = 0.85
a0	Age-dependent biting parameter. Default = 2920
sigma2	Variance of the log heterogeneity in biting rates. Default = 1.67
max_age	Maximum age in days. Default = 100*365
rA	Rate of leaving asymptomatic infection. Default = 0.00512821
rT	Rate of leaving treatment. Default = 0.2
rD	Rate of leaving clinical disease. Default = 0.2
rU	Rate of recovering from subpatent infection. Default = 0.00906627
rP	Rate of leaving prophylaxis. Default = 0.06666667
dE	Latent period of human infection. Default = 12
delayGam	Lag from parasites to infectious gametocytes. Default = 12.5
cD	Untreated disease contribution to infectiousness. Default = 0.0676909
cT	Treated disease contribution to infectiousness. Default = 0.322 * cD
cU	Subpatent disease contribution to infectiousness. Default = 0.006203
gamma1	Parameter for infectiousness of state A. Default = 1.82425
d1	Minimum probability due to maximum immunity. Default = 0.160527
dID	Inverse of decay rate. Default = 3650
ID0	Scale parameter. Default = 1.577533
kD	Shape parameter. Default = 0.476614
uD	Duration in which immunity is not boosted. Default = 9.44512
aD	Scale parameter relating age to immunity. Default = 8001.99
fD0	Time-scale at which immunity changes with age. Default = 0.007055
gammaD	Shape parameter relating age to immunity. Default = 4.8183
alphaA	PCR detection probability parameters state A. Default = 0.757
alphaU	PCR detection probability parameters state U. Default = 0.186
b0	Maximum probability due to no immunity. Default = 0.590076
b1	Maximum relative reduction due to immunity. Default = 0.5
dB	Inverse of decay rate. Default = 3650
IB0	Scale parameter. Default = 43.8787
kB	Shape parameter. Default = 2.15506
uB	Duration in which immunity is not boosted. Default = 7.19919
theta0	Maximum probability of severe infection due to no immunity. Default = 0.0749886
theta1	Maximum reduction due to to immunity. Default = 0.0001191
iv0	Scale parameter. Default = 1.09629
kv	Shape parameter. Default = 2.00048
av	Age-dependent modifier. Default = 2493.41
gammaV	Age-dependent modifier. Default = 2.91282

fvS	Age-dependent modifier. Default = 0.141195
pctMort	Percentage of severe cases that die. Default = 0.215
phi0	Maximum probability due to no immunity. Default = 0.791666
phi1	Maximum relative reduction due to immunity. Default = 0.000737
dCA	Inverse of decay rate. Default = 10950
IC0	Scale parameter. Default = 18.02366
kC	Shape parameter. Default = 2.36949
uCA	Duration in which immunity is not boosted. Default = 6.06349
PM	New-born immunity relative to mother's. Default = 0.774368
dCM	Inverse of decay rate of maternal immunity. Default = 67.6952
dVM	Inverse of decay rate. Default = 76.8365
dVA	Inverse of decay rate. Default = 30 * 365
PVM	New-born immunity to severe disease relative to mothers. Default = 0.195768
uVA	Duration in which immunity to severe disease is not boosted. Default = 11.4321
tau1	Duration of host seeking, assumed to be constant between species. Default = 0.69
tau2	Duration of mosquito resting after feed. Default = 2.31
muF	Daily mortality of adult mosquitos. Default = 0.132
nEIP	Number of Erlang-distributed EIP compartments. Default = 6
qEIP	Inverse of the mean duration of the EIP. Default = 1/10 (days)
Q0	Anthrophagy probability. Default = 0.92
DY	number of days in a year. Default = 365
thetaB	proportion of bites on a person in bed. Default = 0.89
thetaI	proportion of bites on a person outdoors. Default = 0.97
r_llin	probability of repeating a feeding attempt due to LLINs. Default = 0.56
s_llin	probability of feeding and surviving in presence of LLINs. Default = 0.03
r_irs	probability of repeating a feeding attempt due to IRS. Default = 0.60
s_irs	probability of feeding and surviving in presence of IRS. Default = 0
qE	mosquito egg lifecycle parameter. Default = 1/3
nE	mosquito egg lifecycle parameter. Default = 2
qL	mosquito larval lifecycle parameter. Default = 1/7
nL	mosquito larval lifecycle parameter. Default = 3
qP	mosquito pupae lifecycle parameter. Default = 1/1
nP	mosquito pupae lifecycle parameter. Default = 2
muE	death rate of egg stage. Default = 0.05
muL	death rate of larval stage. Default = 0.15
muP	death rate of pupae stage. Default = 0.05

muM	death rate of male adult stage. Default = 0.132
eps	eggs laid per day. Default = 58.9
nu	mosquito lifecycle parameter. Default = 1/(4/24
NH	number of humans. Default = 1000
...	Any other parameters needed for non-standard model. If they share the same name as any of the defined parameters model_param_list_create will stop. You can either write any extra parameters you like individually, e.g. model_param_list_create(extra1 = 1, extra2 = 2) and these parameteres will appear appended to the returned list, or you can pass explicitly the ellipsis argument as a list created before, e.g. model_param_list_create(...=list(extra1 = 1, extra2 = 2))

Value

A named vector of all baseline parameters required by the Imperial malaria model.

This function creates all of the necessary parameters for the Imperial model. Parameters furnished by MGDrivE will be removed from this function. Adapted from: https://github.com/mrc-ide/deterministic-malaria-model/blob/master/R/model_parameters.R

A newer version of the model also includes parameters for severe disease. See: https://github.com/mrc-ide/malariasimulation for details.

Examples

```
imperial_model_param_list_create(NH=1500)
imperial_model_param_list_create(qE=1/4)
```

---

make_Q_Imperial *Rate Matrix (Q) for Adult Mosquito SEI Dynamics*

Description

Construct the infinitesimal generator matrix for (individual) adult female infection dynamics. Adult females follow SEI (Susceptible-Exposed-Infectious) style dynamics with a Gamma distributed EIP, with a mean duration 1/q and variance 1/nq^2 (following shape-scale parameterization, EIP ~ Gamma(n,1/nq)). This function only constructs the rate matrix for either a single mosquito or cohort that all emerged at the same time (the rate matrix for a population with emergence is infinite in dimension).

Usage

```
make_Q_Imperial(q, n, mu, FOIv)
```

Arguments

q	related to scale parameter of Gamma distributed EIP (1/q is mean length of EIP)
n	shape parameter of Gamma distributed EIP
mu	mosquito mortality rate
FOIv	equilibrium force of infection on mosquitos

Value

rate matrix for a single (emergence) cohort of SEI mosquito

---

make_Q_SEI *Rate Matrix (Q) for Adult Mosquito SEI Dynamics*

Description

Construct the infinitesimal generator matrix for (individual) adult female infection dynamics. Adult females follow SEI (Susceptible-Exposed-Infectious) style dynamics with a Gamma distributed EIP, with a mean duration 1/q and variance 1/nq^2 (following shape-scale parameterization, EIP ~ Gamma(n,1/nq)). This function only constructs the rate matrix for either a single mosquito or cohort that all emerged at the same time (the rate matrix for a population with emergence is infinite in dimension).

Usage

```
make_Q_SEI(q, n, mu, c, a, x)
```

Arguments

q	related to scale parameter of Gamma distributed EIP (1/q is mean length of EIP)
n	shape parameter of Gamma distributed EIP
mu	mosquito mortality rate
c	human to mosquito transmission efficiency
a	human biting rate
x	prevalence of disease in humans

Value

rate matrix for a single (emergence) cohort of SEI mosquito

movement_prob2rate *Convert Stochastic Matrix to Rate Matrix*

Description

Given a stochastic matrix, return the rate matrix (infinitesimal generator) that would generate it when exponentiated over the interval of unit time.

Usage

```
movement_prob2rate(tau)
```

Arguments

tau	a row normalized stochastic matrix

Details

Warning: if the matrix provided has diagonal-only rows (i.e., the location is independent), the rate matrix will return 0 in that row, as there is no movement rate that can generate that scenario.

Value

a list with two elements: gamma negative diagonal of the rate matrix, mat matrix of row normalized off-diagonal elements

Examples

```
# generate random matrix for example
#  This represents a 3-node landscape, with random movement between nodes
moveMat <- matrix(data = runif(n = 9), nrow = 3, ncol = 3)
moveMat <- moveMat/rowSums(moveMat)

moveRate <- movement_prob2rate(tau = moveMat)
```

mu_ts *Mosquito Death Rates, Comoros Islands*

Description

This is a matrix containing estimated mosquito death rates from the Comoros Islands, between Mozambique and Madagascar. It provides hourly death rates over the course of one year.

Usage

```
data(mu_ts)
```

Format

matrix with 3 named columns and 8760 rows:

Grande_Comore Hourly death rates for main island

Moheli Hourly death rates for second island

Anjouan Hourly death rates for smallest island

---

sim_trajectory_base_CSV

Simulate Trajectory From one SPN Model

Description

This is an internal function to `sim_trajectory_CSV`. It does the actual sampling once all of the functions have been checked and setup.

Usage

```
sim_trajectory_base_CSV(
  x0,
  times,
  stepFun,
  folders,
  stage,
  events0 = NULL,
  batch = NULL,
  Sout = NULL,
  verbose = TRUE
)
```

Arguments

`x0`	the initial marking of the SPN (initial state)
`times`	sequence of sampling times
`stepFun`	a sampling function
`folders`	vector of folders to write output
`stage`	vector of life-stages to print
`events0`	a `data.frame` of events (uses the same format as required in package `deSolve` for consistency, see `events` for more information)
`batch`	a `list` of batch migration events, created from `batch_migration`, may be set to `NULL` if not used
`Sout`	an optional matrix to track event firings
`verbose`	print a progress bar?

