Package: malariasimulation (via r-universe)

October 3, 2024

Title An individual based model for malaria

Version 2.0.0

Description Specifies the latest and greatest malaria model.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Remotes mrc-ide/malariaEquilibrium, mrc-ide/individual

Imports individual (>= 0.1.17), malariaEquilibrium (>= 1.0.1), Rcpp, statmod, MASS, dqrng (>= 0.4), sitmo, BH, R6, progress

Suggests testthat, mockery, knitr, rmarkdown, reshape2, DiagrammeR, cowplot, ggplot2, covr, mgcv

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

LinkingTo Rcpp, individual, BH, testthat, dqrng, sitmo

VignetteBuilder knitr

Depends R (>= 2.10)

Repository https://mrc-ide.r-universe.dev

RemoteUrl https://github.com/mrc-ide/malariasimulation

RemoteRef master

RemoteSha 9b4544105cc2ff47122d56d95e11c547900f2bd7

Contents

`AL_params` *Preset parameters for the AL drug*

Description

From SI of Commun. 5:5606 doi: 10.1038/ncomms6606 (2014)

Usage

```
AL_params
```

Format

An object of class `numeric` of length 4.

Details

Use a vector of preset parameters for the AL drug (artemether-lumefantrine)

Default parameters, from L to R, are: drug_efficacy: 0.95, drug_rel_c: 0.05094, drug_prophylaxis_shape: 11.3, drug_prophylaxis_scale: 10.6

`arab_params` *Preset parameters for the An. arabiensis vector*

Description

Preset parameters for the An. arabiensis vector

Usage

```
arab_params
```

Format

An object of class `list` of length 7.

Details

Default parameters: species: "arab" blood_meal_rates: 0.3333333 foraging_time: .69 Q0: 0.71 phi_bednets: 0.8 phi_indoors: 0.86 mum: 0.132

parameters from: https://www.pnas.org/content/pnas/early/2019/07/02/1820646116.full.pdf `calculate_carrying_capacity` *Calculate the vector carrying capacity*

Description

taken from "Modelling the impact of vector control interventions on Anopheles gambiae population dynamics"

Usage

```
calculate_carrying_capacity(parameters, m, species)
```

Arguments

`parameters`	model parameters
`m`	number of adult mosquitoes
`species`	index of the species to calculate for

`CorrelationParameters` *Class: Correlation parameters*

Description

Class: Correlation parameters

Class: Correlation parameters

Details

This class implements functionality that allows interventions to be correlated, positively or negatively. By default, interventions are applied independently and an individual's probability of receiving two interventions (either two separate interventions or two rounds of the same one) is the product of the probability of receiving each one.

By setting a positive correlation between two interventions, we can make it so that the individuals that receive intervention A are more likely to receive intervention B. Conversely, a negative correlation will make it such that individuals that receive intervention A are less likely to also receive intervention B.

Broadly speaking, the implementation works by assigning at startup a weight to each individual and intervention pair, reflecting how likely an individual is to receive that intervention. Those weights are derived stochastically from the configured correlation parameters.

For a detailed breakdown of the calculations, see Protocol S2 of Griffin et al. (2010). Derive the mvnorm from the configured correlations.

If a `restored_mvnorm` is specified, its columns (corresponding to restored interventions) will be re-used as is. Missing columns (for new interventions) are derived in accordance with the restored data.

Methods

Public methods:

- `CorrelationParameters$new()`
- `CorrelationParameters$inter_round_rho()`
- `CorrelationParameters$inter_intervention_rho()`
- `CorrelationParameters$sigma()`
- `CorrelationParameters$mvnorm()`
- `CorrelationParameters$save_state()`
- `CorrelationParameters$restore_state()`
- `CorrelationParameters$clone()`

Method `new()`**:** initialise correlation parameters

Usage:

```
CorrelationParameters$new(population, interventions)
```

Arguments:

`population` popularion size

`interventions` character vector with the name of enabled interventions

Method `inter_round_rho()`**:** Add rho between rounds

Usage:

```
CorrelationParameters$inter_round_rho(int, rho)
```

Arguments:

`int` string representing the intervention to update

`rho` value between 0 and 1 representing the correlation between rounds of the intervention

Method `inter_intervention_rho()`**:** Add rho between interventions

Usage:

```
CorrelationParameters$inter_intervention_rho(int_1, int_2, rho)
```

Arguments:

`int_1` string representing the first intervention

`int_2` string representing the second intervention (intechangable with int_1)

`rho` value between -1 and 1 representing the correlation between rounds of the intervention

Method `sigma()`**:** Standard deviation of each intervention between rounds

Usage:

```
CorrelationParameters$sigma()
```

Method `mvnorm()`**:** multivariate norm draws for these parameters

Usage:

```
CorrelationParameters$mvnorm()
```

Method `save_state()`**:** Save the correlation state.

Usage:

```
CorrelationParameters$save_state()
```

Method `restore_state()`**:** Restore the correlation state.

Only the randomly drawn weights are restored. The object needs to be initialized with the same rhos.

Usage:

```
CorrelationParameters$restore_state(timestep, state)
```

Arguments:

`timestep` the timestep at which simulation is resumed. This parameter's value is ignored, it only exists to conform to a uniform interface.

`state` a previously saved correlation state, as returned by the save_state method.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
CorrelationParameters$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`create_pev_profile` *create a PEV profile*

Description

creates a data structure for holding pre-erythrocytic vaccine profile parameters. Parameters are validated on creation.