Value

no return, prints .csv files into provided folders

`sim_trajectory_base_CSV_decoupled` *Simulate Trajectory From one SPN Model*

Description

This is an internal function to `sim_trajectory_CSV`. It does the actual sampling once all of the functions have been checked and setup.

Usage

```
sim_trajectory_base_CSV_decoupled(
  x0,
  h0,
  SPN_P,
  theta,
  times,
  stepFun,
  events0 = NULL,
  batch = NULL,
  Sout = NULL,
  verbose = TRUE,
  human_ode = "Imperial",
  cube = NULL,
  folders = folders
)
```

Arguments

`x0`	the initial marking of the SPN (initial state)
`h0`	initial human state distribution
`SPN_P`	stochastic petri net, places
`theta`	parameters
`times`	sequence of sampling times
`stepFun`	a sampling function
`events0`	a `data.frame` of events (uses the same format as required in package `deSolve` for consistency, see `events` for more information)
`batch`	a `list` of batch migration events, created from `batch_migration`, may be set to `NULL` if not used
`Sout`	an optional matrix to track event firings
`verbose`	print a progress bar?

human_ode	ode function used for human states
cube	inheritance cube
folders	vector of folders to write output

Value

no return, prints .csv files into provided folders

`sim_trajectory_base_R` *Simulate Trajectory From one SPN Model*

Description

This is an internal function to `sim_trajectory_R`. It does the actual sampling once all of the functions have been checked and setup.

Usage

```
sim_trajectory_base_R(
  x0,
  times,
  num_reps,
  stepFun,
  events = NULL,
  batch = NULL,
  Sout = NULL,
  verbose = TRUE
)
```

Arguments

x0	the initial marking of the SPN (initial state)
times	sequence of sampling times
num_reps	number of repetitions to run
stepFun	a sampling function
events	a `data.frame` of events (uses the same format as required in package `deSolve` for consistency, see `events` for more information)
batch	a `list` of batch migration events, created from `batch_migration`, may be set to `NULL` if not used
Sout	an optional matrix to track event firings
verbose	print a progress bar?

Value

matrix of sampled values

`sim_trajectory_base_R_decoupled_Imperial` *Simulate Trajectory From one SPN Model using Imperial Malaria model*

Description

This is an internal function to `sim_trajectory_R`. It does the actual sampling once all of the functions have been checked and setup.

Usage

```
sim_trajectory_base_R_decoupled_Imperial(
  x0,
  h0,
  SPN_P,
  theta,
  times,
  num_reps,
  stepFun,
  events = NULL,
  batch = NULL,
  Sout = NULL,
  verbose = TRUE,
  cube = NULL
)
```

Arguments

`x0`	the initial marking of the SPN (initial state)
`h0`	initial human state distribution
`SPN_P`	stochastic petri net, places
`theta`	parameters
`times`	sequence of sampling times
`num_reps`	number of repetitions to run
`stepFun`	a sampling function
`events`	a `data.frame` of events (uses the same format as required in package `deSolve` for consistency, see `events` for more information)
`batch`	a `list` of batch migration events, created from `batch_migration`, may be set to `NULL` if not used
`Sout`	an optional matrix to track event firings
`verbose`	print a progress bar?
`cube`	inheritance cube

Value

matrix of sampled values

`sim_trajectory_base_R_decoupled_SIS` *Simulate Trajectory From one SPN Model using Human SIS model*

Description

This is an internal function to `sim_trajectory_R`. It does the actual sampling once all of the functions have been checked and setup.

Usage

```
sim_trajectory_base_R_decoupled_SIS(
  x0,
  h0,
  SPN_P,
  theta,
  times,
  num_reps,
  stepFun,
  events = NULL,
  batch = NULL,
  Sout = NULL,
  verbose = TRUE,
  cube = NULL
)
```

Arguments

`x0`	the initial marking of the SPN (initial state)
`h0`	initial human state distribution
`SPN_P`	stochastic petri net, places
`theta`	parameters
`times`	sequence of sampling times
`num_reps`	number of repetitions to run
`stepFun`	a sampling function
`events`	a `data.frame` of events (uses the same format as required in package `deSolve` for consistency, see `events` for more information)
`batch`	a `list` of batch migration events, created from `batch_migration`, may be set to `NULL` if not used
`Sout`	an optional matrix to track event firings
`verbose`	print a progress bar?
`cube`	inheritance cube

Value

matrix of sampled values

`sim_trajectory_CSV` *Simulate Trajectory From a SPN Model*

Description

This function provides a unified interface to the various simulation algorithms for SPN, returning output sampled at a lattice of time points to the user, and handling various exogenous events that may occur during the simulation (such as release of adult mosquitoes).

Usage

```
sim_trajectory_CSV(
  x0,
  tmax,
  dt = 1,
  dt_stoch = 0.1,
  folders = "./",
  stage = c("M", "F"),
  S,
  hazards,
  Sout = NULL,
  sampler = "tau",
  method = "lsoda",
  events = NULL,
  batch = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

`x0`	the initial marking of the SPN (initial state, M0)
`tmax`	the final time to end simulation
`dt`	the time-step at which to return output (**not** the time-step of the sampling algorithm)
`dt_stoch`	time-step used for approximation of hazards
`folders`	vector of folders to write output
`stage`	life-stages to print. Any combination of: "E", "L", "P"," M", "U", "F", "H"
`S`	a stoichiometry `Matrix-class` object
`hazards`	list of hazard functions
`Sout`	an optional matrix to track event firings

sampler	determines sampling algorithm, one of; "ode", "tau", "cle", or "dm"
method	if sampler is "ode", the solver to use, from deSolve
events	a data.frame of events
batch	a list of batch migration events, created from batch_migration, may be set to NULL if not used
verbose	print a progress bar?
...	further named arguments passed to the step function

Details

dt_stoch is used by the Poisson Time-Step (step_PTS) and Chemical Langevin (step_CLE) methods to approximate the hazards. A smaller dt_stoch provides a better approximation, but will take longer to run.

The stoichiometry matrix (S) is generated in spn_S.

The list of hazards (hazards) come from spn_hazards.

Several samplers are provided. The default is a Poisson Time-Step (step_PTS) method. Other options are Gillespie's Direct Method (step_DM) and a Chemical Langevin sampler (step_CLE). Additionally, for convenience, an ODE "sampler" (step_ODE) is provided for compatibility with other samplers. This function uses methods from deSolve.

If using the ode sampler, several methods are provided in the deSolve package, see ode. For inhomogeneous systems, consider using the "rk4" method to avoid excessive integration times.

Additionally, events objects must follow the format required by deSolve. This was done for consistency, see events for more information.

This function writes all output to .csv files. Each simulation is written to a folder element - the number of repetitions is the number of folders provided. What life-stages get recorded is specified by the stage parameter. All life-stages can be stored, or any subset thereof. Females are split by infection status, i.e. by "S", "E", or "I".

This function tracks state variables specified by argument stage by default; an optional argument Sout can be provided to track number of event firings each time step (for discrete stochastic simulations), or cumulative intensity (for continuous stochastic simulations), or the rate function of particular events for ODE simulation. The matrix must have number of columns equal to number of events in the system (the number of hazard functions), and a row for each tracking variable. If Sout is provided, it outputs an additional csv, "Tracking.csv". The function track_hinf is provided, which builds a matrix to track human infection events.

To return simulations to R for further processing, see sim_trajectory_R.

Value

NULL - prints output to .csv files

`sim_trajectory_CSV_decoupled` *Simulate Trajectory From a SPN Model*

Description

This function provides a unified interface to the various simulation algorithms for SPN, returning output sampled at a lattice of time points to the user, and handling various exogenous events that may occur during the simulation (such as release of adult mosquitoes). This function is used in decoupled sampling, where the mosquito and human states are separated. This is used primarily when using the Imperial model of malaria transmission.

Usage

```
sim_trajectory_CSV_decoupled(
  x0,
  h0,
  inf_labels,
  tmax,
  dt = 1,
  dt_stoch = 0.1,
  folders = "./",
  S,
  hazards,
  SPN_P,
  theta,
  Sout = NULL,
  sampler = "tau",
  method = "lsoda",
  events = NULL,
  batch = NULL,
  verbose = TRUE,
  human_ode = "Imperial",
  cube = cube,
  ...
)
```

Arguments

`x0`	the initial marking of the SPN (initial state, M0)
`h0`	the initial human state distribution
`inf_labels`	labels corresponding to female mosquito infection hazard
`tmax`	the final time to end simulation
`dt`	the time-step at which to return output (**not** the time-step of the sampling algorithm)

dt_stoch	time-step used for approximation of hazards
folders	vector of folders to write output
S	a stoichiometry Matrix-class object
hazards	list of hazard functions
SPN_P	stochastic petri net places
theta	parameters
Sout	an optional matrix to track event firings
sampler	determines sampling algorithm, one of; "ode", "tau", "cle", or "dm"
method	if sampler is "ode", the solver to use, from deSolve
events	a data.frame of events
batch	a list of batch migration events, created from batch_migration, may be set to NULL if not used
verbose	print a progress bar?
human_ode	human ode function
cube	inheritance cube
...	further named arguments passed to the step function

Details

dt_stoch is used by the Poisson Time-Step (step_PTS) and Chemical Langevin (step_CLE) methods to approximate the hazards. A smaller dt_stoch provides a better approximation, but will take longer to run.

The stoichiometry matrix (S) is generated in spn_S.

The list of hazards (hazards) come from spn_hazards.

Several samplers are provided. The default is a Poisson Time-Step (step_PTS) method. Other options are Gillespie's Direct Method (step_DM) and a Chemical Langevin sampler (step_CLE). Additionally, for convenience, an ODE "sampler" (step_ODE) is provided for compatibility with other samplers. This function uses methods from deSolve.

If using the ode sampler, several methods are provided in the deSolve package, see ode. For inhomogeneous systems, consider using the "rk4" method to avoid excessive integration times.

Additionally, events objects must follow the format required by deSolve. This was done for consistency, see events for more information.

This function writes all output to .csv files. Each simulation is written to a folder element - the number of repetitions is the number of folders provided. For now, only adult mosquito states, human states, clinical incidence, and pathogen prevalence are written to CSVs.

This function tracks state variables specified by argument stage by default; an optional argument Sout can be provided to track number of event firings each time step (for discrete stochastic simulations), or cumulative intensity (for continuous stochastic simulations), or the rate function of particular events for ODE simulation. The matrix must have number of columns equal to number of events in the system (the number of hazard functions), and a row for each tracking variable. If Sout is provided, it outputs an additional csv, "Tracking.csv". The function track_hinf is provided, which builds a matrix to track human infection events.

To return simulations to R for further processing, see sim_trajectory_R.

Value

NULL - prints output to .csv files

`sim_trajectory_R` *Simulate Trajectory From a SPN Model*

Description

This function provides a unified interface to the various simulation algorithms for SPN, returning output sampled at a lattice of time points to the user, and handling various exogenous events that may occur during the simulation (such as release of adult mosquitoes).

Usage

```
sim_trajectory_R(
  x0,
  tmax,
  dt = 1,
  dt_stoch = 0.1,
  num_reps = 1,
  S,
  hazards,
  Sout = NULL,
  sampler = "tau",
  method = "lsoda",
  events = NULL,
  batch = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

`x0`	the initial marking of the SPN (initial state, M0)
`tmax`	the final time to end simulation (all simulations start at 0)
`dt`	the time-step at which to return output (**not** the time-step of the sampling algorithm)
`dt_stoch`	time-step used for approximation of hazards
`num_reps`	number of repetitions to run, default is 1.
`S`	a stoichiometry `Matrix-class` object
`hazards`	list of hazard functions
`Sout`	an optional matrix to track event firings
`sampler`	determines sampling algorithm, one of; "ode", "tau", "cle", or "dm"
`method`	if `sampler` is "ode", the solver to use, from `deSolve`

events	a data.frame of events, may be set to NULL if not used
batch	a list of batch migration events, created from batch_migration, may be set to NULL if not used
verbose	print a progress bar?
...	further named arguments passed to the step function

Details

dt_stoch is used by the Poisson Time-Step (step_PTS) and Chemical Langevin (step_CLE) methods to approximate the hazards. A smaller dt_stoch provides a better approximation, but will take longer to run.

The stoichiometry matrix (S) is generated in spn_S.

The list of hazards (hazards) come from spn_hazards.

Several samplers are provided. The default is a Poisson Time-Step (step_PTS) method. Other options are Gillespie's Direct Method (step_DM) and a Chemical Langevin sampler (step_CLE). Additionally, for convenience, an ODE "sampler" (step_ODE) is provided for compatibility with other samplers. This function uses methods from deSolve.

If using the ode sampler, several methods are provided in the deSolve package, see ode. For inhomogeneous systems, consider using the "rk4" method to avoid excessive integration times.

Additionally, events objects must follow the format required by deSolve. This was done for consistency, see events for more information.

This function tracks state variables by default; an optional argument Sout can be provided to track number of event firings each time step (for discrete stochastic simulations), or cumulative intensity (for continuous stochastic simulations), or the rate function of particular events for ODE simulation. The matrix must have number of columns equal to number of events in the system (the number of hazard functions), and a row for each tracking variable. The function track_hinf is provided, which builds a matrix to track human infection events.

To save output as .csv files, see sim_trajectory_CSV.

Value

a list with 2 elements: "state" is the array of returned state values, and "events" will return events tracked with Sout if provided, otherwise is NULL

---

sim_trajectory_R_decoupled

Simulate Trajectory From a SPN Model

Description

This function provides a unified interface to the various simulation algorithms for SPN, returning output sampled at a lattice of time points to the user, and handling various exogenous events that may occur during the simulation (such as release of adult mosquitoes).

Usage

```
sim_trajectory_R_decoupled(
  x0,
  h0,
  tmax,
  inf_labels,
  dt = 1,
  dt_stoch = 0.1,
  num_reps = 1,
  S,
  hazards,
  SPN_P,
  theta,
  Sout = NULL,
  sampler = "tau",
  method = "lsoda",
  events = NULL,
  batch = NULL,
  verbose = TRUE,
  human_ode = "Imperial",
  cube = cube,
  ...
)
```

Arguments

x0	the initial marking of the SPN (initial state, M0)
h0	the initial human state distribution
tmax	the final time to end simulation (all simulations start at 0)
inf_labels	labels corresponding to female mosquito infection hazard
dt	the time-step at which to return output (**not** the time-step of the sampling algorithm)
dt_stoch	time-step used for approximation of hazards
num_reps	number of repetitions to run, default is 1.
S	a stoichiometry Matrix-class object
hazards	list of hazard functions
SPN_P	stochastic petri net places
theta	parameters
Sout	an optional matrix to track event firings
sampler	determines sampling algorithm, one of; "ode", "tau", "cle", or "dm"
method	if sampler is "ode", the solver to use, from deSolve
events	a data.frame of events, may be set to NULL if not used
batch	a list of batch migration events, created from batch_migration, may be set to NULL if not used

verbose	print a progress bar?
human_ode	human ode function
cube	inheritance cube
...	further named arguments passed to the step function

Details

dt_stoch is used by the Poisson Time-Step (step_PTS) and Chemical Langevin (step_CLE) methods to approximate the hazards. A smaller dt_stoch provides a better approximation, but will take longer to run.

The stoichiometry matrix (S) is generated in spn_S.

The list of hazards (hazards) come from spn_hazards.

Several samplers are provided. The default is a Poisson Time-Step (step_PTS) method. Other options are Gillespie's Direct Method (step_DM) and a Chemical Langevin sampler (step_CLE). Additionally, for convenience, an ODE "sampler" (step_ODE) is provided for compatibility with other samplers. This function uses methods from deSolve.

If using the ode sampler, several methods are provided in the deSolve package, see ode. For inhomogeneous systems, consider using the "rk4" method to avoid excessive integration times.

Additionally, events objects must follow the format required by deSolve. This was done for consistency, see events for more information.

This function tracks state variables by default; an optional argument Sout can be provided to track number of event firings each time step (for discrete stochastic simulations), or cumulative intensity (for continuous stochastic simulations), or the rate function of particular events for ODE simulation. The matrix must have number of columns equal to number of events in the system (the number of hazard functions), and a row for each tracking variable. The function track_hinf is provided, which builds a matrix to track human infection events.

To save output as .csv files, see sim_trajectory_CSV.

Value

a list with 2 elements: "state" is the array of returned state values, and "events" will return events tracked with Sout if provided, otherwise is NULL

---

solve_muAqua *Solve for Constant Aquatic Mortality*

Description

In MGDrivE, the model was typically solved at equilibrium assuming the density-independent mortality was constant over aquatic stages (eggs, larvae, pupae), given a daily growth rate, r_M. Given that growth rate, it solved for that mortality μ_{Aqua} by relating it with R_M, the per-generation growth rate of the population, calculable from r_M and the mean duration of life stages. This function uses uniroot to solve for mu_{Aqua}.

Usage

```
solve_muAqua(params, rm)
```

Arguments

params	a named list of parameters
rm	the daily growth rate

Details

This function needs the following parameters in params:

- muF: adult female mortality
- beta: rate of egg laying
- phi: sex ratio at emergence
- qE: inverse of mean duration of egg stage
- nE: shape parameter of Erlang-distributed egg stage
- qL: inverse of mean duration of larval stage
- nL: shape parameter of Erlang-distributed larval stage
- qP: inverse of mean duration of pupal stage
- nP: shape parameter of Erlang-distributed pupal stage

Value

location of the root, as provided from uniroot

Examples

```
theta <- list(qE = 1/4, nE = 2, qL = 1/5, nL = 3, qP = 1/6, nP = 2, muF = 1/12,
              beta = 32, phi = 0.5);
muAqatic <- solve_muAqua(params = theta, rm = 1.096)
```

---

split_aggregate_CSV *Split CSV output by Patch and Aggregate by Mate or Dwell-Stage*

Description

This function reads in the output files from sim_trajectory_CSV and splits them into smaller files. The files are output by patch, with the appropriate patch numbers for mosquitoes or humans, and specific stages are aggregated by a given metric.

Usage

```
split_aggregate_CSV(
  read_dir,
  write_dir = read_dir,
  stage = c("E", "L", "P", "M", "U", "FS", "FE", "FI", "H"),
  spn_P,
  tmax,
  dt,
  erlang = FALSE,
  sum_fem = FALSE,
  rem_file = FALSE,
  verbose = TRUE
)
```

Arguments

read_dir	Directory where output was written to
write_dir	Directory to write output to. Default is read_dir
stage	Life stage to print, see details
spn_P	Places object, see details
tmax	The final time to end simulation
dt	The time-step at which to return output (**not** the time-step of the sampling algorithm)
erlang	Boolean, default is FALSE, to return summaries by genotype
sum_fem	if TRUE, in addition to FS, FE, FI output by node and repetition, output an additional file F which sums over infection states (S,E,I). Does nothing if the simulation did not include epi dynamics.
rem_file	Remove original output? Default is FALSE
verbose	Chatty? Default is TRUE

Details

Given the read_dir, this function assumes the follow file structure:

- read_dir
  - repetition 1
    - M.csv
    - FS.csv
    - ...
  - repetition 2
    - M.csv
    - FS.csv

    * ...
  - repetition 3
  - ...