Usage

```
create_pev_profile(vmax, alpha, beta, cs, rho, ds, dl)
```

Arguments

`vmax`	maximum efficacy of the vaccine
`alpha`	shape parameter for the vaccine efficacy model
`beta`	scale parameter for the vaccine efficacy model
`cs`	peak parameters for the antibody model (vector of mean and std. dev)
`rho`	delay parameters for the antibody model (vector of mean and std. dev)
`ds`	delay parameters for the antibody model, short-term waning (vector of mean and std. dev)
`dl`	delay parameters for the antibody model, long-term waning (vector of mean and std. dev)

create_progress_process

Create progress process

Description

Create progress process

Usage

```
create_progress_process(timesteps)
```

Arguments

`timesteps`	Simulation timesteps

Value

Progress bar process function

DHA_PQP_params

Preset parameters for the DHA-PQP drug

Description

From SI of Commun. 5:5606 doi: 10.1038/ncomms6606 (2014)

Usage

```
DHA_PQP_params
```

Format

An object of class `numeric` of length 4.

Details

Use a vector of preset parameters for the DHA-PQP drug (dihydroartemisinin-piperaquine)

Default parameters, from L to R, are: drug_efficacy: 0.95, drug_rel_c: 0.09434, drug_prophylaxis_shape: 4.4, drug_prophylaxis_scale: 28.1

`find_birthrates` *Calculate the birthrate for a population in equilibrium*

Description

Calculate the birthrate for a population in equilibrium

Usage

```
find_birthrates(pops, age_high, deathrates)
```

Arguments

`pops`	a vector of populations
`age_high`	a vector of age groups
`deathrates`	vector of deathrates for each age group

`fun_params` *Preset parameters for the An. funestus vector*

Description

Preset parameters for the An. funestus vector

Usage

```
fun_params
```

Format

An object of class `list` of length 7.

Details

Default parameters: species: "fun" blood_meal_rates: 0.3333333 foraging_time: .69 Q0: 0.94 phi_bednets: 0.78 phi_indoors: 0.87 mum: 0.112

parameters from: https://www.pnas.org/content/pnas/early/2019/07/02/1820646116.full.pdf

gamb_params *Preset parameters for the An. gambiae s.s vector*

Description

Preset parameters for the An. gambiae s.s vector

Usage

```
gamb_params
```

Format

An object of class `list` of length 7.

Details

Default parameters: species: "gamb" blood_meal_rates: 0.3333333 foraging_time: .69 Q0: 0.92 phi_bednets: 0.85 phi_indoors: 0.90 mum: 0.132

parameters from: https://www.pnas.org/content/pnas/early/2019/07/02/1820646116.full.pdf

get_antimalarial_resistance_parameters *Retrieve resistance parameters*

Description

Retrieve the resistance parameters associated with the drug each individual receiving clinical treatment has been administered in the current time step.

Usage

```
get_antimalarial_resistance_parameters(parameters, drugs, timestep)
```

Arguments

`parameters`	the model parameters
`drugs`	vector of integers representing the drugs administered to each individual receiving treatment
`timestep`	the current time step

get_correlation_parameters

Get default correlation parameters

Description

returns a `CorrelationParameters` object for you edit. By default, all correlations are set to 0

Usage

```
get_correlation_parameters(parameters)
```

Arguments

`parameters`	model parameters

Examples

```
# get the default model parameters
parameters <- get_parameters()

# Set some vaccination strategy
parameters <- set_mass_pev(
  parameters,
  profile = rtss_profile,
  timesteps = 100,
  coverages = .9,
  min_wait = 0,
  min_ages = 100,
  max_ages = 1000,
  booster_spacing = numeric(0),
  booster_coverage = numeric(0),
  booster_profile = NULL
)

# Set some smc strategy
parameters <- set_drugs(parameters, list(SP_AQ_params))
parameters <- set_smc(
  parameters,
  drug = 1,
  timesteps = 100,
  coverages = .9,
  min_age = 100,
  max_age = 1000
)

# Correlate the vaccination and smc targets
correlations <- get_correlation_parameters(parameters)
correlations$inter_intervention_rho('pev', 'smc', 1)
```

```
# Correlate the rounds of smc
correlations$inter_round_rho('smc', 1)

# You can now pass the correlation parameters to the run_simulation function
```

get_init_carrying_capacity *Get initialised carrying capacity for each species*

Description

Get initialised carrying capacity for each species

Usage

```
get_init_carrying_capacity(parameters)
```

Arguments

parameters	the model parameters

Value

a vector of carrying initialised carrying capacity estimates for each vector species

get_parameters *Get model parameters*

Description

get_parameters creates a named list of parameters for use in the model. These parameters are passed to process functions. These parameters are explained in "The US President's Malaria Initiative, Plasmodium falciparum transmission and mortality: A modelling study."

Usage

```
get_parameters(overrides = list())
```

Arguments

`overrides` a named list of parameter values to use instead of defaults The parameters are defined below.

initial state proportions:

- s_proportion - the proportion of `human_population` that begin as susceptible; default = 0.420433246
- d_proportion - the proportion of `human_population` that begin with clinical disease; default = 0.007215064
- a_proportion - the proportion of `human_population` that begin as asymptomatic; default = 0.439323667
- u_proportion - the proportion of `human_population` that begin as subpatents; default = 0.133028023
- t_proportion - the proportion of `human_population` that begin treated; default = 0

human fixed state transitions:

- dd - the delay for humans to move from state D to A; default = 5
- dt - the delay for humans to move from state Tr to S; default = 5
- da - the delay for humans to move from state A to U; default = 195
- du - the delay for humans to move from state U to S; default = 110

human demography parameters:

- human_population - the initial number of humans to model; default = 100
- average_age - the average age of humans (in timesteps), this is only used if custom_demography is FALSE; default = 7665
- custom_demography - population demography given; default = FALSE

initial immunity values:

- init_icm - the immunity from clinical disease at birth; default = 0
- init_ivm - the immunity from severe disease at birth; default = 0
- init_ib - the initial pre-erythrocitic immunity; default = 0
- init_ica - the initial acquired immunity from clinical disease; default = 0
- init_iva - the initial acquired immunity from severe disease; default = 0
- init_id - the initial acquired immunity to detectability; default = 0

immunity decay rates:

- rm - decay rate for maternal immunity to clinical disease; default = 67.6952
- rvm - decay rate for maternal immunity to severe disease; default = 76.8365
- rb - decay rate for acquired pre-erythrocytic immunity; default = 3650
- rc - decay rate for acquired immunity to clinical disease; default = 10950
- rva - decay rate for acquired immunity to severe disease; default = 10950
- rid - decay rate for acquired immunity to detectability; default = 3650

immunity boost grace periods:

- ub - period in which pre-erythrocytic immunity cannot be boosted; default = 7.2

- uc - period in which clinical immunity cannot be boosted; default = 6.06
- uv - period in which severe immunity cannot be boosted; default = 11.4321
- ud - period in which immunity to detectability cannot be boosted; default = 9.44512

maternal immunity parameters:

- pcm - new-born clinical immunity relative to mother's; default = 0.774368
- pvm - new-born severe immunity relative to mother's; default = 0.195768

unique biting rate:

- a0 - age dependent biting parameter; default = 2920
- rho - age dependent biting parameter; default = 0.85
- sigma_squared - heterogeneity parameter; default = 1.67
- n_heterogeneity_groups - number discretised groups for heterogeneity, used for sampling mothers; default = 5

probability of pre-erythrocytic infection:

- b0 - maximum probability due to no immunity; default = 0.59
- b1 - maximum reduction due to immunity; default = 0.5
- ib0 - scale parameter; default = 43.9
- kb - shape parameter; default = 2.16

probability of detection by light-microscopy when asymptomatic:

- fd0 - time-scale at which immunity changes with age; default = 0.007055
- ad - scale parameter relating age to immunity; default = 7993.5
- gammad - shape parameter relating age to immunity; default = 4.8183
- d1 - minimum probability due to immunity; default = 0.160527
- id0 - scale parameter; default = 1.577533
- kd - shape parameter; default = 0.476614

probability of clinical infection:

- phi0 - maximum probability due to no immunity; default = 0.792
- phi1 - maximum reduction due to immunity; default = 0.00074
- ic0 - scale parameter; default = 18.02366
- kc - shape parameter; default = 2.36949

probability of severe infection:

- theta0 - maximum probability due to no immunity; default = 0.0749886
- theta1 - maximum reduction due to immunity; default = 0.0001191
- iv0 - scale parameter; default = 1.09629
- kv - shape parameter; default = 2.00048
- fv0 - age dependent modifier; default = 0.141195
- av - age dependent modifier; default = 2493.41
- gammav - age dependent modifier; default = 2.91282

infectivity towards mosquitoes:

- cd - infectivity of clinically diseased humans towards mosquitoes; default = 0.068

- gamma1 - parameter for infectivity of asymptomatic humans; default = 1.82425
- cu - infectivity of sub-patent infection; default = 0.0062
- ct - infectivity of treated infection; default = 0.021896

mosquito fixed state transitions (including mortality):

- del - the delay for mosquitoes to move from state E to L; default = 6.64
- dl - the delay for mosquitoes to move from state L to P; default = 3.72
- dpl - the delay mosquitoes to move from state P to Sm; default = 0.643
- me - early stage larval mortality rate; default = 0.0338
- ml - late stage larval mortality rate; default = 0.0348
- mup - the rate at which pupal mosquitoes die; default = 0.249
- mum - the rate at which developed mosquitoes die; default (An. gambiae) = .132

vector biology: species specific values are vectors please set species parameters using the convenience function `set_species`

- beta - the average number of eggs laid per female mosquito per day; default = 21.2
- total_M - the initial number of adult mosquitos in the simulation; default = 1000
- init_foim - the FOIM used to calculate the equilibrium state for mosquitoes; default = 0
- species - names of the species in the simulation; default = "gamb"
- species_proportions - the relative proportions of each species; default = 1
- blood_meal_rates - the blood meal rates for each species; default = 1/3
- Q0 - proportion of blood meals taken on humans; default = 0.92
- foraging_time - time spent taking blood meals; default = 0.69

seasonality and carrying capacity parameters: please set flexible carrying capacity using the convenience function `set_carrying_capacity`

- model_seasonality - boolean switch TRUE iff the simulation models seasonal rainfall; default = FALSE
- g0 - rainfall fourier parameter; default = 2
- g - rainfall fourier parameter; default = 0.3, 0.6, 0.9
- h - rainfall fourier parameters; default = 0.1, 0.4, 0.7
- gamma - effect of density dependence on late instars relative to early instars; default = 13.25
- rainfall_floor - the minimum rainfall value (must be above 0); default 0.001
- carrying_capacity; default = FALSE
- carrying_capacity_timesteps; default = NULL
- carrying_capacity_values; default = NULL#'

parasite incubation periods:

- de - duration of the human latent period of infection; default = 12
- delay_gam - lag from parasites to infectious gametocytes; default = 12.5

- dem - extrinsic incubation period in mosquito population model; default = 10

treatment parameters: please set treatment parameters with the convenience functions `set_drugs` and `set_clinical_treatment`

- drug_efficacy - a vector of efficacies for available drugs; default = turned off
- drug_rel_c - a vector of relative onward infectiousness values for drugs; default = turned off
- drug_prophylaxis_shape - a vector of shape parameters for weibull curves to model prophylaxis for each drug; default = turned off
- drug_prophylaxis_scale - a vector of scale parameters for weibull curves to model prophylaxis for each drug; default = turned off
- clinical_treatment_drugs - a vector of drugs that are available for clinically diseased (these values refer to the index in drug_* parameters); default = NULL, NULL, NULL
- clinical_treatment_coverage - a vector of coverage values for each drug; default = NULL, NULL, NULL

MDA, SMC and PMC parameters: please set these parameters with the convenience functions `set_mda`, `set_smc` and `set_pmc`, with `peak_season_offset`

bednet, irs and mosquito feeding cycle parameters: please set vector control strategies using `set_bednets` and `set_spraying`