This function expects the `write_dir` to be empty, and it sets up the same file structure as the `read_dir`. For a 2-node simulation, the output will be organized similar to:

- write_dir
  - repetition 1
    * M_0001.csv
    * M_0002.csv
    * FS_0001.csv
    * FS_0001.csv
    * ...
  - repetition 2
    * M_0001.csv
    * M_0002.csv
    * FS_0001.csv
    * FS_0001.csv
    * ...
  - repetition 3
  - ...

`stage` defines which life-stages the function will analyze. These stages must be any combination of: "E", "L", "P", "M", "U", "FS", "FE", "FI", "H". These must come from the set of stages provided to `sim_trajectory_CSV` via the `stage` argument. It can be less than what was printed by the simulation, but any extra stages provided, but not printed, will throw a warning and then be ignored.

`erlang` defines how aquatic (eggs, larvae, and pupae) stages and adult females (only mated females) are aggregated. By default, `erlang` is FALSE, and all of these stages are summarized by genotype only, combining any Erlang-distributed dwell stages (for eggs, larvae, and pupae) or latent infection (for adult females) stages. If `erlang` is TRUE, summaries are returned by dwell stage or infection status, combining any genotype information.
Female summaries always combine over mate-genotype, so only female genotypes are returned.

The places (`spn_P`) object is generated from one of the following: `spn_P_lifecycle_node`, `spn_P_lifecycle_network`, `spn_P_epiSIS_node`, `spn_P_epiSIS_network`, `spn_P_epiSEIR_node`, or `spn_P_epiSEIR_network`.

`tmax`, `dt` define the last sampling time, and each sampling time in-between.

For more details about using this function to process CSV output see: `vignette("data-analysis", package = "MGDrivE2")`

Value

Writes output to files in write_dir

`split_aggregate_CSV_decoupled` *Split CSV output for decoupled sampling with Imperial malaria model*

Description

This function reads in the output files from `sim_trajectory_CSV` and splits them into smaller files. The files are output by patch, with the appropriate patch numbers for mosquitoes or humans, and specific stages are aggregated by a given metric.

Usage

```
split_aggregate_CSV_decoupled(
  read_dir,
  write_dir = read_dir,
  spn_P,
  tmax,
  dt,
  human_states,
  sum_fem = FALSE,
  rem_file = FALSE,
  verbose = TRUE,
  erlang = FALSE
)
```

Arguments

`read_dir`	Directory where output was written to
`write_dir`	Directory to write output to. Default is read_dir
`spn_P`	Places object, see details
`tmax`	The final time to end simulation
`dt`	The time-step at which to return output (**not** the time-step of the sampling algorithm)
`human_states`	human state distribution
`sum_fem`	if `TRUE`, in addition to FS, FE, FI output by node and repetition, output an additional file F which sums over infection states (S,E,I). Does nothing if the simulation did not include epi dynamics.
`rem_file`	Remove original output? Default is FALSE
`verbose`	Chatty? Default is TRUE
`erlang`	erlang distributed states

Details

Given the `read_dir`, this function assumes the follow file structure:

- read_dir
  - repetition 1
    - M.csv
    - FS.csv
    - ...
  - repetition 2
    - M.csv
    - FS.csv
    - ...
  - repetition 3
  - ...

This function expects the `write_dir` to be empty, and it sets up the same file structure as the `read_dir`. For a 2-node simulation, the output will be organized similar to:

- write_dir
  - repetition 1
    - M_0001.csv
    - M_0002.csv
    - FS_0001.csv
    - FS_0001.csv
    - ...
  - repetition 2
    - M_0001.csv
    - M_0002.csv
    - FS_0001.csv
    - FS_0001.csv
    - ...
  - repetition 3
  - ...

The places (`spn_P`) object is generated from one of the following: `spn_P_lifecycle_node`, `spn_P_lifecycle_network`, `spn_P_epiSIS_node`, `spn_P_epiSIS_network`, `spn_P_epiSEIR_node`, or `spn_P_epiSEIR_network`.

`tmax`, `dt` define the last sampling time, and each sampling time in-between.

For more details about using this function to process CSV output see: `vignette("data-analysis", package = "MGDrivE2")`

Value

Writes output to files in write_dir

---

spn_hazards *Make Hazards (Lambda) For a MGDrivE2: Node and Network Simulations*

Description

Using the structural (topological) SPN model as well as parameters in the cube and params objects, generate a list (of length |v|) of hazards, each implemented as a function closure.

Usage

```
spn_hazards(
  spn_P,
  spn_T,
  cube,
  params,
  type = "life",
  log_dd = TRUE,
  exact = TRUE,
  tol = 1e-12,
  verbose = TRUE,
  trap_idx = NULL
)
```

Arguments

spn_P	the set of places (P) (see details)
spn_T	the set of transitions (T) (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)
params	a named list of parameters (see details)
type	string indicating type of hazards, one of; "life", "SIS", or "SEIR"
log_dd	if TRUE, use logistic (carrying capacity) density dependent hazards, if FALSE use Lotka-Volterra density dependent hazards for larval mortality
exact	boolean, make exact (integer input) hazards? Default is TRUE
tol	if exact=FALSE, the value of hazard below which it is clipped to 0
verbose	display a progress bar when making hazards?
trap_idx	Vector of indices corresponding to trap nodes.

Details

If these hazards will be used in a continuous approximation algorithm, such as an ODE method (step_ODE) or Gillespie's Direct Method (step_DM), it is recommended to use exact=FALSE. If the hazards will be used in an integer state space method, such as tau-leaping (step_PTS) or Chemical Langevin (step_CLE) methods, it is recommended to use exact=TRUE.

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The set of transitions (spn_T) is generated from one of the following: spn_T_lifecycle_node, spn_T_lifecycle_network, spn_T_epiSIS_node, spn_T_epiSIS_network, spn_T_epiSEIR_node, spn_T_epiSEIR_network.

The params objected is generated from either equilibrium_lifeycle or equilibrium_SEI_SIS; it is the "params" object in the return list. The equilibrium function used must match the type parameter.

The type parameter indicates what type of simulation is being run. It is one of: "life", "SIS", or "SEIR". This must match the params object supplied.

Use of this function is demonstrated in many vignettes, browseVignettes(package = "MGDrivE2")

Value

list of length 2: hazards is a list of named closures for every state transition in the model, flag is a boolean indicating exact or approximate

---

spn_hazards_decoupled *Make Hazards (Lambda) For a MGDrivE2: Node and Network Simulations*

Description

Using the structural (topological) SPN model as well as parameters in the cube and params objects, generate a list (of length |v|) of hazards, each implemented as a function closure.

Usage

```
spn_hazards_decoupled(
  spn_P,
  spn_T,
  cube,
  params,
  type = "SIS",
  log_dd = TRUE,
  exact = TRUE,
  tol = 1e-12,
  verbose = TRUE
)
```

Arguments

spn_P	the set of places (P) (see details)
spn_T	the set of transitions (T) (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)
params	a named list of parameters (see details)
type	string indicating type of hazards, one of; "life", "SIS", "Imperial" or "SEIR"
log_dd	if TRUE, use logistic (carrying capacity) density dependent hazards, if FALSE use Lotka-Volterra density dependent hazards for larval mortality
exact	boolean, make exact (integer input) hazards? Default is TRUE
tol	if exact=FALSE, the value of hazard below which it is clipped to 0
verbose	display a progress bar when making hazards?

Details

If these hazards will be used in a continuous approximation algorithm, such as an ODE method (step_ODE) or Gillespie's Direct Method (step_DM), it is recommended to use exact=FALSE. If the hazards will be used in an integer state space method, such as tau-leaping (step_PTS) or Chemical Langevin (step_CLE) methods, it is recommended to use exact=TRUE.

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The set of transitions (spn_T) is generated from one of the following: spn_T_lifecycle_node, spn_T_lifecycle_network, spn_T_epiSIS_node, spn_T_epiSIS_network, spn_T_epiSEIR_node, spn_T_epiSEIR_network.

The params objected is generated from either equilibrium_lifeycle or equilibrium_SEI_SIS; it is the "params" object in the return list. The equilibrium function used must match the type parameter.

The type parameter indicates what type of simulation is being run. It is one of: "life", "SIS", or "SEIR". This must match the params object supplied.

Use of this function is demonstrated in many vignettes, browseVignettes(package = "MGDrivE2")

Value

list of length 2: hazards is a list of named closures for every state transition in the model, flag is a boolean indicating exact or approximate

---

spn_Post *Make Post Matrix For a Petri Net*

Description

Generate the Post (|v| by |u|) matrix for the SPN. This gives the edges from T to P (output arcs) in the bipartite network.

Usage

```
spn_Post(spn_P, spn_T)
```

Arguments

spn_P	set of places (P) (see details)
spn_T	set of transitions (T) (see details)

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The set of transitions (spn_T) is generated from one of the following: spn_T_lifecycle_node, spn_T_lifecycle_network, spn_T_epiSIS_node, spn_T_epiSIS_network, spn_T_epiSEIR_node, spn_T_epiSEIR_network.

Value

a matrix of type dgCMatrix-class

---

spn_Pre *Make Pre Matrix For a Petri Net*

Description

Generate the Pre (|v| by |u|) matrix for the SPN. This gives the edges from P to T (input arcs) in the bipartite network.