- bednets - boolean for if bednets are enabled; default = FALSE
- phi_bednets - proportion of bites taken in bed; default = 0.85
- k0 - proportion of females bloodfed with no net; default = 0.699
- spraying - boolean for if indoor spraying is enabled; default = FALSE
- phi_indoors - proportion of bites taken indoors; default = 0.90

PEV parameters: please set vaccine strategies with the convenience functions `set_pev_epi` and `set_mass_pev`

- pev_doses - the dosing schedule before the vaccine takes effect; default = c(0, 1.5 * 30, 3 * 30) default = 365

TBV parameters: please set TBV parameters with the convenience functions in `set_tbv`

- tbv_mt - effect on treated infectiousness; default = 35
- tbv_md - effect on diseased infectiousness; default = 46.7
- tbv_ma - effect on asymptomatic infectiousness; default = 3.6
- tbv_mu - effect on subpatent infectiousness; default = 0.8
- tbv_k - scale parameter for effect on infectiousness; default = 0.9
- tbv_tau - peak antibody parameter; default = 22
- tbv_rho - antibody component parameter; default = 0.7
- tbv_ds - antibody short-term delay parameter; default = 45
- tbv_dl - antibody long-term delay parameter; default = 591
- tbv_tra_mu - transmission reduction parameter; default = 12.63
- tbv_gamma1 - transmission reduction parameter; default = 2.5

- tbv_gamma2 - transmission reduction parameter; default = 0.06

Antimalarial resistance parameters: please set antimalarial resistance parameters with the convenience functions in `set_antimalarial_resistance`

- antimalarial_resistance - boolean for if antimalarial resistance is enabled; default = FALSE
- antimalarial_resistance_drug - vector of drugs for which resistance can be parameterised; default = NULL
- antimalarial_resistance_timesteps - vector of time steps on which resistance updates occur; default = NULL
- artemisinin_resistant_proportion - vector of proportions of infections resistant to the artemisinin component of a given drug; default = NULL
- partner_drug_resistance_proportion - vector of proportions of infections resistant to the parter drug component of a given drug; default = NULL
- slow_parasite_clearance_probability - vector of probabilities of slow parasite clearance for a given drug; default = NULL
- early_treatment_failure_probability - vector of probabilities of early treatment failure for a given drug; default = NULL
- late_clinical_failure_probability - vector of probabilities of late clinical failure for a given drug; default = NULL
- late_parasitological_failure_probability - vector of probabilities of late parasitological failure for a given drug; default = NULL
- reinfection_during_prophylaxis_probability - vector of probabilities of reinfection during prophylaxis for a given drug; default = NULL
- dt_slow_parasite_clearance - the delay for humans experiencing slow parasite clearance to move from state Tr to S; default = NULL

rendering: All values are in timesteps and all ranges are inclusive. Please set rendered age groups using the convenience function

- age_group_rendering_min_ages - the minimum ages for population size outputs; default = turned off
- age_group_rendering_max_ages - the corresponding max ages; default = turned off
- incidence_rendering_min_ages - the minimum ages for incidence outputs (includes asymptomatic microscopy +); default = turned off
- incidence_rendering_max_ages - the corresponding max ages; default = turned off
- clinical_incidence_rendering_min_ages - the minimum ages for clinical incidence outputs (symptomatic); default = 0
- clinical_incidence_rendering_max_ages - the corresponding max ages; default = 1825
- severe_incidence_rendering_min_ages - the minimum ages for severe incidence outputs; default = turned off
- severe_incidence_rendering_max_ages - the corresponding max ages; default = turned off
- prevalence_rendering_min_ages - the minimum ages for clinical prevalence outputs; default = 730

- prevalence_rendering_max_ages - the corresponding max ages; default = 3650

mixing:

- rdt_intercept - the y intercept for the log logit relationship betweeen rdt and PCR prevalence; default = -0.968
- rdt_coeff - the coefficient for the log logit relationship betweeen rdt and PCR prevalence; default = 1.186

miscellaneous:

- mosquito_limit - the maximum number of mosquitoes to allow for in the simulation; default = 1.00E+05
- individual_mosquitoes - boolean whether adult mosquitoes are modeled individually or compartmentally; default = TRUE
- human_population_timesteps - the timesteps at which the population should change; default = 0
- r_tol - the relative tolerance for the ode solver; default = 1e-4
- a_tol - the absolute tolerance for the ode solver; default = 1e-4
- ode_max_steps - the max number of steps for the solver; default = 1e6
- enable_heterogeneity - boolean whether to include heterogeneity in biting rates; default = TRUE

`parameterise_mosquito_equilibrium`

Parameterise total_M and carrying capacity for mosquitos from EIR

Description

NOTE: the inital EIR is likely to change unless the rest of the model is in equilibrium. NOTE: please set seasonality first, since the mosquito_limit will estimate an upper bound from the peak season.

max_total_M is calculated using the equilibrium solution from "Modelling the impact of vector control interventions on Anopheles gambiae population dynamics"

Usage

```
parameterise_mosquito_equilibrium(parameters, EIR)
```

Arguments

`parameters`	the parameters to modify
`EIR`	to work from

parameterise_total_M *Parameterise total_M*

Description

Sets total_M and an upper bound for the number of mosquitoes in the simulation. NOTE: please set seasonality first, since the mosquito_limit will estimate an upper bound from the peak season.