Usage

```
spn_Pre(spn_P, spn_T)
```

Arguments

spn_P	set of places (P) (see details)
spn_T	set of transitions (T) (see details)

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The set of transitions (spn_T) is generated from one of the following: spn_T_lifecycle_node, spn_T_lifecycle_network, spn_T_epiSIS_node, spn_T_epiSIS_network, spn_T_epiSEIR_node, spn_T_epiSEIR_network.

Value

a matrix of type dgCMatrix-class

spn_P_epiSEIR_network *Make Places (P) For a Network (SEI Mosquitoes - SEIR Humans)*

Description

This function makes the set of places (P) for a SPN model of a metapopulation network for simulation of coupled SEI-SEIR dynamics. It is the network version of spn_P_epiSEIR_node.

Usage

```
spn_P_epiSEIR_network(node_list, params, cube)
```

Arguments

node_list	a character vector specifying what type of nodes to create; (m = a node with only mosquitoes, h = a node with only humans, b = a node with both humans and mosquitoes)
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SEIR

For examples of using this function, see: vignette("seir-dynamics", package = "MGDrivE2")

Value

a list with two elements: ix contains labeled indices of the places by life stage and node, u is the character vector of places (P)

spn_P_epiSEIR_node *Make Places (P) For a Node (SEI Mosquitoes - SEIR Humans)*

Description

This function makes the set of places (P) for a SPN. It is used alone if our model is a single-node metapopulation for mosquito SEI and human SEIR dynamics; otherwise it is used as part of other functions to make SPN models with larger state spaces (metapopulation models, spn_P_epiSEIR_network).

Usage

```
spn_P_epiSEIR_node(params, cube)
```

Arguments

params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SEIR

For examples of using this function, see: vignette("seir-dynamics", package = "MGDrivE2")

Value

a list with two elements: ix contains labeled indices of the places by life stage, u is the character vector of places (P)

---

spn_P_epiSIS_network *Make Places (P) For a Network (SEI Mosquitoes - SIS Humans)*

Description

This function makes the set of places (P) for a SPN model of a metapopulation network for simulation of coupled SEI-SIS dynamics. It is the network version of spn_P_epiSIS_node.

Usage

```
spn_P_epiSIS_network(node_list, params, cube)
```

Arguments

node_list	a character vector specifying what type of nodes to create; (m = a node_id with only mosquitoes, h = a node_id with only humans, b = a node_id with both humans and mosquitoes)
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SIS

For examples of using this function, see: vignette("epi-network", package = "MGDrivE2")

Value

a list with two elements: ix contains labeled indices of the places by life stage and node_id, u is the character vector of places (P)

spn_P_epiSIS_node *Make Places (P) For a Node (SEI Mosquitoes - SIS Humans)*

Description

This function makes the set of places (P) for a SPN. It is used alone if our model is a single-node metapopulation for mosquito SEI and human SIS dynamics; otherwise it is used as part of other functions to make SPN models with larger state spaces (metapopulation models, see spn_P_epiSIS_network).

Usage

```
spn_P_epiSIS_node(params, cube)
```

Arguments

params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SIS

For examples of using this function, see: vignette("epi-node", package = "MGDrivE2")

Value

a list with two elements: ix contains labeled indices of the places by life stage, u is the character vector of places (P)

---

spn_P_epi_decoupled_node *Make Places (P) For a Node (SEI Mosquitoes). Note in the v2 epi module, we only use the SPN framework for the mosquito component of the model. The human compoenent will be handled separately in the sampler, and will be formulated as an ODE. This function makes the set of places (P) for a SPN. It is used alone if our model is a single-node metapopulation for mosquito SEI and dynamics; This is used by both SIS and Imperial transmission models.*

Description

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SIS

Usage

```
spn_P_epi_decoupled_node(params, cube)
```

Arguments

params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

For examples of using this function, see: `vignette("epi-node", package = "MGDrivE2")`

Value

a list with two elements: ix contains labeled indices of the places by life stage, u is the character vector of places (P)

---

spn_P_lifecycle_network

Make Places (P) For a Network (Mosquitoes only)

Description

This function makes the set of places (P) for a SPN model of a metapopulation network. It is the network version of spn_P_lifecycle_node.

Usage

```
spn_P_lifecycle_network(num_nodes, params, cube)
```

Arguments

num_nodes	number of nodes in the network
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, and nP parameters to be specified. For more details, see equilibrium_lifeycle

For examples of using this function, see: `vignette("lifecycle-network", package = "MGDrivE2")`

Value

a list with two elements: ix contains labeled indices of the places by life stage and node_id, u is the character vector of places (P)

spn_P_lifecycle_node *Make Places (P) For a Node (Mosquitoes only)*

Description

This function makes the set of places (P) for a SPN. It is used alone if our model is a single-node metapopulation for mosquito dynamics only; otherwise it is used as part of other functions to make SPN models with larger state spaces (metapopulation models, see spn_P_lifecycle_network).

Usage

```
spn_P_lifecycle_node(params, cube)
```

Arguments

params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, and nP parameters to be specified. For more details, see equilibrium_lifeycle

For examples of using this function, see: vignette("lifecycle-node", package = "MGDrivE2")

Value

a list with two elements: ix contains labeled indices of the places by life stage, u is the character vector of places (P)

spn_S *Make stoichiometry Matrix For a Petri Net*

Description

Generate the stoichiometry (|u| by |v|) matrix for the SPN. Each column gives the net effect of that transition firing upon the state space of the model. Internally, this creates a Pre (spn_Pre) and Post (spn_Post) matrix, and then calculates the final stoichiometry.

Usage

```
spn_S(spn_P, spn_T)
```

Arguments

spn_P	set of places (P) (see details)
spn_T	set of transitions (T) (see details)

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The set of transitions (spn_T) is generated from one of the following: spn_T_lifecycle_node, spn_T_lifecycle_network, spn_T_epiSIS_node, spn_T_epiSIS_network, spn_T_epiSEIR_node, spn_T_epiSEIR_network.

Value

stoichiometry matrix representing the net effect of a transition in the SPN state model.

---

spn_T_epiSEIR_network *Make Transitions (T) For a Network (SEI Mosquitoes - SEIR Humans)*

Description

This function makes the set of transitions (T) for a SPN model of a metapopulation network for simulation of coupled SEI-SEIR dynamics. It is the network version of spn_T_epiSEIR_node.

Usage

```
spn_T_epiSEIR_network(node_list, spn_P, params, cube, h_move, m_move)
```

Arguments

node_list	a character vector specifying what type of nodes to create; (m = a node with only mosquitoes, h = a node with only humans, b = a node with both humans and mosquitoes)
spn_P	set of places produced by spn_P_epiSEIR_network
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)
h_move	binary adjacency matrix indicating if movement of humans between nodes is possible or not
m_move	binary adjacency matrix indicating if movement of mosquitoes between nodes is possible or not

Details

This function takes the places produced from spn_P_epiSEIR_network and builds all possible transitions between subsets of those places.

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SEIR

While this function produces all structural information related to transitions, hazards are produced by a separate function, spn_hazards.

For larger networks, this function may take some time to return, please be patient; the Petri Net modeling formalism trades additional computation time at model initialization for faster sampling of trajectories within a simulation.

Please note, the movement matrices (h_move and m_move) are NOT stochastic matrices, just binary matrices that say if i,j can exchange population. Diagonal elements must be FALSE, and both matrices are checked for validity; the function will stop with errors if the adjacency matrix specifies illegal movement rules (e.g.; mosquito movement from a "h" node to a "b" node)

For examples of using this function, see: vignette("seir-dynamics", package = "MGDrivE2")

Value

a list with two elements: T contains transitions packets as lists, v is the character vector of transitions (T)

---

spn_T_epiSEIR_node *Make Transitions (T) For a Node (SEI Mosquitoes - SEIR Humans)*

Description

This function makes the set of transitions (T) for a SPN. It is used alone if our model is a single-node metapopulation of mosquito and human dynamics; otherwise it is used as part of other functions to make SPN models with larger state spaces (metapopulation models, see spn_T_epiSEIR_network).

Usage

```
spn_T_epiSEIR_node(spn_P, params, cube)
```

Arguments

spn_P	set of places produced by spn_P_epiSEIR_node
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

This function takes the places produced from spn_P_epiSEIR_node and builds all possible transitions between subsets of those places.

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SEIR

While this function produces all structural information related to transitions, hazards are produced by a separate function, spn_hazards.

For examples of using this function, see: vignette("seir-dynamics", package = "MGDrivE2")

Value

a list with two elements: T contains transitions packets as lists, v is the character vector of transitions (T)

---

spn_T_epiSIS_network *Make Transitions (T) For a Network (SEI Mosquitoes - SIS Humans)*

Description

This function makes the set of transitions (T) for a SPN model of a metapopulation network for simulation of coupled SEI-SIS dynamics. It is the network version of spn_T_epiSIS_node.

Usage

```
spn_T_epiSIS_network(node_list, spn_P, params, cube, h_move, m_move)
```

Arguments

node_list	a character vector specifying what type of nodes to create; (m = a node with only mosquitoes, h = a node with only humans, b = a node with both humans and mosquitoes)
spn_P	set of places produced by spn_P_epiSIS_network
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)
h_move	binary adjacency matrix indicating if movement of humans between nodes is possible or not
m_move	binary adjacency matrix indicating if movement of mosquitoes between nodes is possible or not

Details

This function takes the places produced from spn_P_epiSIS_network and builds all possible transitions between subsets of those places.