Usage

```
parameterise_total_M(parameters, total_M)
```

Arguments

`parameters`	the parameters to modify
`total_M`	the initial adult mosquitoes in the simulation

parameter_draws *Parameter draws*

Description

1000 draws from the joint posterior fit from

Usage

```
parameter_draws
```

Format

`parameter_draws`**:**
A list of lists of length 1000, each level contains a list of drawn parameters

Source

https://www.nature.com/articles/ncomms4136

`peak_season_offset` *Calculate the yearly offset (in timesteps) for the peak mosquito season*

Description

Calculate the yearly offset (in timesteps) for the peak mosquito season

Usage

```
peak_season_offset(parameters)
```

Arguments

`parameters`	to work from

`r21_booster_profile` *R21 booster vaccine profile*

Description

Parameters for a booster dose of R21 for use with the set_mass_pev and set_pev_epi functions (Schmit + Topazian et al. 2022 Lancet ID)

Usage

```
r21_booster_profile
```

Format

An object of class `list` of length 7.

`r21_profile` *R21 vaccine profile*

Description

Parameters for a primary dose of R21 for use with the set_mass_pev and set_pev_epi functions (Schmit + Topazian et al. 2022 Lancet ID)

Usage

```
r21_profile
```

Format

An object of class `list` of length 7.

`rtss_booster_profile` *RTS,S booster vaccine profile*

Description

Parameters for a booster dose of RTS,S for use with the set_mass_pev and set_pev_epi functions (White MT et al. 2015 Lancet ID)

Usage

```
rtss_booster_profile
```

Format

An object of class `list` of length 7.

`rtss_profile` *RTS,S vaccine profile*

Description

Parameters for a primary dose of RTS,S for use with the set_mass_pev and set_pev_epi functions (White MT et al. 2015 Lancet ID)

Usage

```
rtss_profile
```

Format

An object of class `list` of length 7.

run_metapop_simulation *Run a metapopulation model*

Description

Run a metapopulation model

Usage

```
run_metapop_simulation(
  timesteps,
  parameters,
  correlations = NULL,
  mixing_tt,
  export_mixing,
  import_mixing,
  p_captured_tt,
  p_captured,
  p_success
)
```

Arguments

`timesteps`	the number of timesteps to run the simulation for (in days)
`parameters`	a list of model parameter lists for each population
`correlations`	a list of correlation parameters for each population (default: NULL)
`mixing_tt`	a vector of time steps for each mixing matrix
`export_mixing`	a list of matrices of coefficients for exportation of infectivity. Rows = origin sites, columns = destinations. Each matrix element describes the mixing pattern from destination to origin. Each matrix element must be between 0 and 1. Each matrix is activated at the corresponding timestep in mixing_tt
`import_mixing`	a list of matrices of coefficients for importation of infectivity.
`p_captured_tt`	a vector of time steps for each p_captured matrix
`p_captured`	a list of matrices representing the probability that travel between sites is intervened by a test and treat border check. Dimensions are the same as for `export_mixing`
`p_success`	the probability that an individual who has tested positive (through an RDT) successfully clears their infection through treatment

Value

a list of dataframe of model outputs as in run_simulation

run_resumable_simulation *Run the simulation in a resumable way*

Description

this function accepts an initial simulation state as an argument, and returns the final state after running all of its timesteps. This allows one run to be resumed, possibly having changed some of the parameters.

Usage

```
run_resumable_simulation(
  timesteps,
  parameters = NULL,
  correlations = NULL,
  initial_state = NULL,
  restore_random_state = FALSE
)
```

Arguments

`timesteps`	the timestep at which to stop the simulation
`parameters`	a named list of parameters to use
`correlations`	correlation parameters
`initial_state`	the state from which the simulation is resumed
`restore_random_state`	if TRUE, restore the random number generator's state from the checkpoint.

Value

a list with two entries, one for the dataframe of results and one for the final simulation state.

run_simulation *Run the simulation*

Description

Run the simulation for some time given some parameters. This currently returns a dataframe with the number of individuals in each state at each timestep.

The resulting dataframe contains the following columns:

- timestep: the timestep for the row
- infectivity: the infectivity from humans towards mosquitoes

- FOIM: the force of infection towards mosquitoes (per species)
- mu: the death rate of adult mosquitoes (per species)
- EIR: the Entomological Inoculation Rate (per timestep, per species, over the whole population)
- n_bitten: number of humans bitten by an infectious mosquito
- n_treated: number of humans treated for clinical or severe malaria this timestep
- n_infections: number of humans who get an asymptomatic, clinical or severe malaria this timestep
- natural_deaths: number of humans who die from aging
- S_count: number of humans who are Susceptible
- A_count: number of humans who are Asymptomatic
- D_count: number of humans who have the clinical malaria
- U_count: number of subpatent infections in humans
- Tr_count: number of detectable infections being treated in humans
- ica_mean: the mean acquired immunity to clinical infection over the population of humans
- icm_mean: the mean maternal immunity to clinical infection over the population of humans
- ib_mean: the mean blood immunity to all infection over the population of humans
- id_mean: the mean immunity from detection through microscopy over the population of humans
- n: number of humans between an inclusive age range at this timestep. This defaults to n_730_3650. Other age ranges can be set with prevalence_rendering_min_ages and prevalence_rendering_max_ages parameters.
- n_detect_lm (or pcr): number of humans with an infection detectable by microscopy (or pcr) between an inclusive age range at this timestep. This defaults to n_detect_730_3650. Other age ranges can be set with prevalence_rendering_min_ages and prevalence_rendering_max_ages parameters.
- p_detect_lm (or pcr): the sum of probabilities of detection by microscopy (or pcr) between an inclusive age range at this timestep. This defaults to p_detect_730_3650. Other age ranges can be set with prevalence_rendering_min_ages and prevalence_rendering_max_ages parameters.
- n_inc: number of new infections for humans between an inclusive age range at this timestep. incidence columns can be set with incidence_rendering_min_ages and incidence_rendering_max_ages parameters.
- p_inc: sum of probabilities of infection for humans between an inclusive age range at this timestep. incidence columns can be set with incidence_rendering_min_ages and incidence_rendering_max_ages parameters.
- n_inc_clinical: number of new clinical infections for humans between an inclusive age range at this timestep. clinical incidence columns can be set with clinical_incidence_rendering_min_ages and clinical_incidence_rendering_max_ages parameters.
- p_inc_clinical: sub of probabilities of clinical infection for humans between an inclusive age range at this timestep. clinical incidence columns can be set with clinical_incidence_rendering_min_ages and clinical_incidence_rendering_max_ages parameters.