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SIS

While this function produces all structural information related to transitions, hazards are produced by a separate function, spn_hazards.

For larger networks, this function may take some time to return, please be patient; the Petri Net modeling formalism trades additional computation time at model initialization for faster sampling of trajectories within a simulation.

Please note, the movement matrices (h_move and m_move) are NOT stochastic matrices, just binary matrices that say if i,j can exchange population. Diagonal elements must be FALSE, and both matrices are checked for validity; the function will stop with errors if the adjacency matrix specifies illegal movement rules (e.g.; mosquito movement from a "h" node to a "b" node)

For examples of using this function, see: vignette("epi-network", package = "MGDrivE2")

Value

a list with two elements: T contains transitions packets as lists, v is the character vector of transitions (T)

---

spn_T_epiSIS_node *Make Transitions (T) For a Node (SEI Mosquitoes - SIS Humans)*

Description

This function makes the set of transitions (T) for a SPN. It is used alone if our model is a single-node metapopulation of mosquito and human dynamics; otherwise it is used as part of other functions to make SPN models with larger state spaces (metapopulation models, see spn_T_epiSIS_network).

Usage

```
spn_T_epiSIS_node(spn_P, params, cube)
```

Arguments

spn_P	set of places produced by spn_P_epiSIS_node
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

This function takes the places produced from spn_P_epiSIS_node and builds all possible transitions between subsets of those places.

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SIS

While this function produces all structural information related to transitions, hazards are produced by a separate function, spn_hazards.

For examples of using this function, see: vignette("epi-node", package = "MGDrivE2")

Value

a list with two elements: T contains transitions packets as lists, v is the character vector of transitions (T)

---

spn_T_epi_decoupled_node

Make Transitions (T) For a Node (SEI Mosquitoes)

Description

This function makes the set of transitions (T) for a SPN. It is used alone if our model is a single-node metapopulation of mosquito; otherwise it is used as part of other functions to make SPN models with larger state spaces (metapopulation models, see spn_T_epiSIS_network).

Usage

```
spn_T_epi_decoupled_node(spn_P, params, cube)
```

Arguments

spn_P	set of places produced by spn_P_epi_decoupled_node function
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. cubeMendelian)

Details

This function takes the places produced from spn_P_epi_decoupled_node and builds all possible transitions between subsets of those places.

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, nP, and nEIP parameters to be specified. For more details, see equilibrium_SEI_SIS

While this function produces all structural information related to transitions, hazards are produced by a separate function, spn_hazards. This is used by both decoupled SIS and Imperial transmission model sampling. For examples of using this function, see: vignette("epi-node-decoupled", package = "MGDrivE2")

Value

a list with two elements: T contains transitions packets as lists, v is the character vector of transitions (T)

`spn_T_lifecycle_network` *Make Transitions (T) For a Network (Mosquitoes only)*

Description

This function makes the set of transitions (T) for a SPN model of a metapopulation network. It is the network version of `spn_T_lifecycle_node`.

Usage

```
spn_T_lifecycle_network(spn_P, params, cube, n = NULL, m_move = NULL)
```

Arguments

spn_P	set of places produced by `spn_P_lifecycle_network`
params	a named list of parameters (see details)
cube	an inheritance cube from the MGDrivE package (e.g. `cubeMendelian`)
n	an integer giving the number of nodes
m_move	binary adjacency matrix indicating if movement of mosquitoes between nodes is possible or not

Details

This function takes the places produced from `spn_P_lifecycle_network` and builds all possible transitions between subsets of those places.

The params argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the nE, nL, and nP parameters to be specified. For more details, see `equilibrium_lifeycle`

While this function produces all structural information related to transitions, hazards are produced by a separate function, `spn_hazards`.

For larger networks, this function may take some time to return, please be patient; the Petri Net modeling formalism trades additional computation time at model initialization for faster sampling of trajectories within a simulation.

Please note, the movement matrix (m_move) is NOT a stochastic matrices, just a binary matrix that say if i,j can exchange population. Diagonal elements must be FALSE.

At least one of the arguments n and m_move must be provided. If both are provided n is ignored.

For examples of using this function, see: `vignette("lifecycle-network", package = "MGDrivE2")`

Value

a list with two elements: T contains transitions packets as lists, v is the character vector of transitions (T)

`spn_T_lifecycle_node` *Make Transitions (T) For a Node (Mosquitoes only)*

Description

This function makes the set of transitions (T) for a SPN. It is used alone if our model is a single-node metapopulation for mosquito dynamics only; otherwise it is used as part of other functions to make SPN models with larger state spaces (metapopulation models, see `spn_T_lifecycle_network`).

Usage

```
spn_T_lifecycle_node(spn_P, params, cube)
```

Arguments

`spn_P`	set of places produced by `spn_P_lifecycle_node`
`params`	a named list of parameters (see details)
`cube`	an inheritance cube from the `MGDrivE` package (e.g. `cubeMendelian`)

Details

This function takes the places produced from `spn_P_lifecycle_node` and builds all possible transitions between subsets of those places.

The `params` argument supplies all of the ecological parameters necessary to calculate equilibrium values. This function requires the `nE`, `nL`, and `nP` parameters to be specified. For more details, see `equilibrium_lifeycle`

While this function produces all structural information related to transitions, hazards are produced by a separate function, `spn_hazards`.

For examples of using this function, see: `vignette("lifecycle-node", package = "MGDrivE2")`

Value

a list with two elements: T contains transitions packets as lists, v is the character vector of transitions (T)

step_CLE *Make Chemical Langevin (CLE) Sampler for a SPN model*

Description

Make a function closure to implement a chemical Langevin (continuous-state) approximation for a SPN.

Usage

```
step_CLE(S, Sout, haz, dt = 0.01, maxhaz = 1e+06)
```

Arguments

S	a stoichiometry Matrix-class object
Sout	an optional matrix to track of event firings. In the continuous stochastic model this will be the approximate cumulative intensity of each event.
haz	a list of hazard functions
dt	time-step for Euler-Maruyama method used to solve the SDE system
maxhaz	maximum allowable hazard

Details

The chemical Langevin approximation is a numerical simulation of a Fokker-Planck approximation to the Master equations (Kolmogorov Forwards Equations) governing the stochastic model; the CLE approximation is a second-order approximation that will get the correct mean and variance but higher order moments will be incorrect.

The design of step_CLE is from: Wilkinson, D. J. (2011). Stochastic modeling for systems biology. CRC press

Elements of the N list come from two places: The stoichiometry matrix (S) is generated in spn_S and the hazards (h) come from spn_hazards.

For other samplers, see: step_PTS, step_DM, step_ODE

Value

function closure for use in sim_trajectory_R or sim_trajectory_CSV

step_DM *Make Gillespie's Direct Method (DM) Sampler for a SPN model*

Description

Make a function closure to implement Gillespie's Direct Method sampler for a SPN.

Usage

```
step_DM(S, Sout, haz, maxhaz = 1e+06)
```

Arguments

S	a stoichiometry Matrix-class object
Sout	an optional matrix to track of event firings
haz	a list of hazard functions
maxhaz	maximum allowable hazard

Details

The direct method is an exact sampling algorithm; it simulates each event individually. Because of this it may be extremely slow for non-trivial population sizes, and thus should be used to debug and test rather than for serious Monte Carlo simulation.

The design of step_DM is from: Wilkinson, D. J. (2011). Stochastic modeling for systems biology. CRC press

Elements of the N list come from two places: The stoichiometry matrix (S) is generated in spn_S and the hazards (h) come from spn_hazards.

For other samplers, see: step_CLE, step_PTS, step_ODE

Value

function closure for use in sim_trajectory_R or sim_trajectory_CSV

step_ODE *Make Mean-field Approximation (ODE) Numerical Integrator for a SPN Model*

Description

Make a function closure to implement a first order mean-field ODE approximation for a SPN.

Usage

```
step_ODE(S, Sout, haz, method = "lsoda")
```

Arguments

S	a stoichiometry Matrix-class object
Sout	an optional matrix to track of event firings. In the deterministic case it will return the rate of that event at the end of the time step
haz	a list of hazard functions
method	a character giving the type of numerical integrator used, the default is "lsoda"

Details

This method is equivalent to considering the ODEs describing the time evolution of the mean trajectory (first moment) and setting all higher order moments which appear on the right hand side to zero.

The solvers used within can be found in the deSolve package, see ode. For inhomogeneous systems, consider using the "rk4" method to avoid excessive integration times.

The stoichiometry matrix (S) is generated in spn_S.

The list of hazards (haz) come from spn_hazards.

For other samplers, see: step_CLE, step_PTS, step_DM

Value

function closure for use in sim_trajectory_R or sim_trajectory_CSV

---

step_ODE_decoupled *Make Mean-field Approximation (ODE) Numerical Integrator for a SPN Model for Decoupled Epi Dynamics*

Description

Make a function closure to implement a first order mean-field ODE approximation for a SPN.

Usage

```
step_ODE_decoupled(S, Sout, haz, method = "lsoda", human_ode = "SIS")
```

Arguments

S	a stoichiometry Matrix-class object
Sout	an optional matrix to track of event firings. In the deterministic case it will return the rate of that event at the end of the time step
haz	a list of hazard functions
method	a character giving the type of numerical integrator used, the default is "lsoda"
human_ode	ODE function used for human states

Details

This method is equivalent to considering the ODEs describing the time evolution of the mean trajectory (first moment) and setting all higher order moments which appear on the right hand side to zero.

The solvers used within can be found in the deSolve package, see ode. For inhomogeneous systems, consider using the "rk4" method to avoid excessive integration times.

The stoichiometry matrix (S) is generated in spn_S.