- n_inc_severe: number of new severe infections for humans between an inclusive age range at this timestep. severe incidence columns can be set with severe_incidence_rendering_min_ages and severe_incidence_rendering_max_ages parameters.
- p_inc_severe: the sum of probabilities of severe infection for humans between an inclusive age range at this timestep. severe incidence columns can be set with severe_incidence_rendering_min_ages and severe_incidence_rendering_max_ages parameters.
- E_count: number of mosquitoes in the early larval stage (per species)
- L_count: number of mosquitoes in the late larval stage (per species)
- P_count: number of mosquitoes in the pupal stage (per species)
- Sm_count: number of adult female mosquitoes who are Susceptible (per species)
- Pm_count: number of adult female mosquitoes who are incubating (per species)
- Im_count: number of adult female mosquitoes who are infectious (per species)
- rate_D_A: rate that humans transition from clinical disease to asymptomatic
- rate_A_U: rate that humans transition from asymptomatic to subpatent
- rate_U_S: rate that humans transition from subpatent to susceptible
- net_usage: the number people protected by a bed net
- mosquito_deaths: number of adult female mosquitoes who die this timestep
- n_drug_efficacy_failures: number of clinically treated individuals whose treatment failed due to drug efficacy
- n_early_treatment_failure: number of clinically treated individuals who experienced early treatment failure
- n_successfully_treated: number of clinically treated individuals who are treated successfully (includes individuals who experience slow parasite clearance)
- n_slow_parasite_clearance: number of clinically treated individuals who experienced slow parasite clearance

Usage

```
run_simulation(timesteps, parameters = NULL, correlations = NULL)
```

Arguments

`timesteps`	the number of timesteps to run the simulation for (in days)
`parameters`	a named list of parameters to use
`correlations`	correlation parameters

Value

dataframe of results

run_simulation_with_repetitions *Run the simulation with repetitions*

Description

Run the simulation with repetitions

Usage

```
run_simulation_with_repetitions(
  timesteps,
  repetitions,
  overrides = list(),
  parallel = FALSE
)
```

Arguments

timesteps	the number of timesteps to run the simulation for
repetitions	n times to run the simulation
overrides	a named list of parameters to use instead of defaults
parallel	execute runs in parallel

set_antimalarial_resistance *Parameterise antimalarial resistance*

Description

Parameterise antimalarial resistance

Usage

```
set_antimalarial_resistance(
  parameters,
  drug,
  timesteps,
  artemisinin_resistance_proportion,
  partner_drug_resistance_proportion,
  slow_parasite_clearance_probability,
  early_treatment_failure_probability,
  late_clinical_failure_probability,
  late_parasitological_failure_probability,
  reinfection_during_prophylaxis_probability,
  slow_parasite_clearance_time
)
```

Arguments

`parameters` the model parameters

`drug` the index of the drug which resistance is being set, as set by the set_drugs() function, in the parameter list

`timesteps` vector of time steps for each update to resistance proportion and resistance outcome probability

`artemisinin_resistance_proportion`
vector of updates to the proportions of infections that are artemisinin resistant at time t

`partner_drug_resistance_proportion`
vector of updates to the proportions of infections that are partner-drug resistant at time t

`slow_parasite_clearance_probability`
vector of updates to the proportion of artemisinin-resistant infections that result in early treatment failure

`early_treatment_failure_probability`
vector of updates to the proportion of artemisinin-resistant infections that result in slow parasite clearance

`late_clinical_failure_probability`
vector of updates to the proportion of partner-drug-resistant infections that result in late clinical failure

`late_parasitological_failure_probability`
vector of updates to the proportion of partner-drug-resistant infections that result in late parasitological failure

`reinfection_during_prophylaxis_probability`
vector of updates to the proportion of partner-drug-resistant infections that result in reinfection during prophylaxis

`slow_parasite_clearance_time`
single value representing the mean time individual's experiencing slow parasite clearance reside in the treated state

`set_bednets` *Parameterise a bed net strategy*

Description

The model will distribute bed nets at `timesteps` to a random sample of the entire human population. The sample size will be a proportion of the human population taken from the corresponding `coverages`. The sample *can* contain humans who already have bed nets.

All of the sample "use" their bed nets on the timestep after they are distributed. Incomplete usage is not part of this model.

If a human in the sample already has a bed net, their bed net will be replaced by a new one.

Bed nets will be randomly removed each timestep with a rate of `1 - exp(-1/retention)`

The structure for the bed net model is documented in the S.I. of 10.1038/s41467-018-07357-w

Usage

```
set_bednets(parameters, timesteps, coverages, retention, dn0, rn, rnm, gamman)
```

Arguments

`parameters` a list of parameters to modify

`timesteps` the timesteps at which to distribute bed nets

`coverages` the proportion of the human population who receive bed nets

`retention` the average number of timesteps a net is kept for

`dn0` a matrix of death probabilities for each species over time. With nrows=length(timesteps), ncols=length(species)

`rn` a matrix of repelling probabilities for each species over time With nrows=length(timesteps), ncols=length(species)

`rnm` a matrix of minimum repelling probabilities for each species over time With nrows=length(timesteps), ncols=length(species)

`gamman` a vector of bednet half-lives for each distribution timestep

`set_carrying_capacity` *Parameterise custom baseline carrying capacity*

Description

Allows the user to set a completely flexible and custom carrying capacity for each species

Usage

```
set_carrying_capacity(parameters, timesteps, carrying_capacity_scalers)
```

Arguments

`parameters` the model parameters

`timesteps` vector of timesteps for each rescale change

`carrying_capacity_scalers`
matrix of scaling factors to scale the baseline carrying capacity for each species with nrows = length(timesteps), ncols = length(species)

set_clinical_treatment *Parameterise clinical treatment*

Description

Parameterise clinical treatment

Usage

```
set_clinical_treatment(parameters, drug, timesteps, coverages)
```

Arguments

`parameters`	the model parameters
`drug`	the index of the drug to set as a treatment
`timesteps`	vector of timesteps for each coverage change
`coverages`	vector of coverages for this drug

set_demography *Parameterise variable deathrates*

Description

Parameterise variable deathrates

Usage

```
set_demography(parameters, agegroups, timesteps, deathrates)
```

Arguments

`parameters`	the model parameters
`agegroups`	vector of agegroups (in timesteps)
`timesteps`	vector of timesteps for each change in demography
`deathrates`	matrix of deathrates per age group per timestep. Rows are timesteps from the `timesteps` param. Columns are the age groups from the `agegroups` param.

`set_drugs` *Parameterise drugs to use in the model*

Description

Parameterise drugs to use in the model

Usage

```
set_drugs(parameters, drugs)
```

Arguments

`parameters`	the model parameters
`drugs`	a list of drug parameters, can be set using presets

`set_epi_outputs` *Parameterise age grouped output rendering*

Description

Parameterise age grouped output rendering

Usage

```
set_epi_outputs(
  parameters,
  age_group = NULL,
  incidence = NULL,
  clinical_incidence = NULL,
  severe_incidence = NULL,
  prevalence = NULL,
  ica = NULL,
  icm = NULL,
  iva = NULL,
  ivm = NULL,
  id = NULL,
  ib = NULL
)
```

Arguments

`parameters` the model parameters

`age_group` age breaks for population size outputs; default = NULL

`incidence` age breaks for incidence outputs (D+Tr+A); default = NULL

`clinical_incidence` age breaks for clinical incidence outputs (symptomatic); default = c(0, 1825)

`severe_incidence` age breaks for severe incidence outputs; default = NULL

`prevalence` age breaks for clinical prevalence outputs (pcr and lm detectable infections); default = c(730, 3650)

`ica` age breaks for average acquired clinical immunity; default = NULL

`icm` age breaks for average maternal clinical immunity; default = NULL

`iva` age breaks for average acquired severe immunity; default = NULL

`ivm` age breaks for average maternal severe immunity; default = NULL

`id` age breaks for average immunity to detectability; default = NULL

`ib` age breaks for average blood immunity; default = NULL

Details

this function produces contiguous age groups, inclusive of the lower age limit and exclusive of the upper age limit: e.g., c(0, 10, 100) will produce two age groups: 0-9 and 10-99 in days.

`set_equilibrium` *Set equilibrium*

Description

This will update the IBM parameters to match the equilibrium parameters and set up the initial human and mosquito population to acheive init_EIR

Usage

```
set_equilibrium(parameters, init_EIR, eq_params = NULL)
```

Arguments

`parameters` model parameters to update

`init_EIR` the desired initial EIR (infectious bites per person per day over the entire human population)

`eq_params` parameters from the malariaEquilibrium package, if null. The default malariaEquilibrium parameters will be used

set_mass_pev *Parameterise a vaccine mass distribution strategy*

Description

distribute pre-erythrocytic vaccine to a population in an age range. Efficacy will take effect after the last dose

Usage

```
set_mass_pev(
  parameters,
  profile,
  timesteps,
  coverages,
  min_ages,
  max_ages,
  min_wait,
  booster_spacing,
  booster_coverage,
  booster_profile
)
```

Arguments

`parameters`	a list of parameters to modify
`profile`	a list of details for the vaccine profile, create with `create_pev_profile`
`timesteps`	a vector of timesteps for each round of vaccinations
`coverages`	the coverage for each round of vaccinations
`min_ages`	for the target population, inclusive (in timesteps)
`max_ages`	for the target population, inclusive (in timesteps)
`min_wait`	the minimum acceptable time since the last vaccination (in timesteps); When using both set_mass_pev and set_pev_epi, this represents the minimum time between an individual being vaccinated under one scheme and vaccinated under another.
`booster_spacing`	the timesteps (following the final primary dose) at which booster vaccinations are administered
`booster_coverage`	a matrix of coverages (timesteps x boosters) specifying the proportion the previously vaccinated population to continue receiving booster doses. The rows of the matrix must be the same size as `timesteps`. The columns of the matrix must be the same size as `booster_spacing`.
`booster_profile`	list of lists representing each booster profile, the outer list must be the same length as `booster_spacing`. Create vaccine profiles with `create_pev_profile`

set_mda *Parameterise a Mass Drug Administration*

Description

Parameterise a Mass Drug Administration

Usage

```
set_mda(parameters, drug, timesteps, coverages, min_ages, max_ages)
```

Arguments

`parameters`	a list of parameters to modify
`drug`	the index of the drug to administer
`timesteps`	a vector of timesteps for each round of mda
`coverages`	a vector of the proportion of the target population who receive each round
`min_ages`	a vector of minimum ages of the target population for each round exclusive (in timesteps)
`max_ages`	a vector of maximum ages of the target population for each round exclusive (in timesteps)

set_parameter_draw *Use parameter draw from the join posterior*

Description

Overrides default (median) model parameters with a single draw from the fitted joint posterior. Must be called prior to set_equilibrium.