The list of hazards (haz) come from spn_hazards.

For other samplers, see: step_CLE, step_PTS, step_DM

Value

function closure for use in sim_trajectory_R or sim_trajectory_CSV

---

step_PTS *Make Poisson Time-Step (PTS) Sampler for a SPN Model*

Description

Make a function closure to implement a Poisson time-step (tau-leaping with fixed tau) sampler for a SPN.

Usage

```
step_PTS(S, Sout, haz, dt = 0.01, maxhaz = 1e+06)
```

Arguments

S	a stoichiometry Matrix-class object
Sout	an optional matrix to track of event firings
haz	a list of hazard functions
dt	time-step for tau-leap method
maxhaz	maximum allowable hazard

Details

This sampling algorithm is based on representing a SPN as a set of competing Poisson processes; it thus uses an integer valued state space but approximates the number of events over dt.

The design of step_PTS is from: Wilkinson, D. J. (2011). Stochastic modeling for systems biology. CRC press

Elements of the N list come from two places: The stoichiometry matrix (S) is generated in spn_S and the hazards (h) come from spn_hazards.

For other samplers, see: step_CLE, step_DM, step_ODE

Value

function closure for use in sim_trajectory_R or sim_trajectory_CSV

---

step_PTS_decoupled *Make Poisson Time-Step (PTS) Sampler for a SPN Model*

Description

Make a function closure to implement a Poisson time-step (tau-leaping with fixed tau) sampler for a SPN.

Usage

```
step_PTS_decoupled(S, Sout, haz, dt = 0.01, maxhaz = 1e+06, human_ode = "SIS")
```

Arguments

S	a stoichiometry Matrix-class object
Sout	an optional matrix to track of event firings
haz	a list of hazard functions
dt	time-step for tau-leap method
maxhaz	maximum allowable hazard
human_ode	ode used for human states

Details

This sampling algorithm is based on representing a SPN as a set of competing Poisson processes; it thus uses an integer valued state space but approximates the number of events over dt.

The design of step_PTS is from: Wilkinson, D. J. (2011). Stochastic modeling for systems biology. CRC press

Elements of the N list come from two places: The stoichiometry matrix (S) is generated in spn_S and the hazards (h) come from spn_hazards.

For other samplers, see: step_CLE, step_DM, step_ODE

Value

function closure for use in sim_trajectory_R or sim_trajectory_CSV

summarize_eggs_geno *Summarize Eggs by Genotype*

Description

This function summarizes egg stage by genotype. It calls base_aquatic_geno to do all of the work.

Usage

```
summarize_eggs_geno(out, spn_P)
```

Arguments

out	the output of sim_trajectory_R
spn_P	the places of the SPN, see details

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, genotype, and value.

For examples of using this function, see: vignette("lifecycle-node", package = "MGDrivE2")

Value

a 3 to 5 column dataframe for plotting with ggplot2

summarize_eggs_stage *Summarize Eggs by Erlang-Stage*

Description

This function summarizes egg stage by Erlang-stages. It calls base_aquatic_stage to do all of the work.

Usage

```
summarize_eggs_stage(out, spn_P)
```

Arguments

out	the output of sim_trajectory_R
spn_P	the places of the SPN, see details

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, Erlang-stage, and value.

For examples of using this function, see: vignette("lifecycle-node", package = "MGDrivE2")

Value

a 3 to 5 column dataframe for plotting with ggplot2

---

summarize_females *Summarize Adult Females (One Node or Metapopulation Network, Lifecycle Model)*

Description

For MGDrivE2 simulations of mosquito lifecycle dynamics in a single node or metapopulation network, this function sums over the male mate genotype to get population trajectories of adult female mosquitoes by their genotype.

Usage

```
summarize_females(out, spn_P)
```

Arguments

out	the output of sim_trajectory_R
spn_P	the places of the SPN, see details

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node or spn_P_lifecycle_network.

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, genotype, and value.

For examples of using this function, this or any vignette which visualizes output: vignette("lifecycle-node", package = "MGDrivE2")

Value

a 3 to 5 column dataframe for plotting with ggplot2

summarize_females_epi *Summarize Adult Females (One Node or Metapopulation Network, SEI Mosquitoes)*

Description

For MGDrivE2 simulations of mosquito epidemiological dynamics in a single node or metapopulation network, this function sums over the male mate genotype as well as EIP bins to get population trajectories of adult female mosquitoes by their genotype and (S,E,I) status.

Usage

```
summarize_females_epi(out, spn_P)
```

Arguments

out	the output of sim_trajectory_R
spn_P	the places of the SPN, see details

Details

The places (spn_P) object is generated from one of the following: spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, inf, genotype, and value.

For examples of using this function, this or any vignette which simulates epi dynamics: vignette("epi-node", package = "MGDrivE2")

Value

a 4 to 6 column dataframe for plotting with ggplot2

summarize_humans_epiImperial *Summarize Humans for Imperial Model*

Description

The Imperial model outputs six human states for each age compartment. This function accepts the output matrix and the desired index of an age compartment and returns the trajectory of all human states in that given age compartment (default 1)

Usage

```
summarize_humans_epiImperial(out, index = 1)
```

Arguments

out	the output of sim_trajectory_R
index	the desired age compartment for which to pull trajectory

Value

dataframe for plotting with ggplot2

---

summarize_humans_epiSEIR

Summarize Humans (One Node or Metapopulation Network, SEI Mosquitoes - SEIR Humans)

Description

For MGDrivE2 simulations of mosquito epidemiological dynamics in a node or network, this function summarizes human infection status, S, E, I, and R. It uses base_summarize_humans to do all of the work.

Usage

```
summarize_humans_epiSEIR(out)
```

Arguments

out	the output of sim_trajectory_R

Details

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, inf, genotype, and value.

For examples of using this function, see: vignette("seir-dynamics", package = "MGDrivE2")

Value

a 4 to 6 column dataframe for plotting with ggplot2

summarize_humans_epiSIS *Summarize Humans (One Node or Metapopulation Network, SEI Mosquitoes - SIS Humans)*

Description

For MGDrivE2 simulations of mosquito epidemiological dynamics in a node or network, this function summarizes human infection status, S and I. It uses base_summarize_humans to do all of the work.

Usage

```
summarize_humans_epiSIS(out)
```

Arguments

out	the output of sim_trajectory_R

Details

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, inf, genotype, and value.

For examples of using this function, see: vignette("epi-node", package = "MGDrivE2")

Value

a 4 to 6 column dataframe for plotting with ggplot2

summarize_larvae_geno *Summarize Larvae by Genotype*

Description

This function summarizes larval stage by genotype. It calls base_aquatic_geno to do all of the work.

Usage

```
summarize_larvae_geno(out, spn_P)
```

Arguments

out	the output of sim_trajectory_R
spn_P	the places of the SPN, see details

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, genotype, and value.

For examples of using this function, see: vignette("lifecycle-node", package = "MGDrivE2")

Value

a 3 to 5 column dataframe for plotting with ggplot2

`summarize_larvae_stage` *Summarize Larval by Erlang-Stage*

Description

This function summarizes larval stage by Erlang-stages. It calls base_aquatic_stage to do all of the work.

Usage

```
summarize_larvae_stage(out, spn_P)
```

Arguments

out	the output of sim_trajectory_R
spn_P	the places of the SPN, see details

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, Erlang-stage, and value.

For examples of using this function, see: vignette("lifecycle-node", package = "MGDrivE2")

Value

a 3 to 5 column dataframe for plotting with ggplot2

summarize_males *Summarize Adult Males (One Node or Metapopulation Network)*

Description

For MGDrivE2 simulations of mosquito lifecycle dynamics or human infection dynamics, in a node or metapopulation network, this function summarizes population trajectories of adult male mosquitoes by their genotype.

Usage

```
summarize_males(out)
```

Arguments

out	the output of sim_trajectory_R

Details

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, genotype, and value.

For examples of using this function, this or any vignette which visualizes output: vignette("lifecycle-node", package = "MGDrivE2")

Value

a 3 to 5 column dataframe for plotting with ggplot2

summarize_pupae_geno *Summarize Pupal by Genotype*

Description

This function summarizes pupal stage by genotype. It calls base_aquatic_geno to do all of the work.

Usage

```
summarize_pupae_geno(out, spn_P)
```

Arguments

out	the output of sim_trajectory_R
spn_P	the places of the SPN, see details

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, genotype, and value.

For examples of using this function, see: vignette("lifecycle-node", package = "MGDrivE2")

Value

a 3 to 5 column dataframe for plotting with ggplot2

---

summarize_pupae_stage *Summarize Pupal by Erlang-Stage*

Description

This function summarizes pupal stage by Erlang-stages. It calls base_aquatic_stage to do all of the work.

Usage

```
summarize_pupae_stage(out, spn_P)
```

Arguments

out	the output of sim_trajectory_R
spn_P	the places of the SPN, see details

Details

The places (spn_P) object is generated from one of the following: spn_P_lifecycle_node, spn_P_lifecycle_network, spn_P_epiSIS_node, spn_P_epiSIS_network, spn_P_epiSEIR_node, or spn_P_epiSEIR_network.

The return object depends on the data provided. If the simulation was only 1 node, then no node designation is returned. If only one repetition was performed, no rep designation is returned. Columns always returned include: time, Erlang-stage, and value.