Usage

```
set_parameter_draw(parameters, draw)
```

Arguments

`parameters`	the model parameters
`draw`	the draw to use. Must be an integer between 1 and 1000

`set_pev_epi` *Parameterise a pre-erythrocytic vaccine with an EPI strategy*

Description

distribute vaccine when an individual becomes a certain age. Efficacy will take effect after the last dose

Usage

```
set_pev_epi(
  parameters,
  profile,
  coverages,
  timesteps,
  age,
  min_wait,
  booster_spacing,
  booster_coverage,
  booster_profile,
  seasonal_boosters = FALSE
)
```

Arguments

`parameters` a list of parameters to modify

`profile` a list of details for the vaccine profile, create with `create_pev_profile`

`coverages` a vector of coverages for the primary doses

`timesteps` a vector of timesteps for each change in coverage

`age` the age when an individual will receive the first dose of the vaccine (in timesteps)

`min_wait` the minimum acceptable time since the last vaccination (in timesteps); When seasonal_boosters = TRUE, this represents the minimum time between an individual receiving the final dose and the first booster. When using both set_mass_pev and set_pev_epi, this represents the minimum time between an individual being vaccinated under one scheme and vaccinated under another.

`booster_spacing` the timesteps (following the final primary dose) at which booster vaccinations are administered

`booster_coverage` a matrix of coverages (timesteps x boosters) specifying the proportion the previously vaccinated population to continue receiving booster doses. The rows of the matrix must be the same size as `timesteps`. The columns of the matrix must be the same size as `booster_spacing`.

`booster_profile` list of lists representing each booster profile, the outer list must be the same length as `booster_spacing`. Create vaccine profiles with `create_pev_profile`

`seasonal_boosters`
: logical, if TRUE the first booster timestep is relative to the start of the year, otherwise they are relative to the last primary dose

`set_pmc` *Parameterise a perennial malaria chemoprevention (PMC, formerly IPIi)*

Description

Parameterise a perennial malaria chemoprevention (PMC, formerly IPIi)

Usage

```
set_pmc(parameters, drug, timesteps, coverages, ages)
```

Arguments

`parameters` a list of parameters to modify

`drug` the index of the drug to administer

`timesteps` a vector of timesteps for each change in coverage

`coverages` a vector of proportions of the target population to receive the intervention

`ages` a vector of ages at which PMC is administered (in timesteps)

`set_smc` *Parameterise a Seasonal Malaria Chemoprevention*

Description

Parameterise a Seasonal Malaria Chemoprevention

Usage

```
set_smc(parameters, drug, timesteps, coverages, min_ages, max_ages)
```

Arguments

`parameters` a list of parameters to modify

`drug` the index of the drug to administer

`timesteps` a vector of timesteps for each round of smc

`coverages` a vector of the proportion of the target population who receive each round

`min_ages` a vector of minimum ages of the target population for each round exclusive (in timesteps)

`max_ages` a vector of maximum ages of the target population for each round exclusive (in timesteps) drug

`set_species` *Parameterise the mosquito species to use in the model*

Description

Parameterise the mosquito species to use in the model

Usage

```
set_species(parameters, species, proportions)
```

Arguments

`parameters`	the model parameters
`species`	a list of species presets
`proportions`	a vector of proportions for each species

`set_spraying` *Parameterise an indoor spraying strategy*

Description

The model will apply indoor spraying at `timesteps` to a random sample of the entire human population. The sample size will be a proportion of the human population taken from the corresponding `coverages`. The sample *can* contain humans who have already benefited from spraying.

If a human in the sample lives in a sprayed house, the efficacy of the spraying will be returned to the maximum.

The structure for the indoor residual spraying model is documented in the S.I. of 10.1038/s41467-018-07357-w

Usage

```
set_spraying(
  parameters,
  timesteps,
  coverages,
  ls_theta,
  ls_gamma,
  ks_theta,
  ks_gamma,
  ms_theta,
  ms_gamma
)
```

Arguments

`parameters` a list of parameters to modify

`timesteps` the timesteps at which to spray

`coverages` the proportion of the population who get indoor spraying

`ls_theta` matrix of mortality parameters With nrows=length(timesteps), ncols=length(species)

`ls_gamma` matrix of mortality parameters per timestep With nrows=length(timesteps), ncols=length(species)

`ks_theta` matrix of feeding success parameters per timestep With nrows=length(timesteps), ncols=length(species)

`ks_gamma` matrix of feeding success parameters per timestep With nrows=length(timesteps), ncols=length(species)

`ms_theta` matrix of deterrence parameters per timestep With nrows=length(timesteps), ncols=length(species)

`ms_gamma` matrix of deterrence parameters per timestep With nrows=length(timesteps), ncols=length(species)

`set_tbv` *Parameterise an TBV strategy*

Description

Parameterise an TBV strategy

Usage

```
set_tbv(parameters, timesteps, coverages, ages)
```

Arguments

`parameters` a list of parameters to modify

`timesteps` a vector of timesteps for each round of vaccinations

`coverages` the coverage for each round of vaccinations

`ages` a vector of ages of the target population (in years)

SP_AQ_params *Preset parameters for the SP-AQ drug*

Description

Preset parameters for the SP-AQ drug

Usage

```
SP_AQ_params
```

Format

An object of class `numeric` of length 4.

Details

Use a vector of preset parameters for the SP-AQ drug (sulphadoxine-pyrimethamine and amodiaquine)

Default parameters, from L to R, are: drug_efficacy: 0.9, drug_rel_c: 0.32, drug_prophylaxis_shape: 4.3, drug_prophylaxis_scale: 38.1

steph_params *Preset parameters for the An. stephensi vector*

Description

Preset parameters for the An. stephensi vector

Usage

```
steph_params
```

Format

An object of class `list` of length 7.

Details

Default parameters: species: "steph" blood_meal_rates: 0.3333333 foraging_time: .69 Q0: 0.21 phi_bednets: 0.57 phi_indoors: 0.37 mum: 0.112

parameters reference: https://bmcmedicine.biomedcentral.com/articles/10.1186/s12916-022-02324-1 Values for phi are from: https://github.com/arranhamlet/stephensi_ETH_publication/blob/297352e244f8ed658e8bc3f32be4
L21 values for Q0: are the average from: ttps://github.com/cwhittaker1000/stephenseasonality/blob/main/data/bionomic_spe

Index