For examples of using this function, see: vignette("lifecycle-node", package = "MGDrivE2")

Value

a 3 to 5 column dataframe for plotting with ggplot2

`summarize_stats_CSV` *Summary Statistics for MGDrivE2*

Description

This function reads in all repetitions for each patch and calculates either the mean, quantiles, or both. User chooses the quantiles, up to 4 decimal places, and enters them as a vector. Quantiles are calculated empirically. (order does not matter)

Usage

```
summarize_stats_CSV(
  read_dir,
  write_dir = read_dir,
  mean = TRUE,
  quantiles = NULL,
  spn_P,
  tmax,
  dt,
  rem_file = FALSE,
  verbose = TRUE
)
```

Arguments

`read_dir`	Directory to find repetition folders in
`write_dir`	Directory to write output
`mean`	Boolean, calculate mean or not. Default is TRUE
`quantiles`	Vector of quantiles to calculate. Default is NULL
`spn_P`	Places object, see details
`tmax`	The final time to end simulation
`dt`	The time-step at which to return output (**not** the time-step of the sampling algorithm)
`rem_file`	Remove original output? Default is FALSE
`verbose`	Chatty? Default is TRUE

Details

Given the read_dir, this function assumes the follow file structure:

- read_dir
  - repetition 1

    * M_0001.csv
    * M_0002.csv
    * FS_0001.csv
    * FS_0001.csv
    * ...

  - repetition 2
    * M_0001.csv
    * M_0002.csv
    * FS_0001.csv
    * FS_0001.csv
    * ...

  - repetition 3
  - ...

The places (`spn_P`) object is generated from one of the following: `spn_P_lifecycle_node`, `spn_P_lifecycle_network`, `spn_P_epiSIS_node`, `spn_P_epiSIS_network`, `spn_P_epiSEIR_node`, or `spn_P_epiSEIR_network`.

`t0`, `tt`, `dt` define the first sampling time, the last sampling time, and each sampling time in-between.

Output files are *.csv and contain the mean or quantile in the file name.

For more details about using this function to process CSV output see: `vignette("data-analysis", package = "MGDrivE2")`

Value

Writes output to files in write_dir

`summarize_stats_CSV_decoupled`
Summary Statistics for MGDrivE2 - Decoupled samples

Description

This function reads in all repetitions for each patch and calculates either the mean, quantiles, or both. User chooses the quantiles, up to 4 decimal places, and enters them as a vector. Quantiles are calculated empirically. (order does not matter)

Usage

```
summarize_stats_CSV_decoupled(
  read_dir,
  write_dir = read_dir,
  mean = TRUE,
  quantiles = NULL,
  spn_P,
  tmax,
  dt,
  human_states,
  rem_file = FALSE,
  verbose = TRUE
)
```

Arguments

read_dir	Directory to find repetition folders in
write_dir	Directory to write output
mean	Boolean, calculate mean or not. Default is TRUE
quantiles	Vector of quantiles to calculate. Default is NULL
spn_P	Places object, see details
tmax	The final time to end simulation
dt	The time-step at which to return output (**not** the time-step of the sampling algorithm)
human_states	human state distribution
rem_file	Remove original output? Default is FALSE
verbose	Chatty? Default is TRUE

Details

Given the read_dir, this function assumes the follow file structure:

- read_dir
  - repetition 1
    - M_0001.csv
    - M_0002.csv
    - FS_0001.csv
    - FS_0001.csv
    - ...
  - repetition 2
    - M_0001.csv
    - M_0002.csv

    * FS_0001.csv
    * FS_0001.csv
    * ...

  - repetition 3
  - ...

The places (`spn_P`) object is generated from one of the following: `spn_P_lifecycle_node`, `spn_P_lifecycle_network`, `spn_P_epiSIS_node`, `spn_P_epiSIS_network`, `spn_P_epiSEIR_node`, or `spn_P_epiSEIR_network`.

`t0`, `tt`, `dt` define the first sampling time, the last sampling time, and each sampling time in-between.

Output files are *.csv and contain the mean or quantile in the file name.

For more details about using this function to process CSV output see: `vignette("data-analysis", package = "MGDrivE2")`

Value

Writes output to files in write_dir

`track_hinf` *Make tracking matrix for human infection events*

Description

Create a matrix object for tracking incidence in human population to be passed to either `sim_trajectory_CSV` or `sim_trajectory_R`.

Usage

```
track_hinf(spn_T, S)
```

Arguments

`spn_T`	set of transitions
`S`	stoichiometry matrix

Details

The returned matrix can be passed to the `Sout` argument of `sim_trajectory_CSV` or `sim_trajectory_R`.

Value

a `sparseMatrix` object

Index

∗ **datasets**
mu_ts, 28

add_interventions, 3

base_aquatic_geno, *68*, *72*, *74*
base_aquatic_stage, *68*, *73*, *75*
base_summarize_humans, *71*, *72*
batch_migration, 4, *29–33*, *35*, *37*, *39*, *40*
batch_migration_stage, *4*

calc_move_rate, 5
convert_prevalence_to_eir, 6
cubeMendelian, *8*, *11*, *14*, *16*, *19*, *47*, *49*, *51–62*

equilibrium_Imperial_decoupled, 7
equilibrium_Imperial_decoupled_human, 7
equilibrium_lifeycle, 8, *11*, *13*, *14*, *16*, *17*, *19*, *20*, *48*, *49*, *54*, *55*, *61*, *62*
equilibrium_SEI_decoupled_human, 10
equilibrium_SEI_decoupled_mosy, 10
equilibrium_SEI_Imperial, 13
equilibrium_SEI_SEIR, *9*, *13*, 15, *20*, *51*, *52*, *57*, *58*
equilibrium_SEI_SIS, *9*, *17*, 18, *48*, *49*, *52*, *53*, *59*, *60*
events, *29–33*, *35*, *37*, *39*, *41*

get_shape, 20

human_Imperial_ODE, 21

imperial_model_param_list_create, 22

make_Q_Imperial, 26
make_Q_SEI, *11*, *16*, *19*, 27
movement_prob2rate, 28
mu_ts, 28

ode, *35*, *37*, *39*, *41*, *65*, *66*

sim_trajectory_base_CSV, 29
sim_trajectory_base_CSV_decoupled, 30
sim_trajectory_base_R, 31
sim_trajectory_base_R_decoupled_Imperial, 32
sim_trajectory_base_R_decoupled_SIS, 33
sim_trajectory_CSV, *4*, *29*, *30*, 34, *39*, *41*, *42*, *44*, *45*, *63–67*, *79*
sim_trajectory_CSV_decoupled, 36
sim_trajectory_R, *4*, *31–33*, *35*, *37*, 38, *63–75*, *79*
sim_trajectory_R_decoupled, 39
solve_muAqua, 41
sparseMatrix, *79*
split_aggregate_CSV, 42
split_aggregate_CSV_decoupled, 45
spn_hazards, *35*, *37*, *39*, *41*, 47, *57–67*
spn_hazards_decoupled, 48
spn_P_epi_decoupled_node, 53, *60*
spn_P_epiSEIR_network, *9*, *12*, *14*, *16*, *19*, *44*, *46*, *48–51*, 51, *56*, *57*, *68–70*, *73*, *75*, *77*, *79*
spn_P_epiSEIR_node, *9*, *12*, *14*, *16*, *19*, *44*, *46*, *48–51*, 51, *56–58*, *68–70*, *73*, *75*, *77*, *79*
spn_P_epiSIS_network, *9*, *12*, *14*, *16*, *19*, *44*, *46*, *48–50*, 52, *53*, *56*, *58*, *59*, *68–70*, *73*, *75*, *77*, *79*
spn_P_epiSIS_node, *9*, *12*, *14*, *16*, *19*, *44*, *46*, *48–50*, *52*, 53, *56*, *59*, *60*, *68–70*, *73*, *75*, *77*, *79*
spn_P_lifecycle_network, *9*, *12*, *14*, *16*, *19*, *44*, *46*, *48–50*, 54, *55*, *56*, *61*, *68*, *69*, *73*, *75*, *77*, *79*
spn_P_lifecycle_node, *9*, *12*, *14*, *16*, *19*, *44*, *46*, *48–50*, *54*, 55, *56*, *62*, *68*, *69*, *73*, *75*, *77*, *79*
spn_Post, 49, *55*

spn_Pre, 50, *55*
spn_S, *35*, *37*, *39*, *41*, 55, *63–67*
spn_T_epi_decoupled_node, 60
spn_T_epiSEIR_network, *48–50*, *56*, 56, *57*
spn_T_epiSEIR_node, *48–50*, *56*, 57
spn_T_epiSIS_network, *48–50*, *56*, 58, *59, 60*
spn_T_epiSIS_node, *48–50*, *56*, *58*, 59
spn_T_lifecycle_network, *48–50*, *56*, 61, *62*
spn_T_lifecycle_node, *48–50*, *56*, *61*, 62
step_CLE, *35*, *37*, *39*, *41*, *48, 49*, 63, *64–67*
step_DM, *35*, *37*, *39*, *41*, *48, 49*, *63*, 64, *65–67*
step_ODE, *35*, *37*, *39*, *41*, *48, 49*, *63, 64*, 64, *66, 67*
step_ODE_decoupled, 65
step_PTS, *35*, *37*, *39*, *41*, *48, 49*, *63–66*, 66
step_PTS_decoupled, 67
summarize_eggs_geno, 68
summarize_eggs_stage, 68
summarize_females, 69
summarize_females_epi, 70
summarize_humans_epiImperial, 70
summarize_humans_epiSEIR, 71
summarize_humans_epiSIS, 72
summarize_larvae_geno, 72
summarize_larvae_stage, 73
summarize_males, 74
summarize_pupae_geno, 74
summarize_pupae_stage, 75
summarize_stats_CSV, 76
summarize_stats_CSV_decoupled, 77

track_hinf, *35*, *37*, *39*, *41*, 79

uniroot, *41*