Package 'Rschistox'

March 8, 2023

Type Package

Title Simulation of schistosomiasis transmission in the population, with treatment strategies

Version 0.1.0

Author Who wrote it

Maintainer The package maintainer <yourself@somewhere.net>

Description An R package that simulates schistosomiasis transmission in the population. It is adapted from the schistoxpkg in Julia.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports tidyverse

RoxygenNote 7.2.3

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

`administer_drug` *Administer drugs function to administer drug to a specific variable (e.g. female_worms or eggs). input the variable, the indices to apply to and the effectiveness of treatment administer_drug(humans, indices, drug_effectiveness) administer mda drugs to chosen individuals in the population. If they adhere to the drugs, then they reduce male and female worms with a given efficacy alongside removing eggs*

Description

Administer drugs function to administer drug to a specific variable (e.g. female_worms or eggs). input the variable, the indices to apply to and the effectiveness of treatment administer_drug(humans, indices, drug_effectiveness) administer mda drugs to chosen individuals in the population. If they adhere to the drugs, then they reduce male and female worms with a given efficacy alongside removing eggs

Usage

```
administer_drug(humans, indices, drug_effectiveness)
```

`administer_vaccine` *Administer vaccine ——— function to administer drug to a specific variable (e.g. female_worms or eggs). input the variable, the indices to apply to and the effectiveness of treatment administer_vaccine(humans, indices, vaccine_effectiveness, vaccine_duration) administer vaccine to chosen individuals in the population. reduce male and female worms with a given efficacy alongside removing eggs and adding to their vaccine status signifying that they will have increased immunity for a chosen period of time*

Description

Administer vaccine ——— function to administer drug to a specific variable (e.g. female_worms or eggs). input the variable, the indices to apply to and the effectiveness of treatment administer_vaccine(humans, indices, vaccine_effectiveness, vaccine_duration) administer vaccine to chosen individuals in the population. reduce male and female worms with a given efficacy alongside removing eggs and adding to their vaccine status signifying that they will have increased immunity for a chosen period of time

Usage

```
administer_vaccine(humans, indices, vaccine_effectiveness, vaccine_duration)
```

`birth_of_human` *Birth of humans add an individual to the population*

Description

Birth of humans add an individual to the population

Usage

```
birth_of_human(humans, pars)
```

`calculate_worm_pairs` *Calculate worm pairs in humans ———————————— function calculate_worm_pairs(humans) return min(sum(humans.female_worms), sum(humans.male_worms)) end calculate how many pairs of worms there are in each human host*

Description

Calculate worm pairs in humans ———————————— function calculate_worm_pairs(humans) return min(sum(humans.female_worms), sum(humans.male_worms)) end calculate how many pairs of worms there are in each human host

Usage

```
calculate_worm_pairs(female_worms, male_worms)
```

`cercariae_death` *kill cercariae in the environment function to kill cercariae in the environment Kill a chosen proportion of cercariae in the environment governed by the cercariae_survival parameter in the pars struct. This parameter governs what proportion of cercariae survive for for one additional day, so if the time step is greater than one, we have to calculate the correct proportion who die over the chosen time step*

Description

kill cercariae in the environment function to kill cercariae in the environment Kill a chosen proportion of cercariae in the environment governed by the cercariae_survival parameter in the pars struct. This parameter governs what proportion of cercariae survive for for one additional day, so if the time step is greater than one, we have to calculate the correct proportion who die over the chosen time step

Usage

```
cercariae_death(cercariae, miracidia, pars)
```

`cercariae_uptake` *Cercariae uptake ———————————— cercariae_uptake(humans, cercariae, miracidia, pars) uptake cercariae into humans, whilst updating cercariae with miracidia. Uptaken cercariae immediately become worms in this formulation*

Description

Cercariae uptake ———————————— cercariae_uptake(humans, cercariae, miracidia, pars) uptake cercariae into humans, whilst updating cercariae with miracidia. Uptaken cercariae immediately become worms in this formulation

Usage

```
cercariae_uptake(humans, cercariae, miracidia, pars)
```

`cercariae_uptake_with_human_larvae` *cercariae uptake human larvae ———————————- uptake cercariae into humans, whilst updating cercariae with matured miracidia. Uptaken cercariae become larvae within humans, rather than immediately into worms with this function.*

Description

cercariae uptake human larvae ———————————- uptake cercariae into humans, whilst updating cercariae with matured miracidia. Uptaken cercariae become larvae within humans, rather than immediately into worms with this function.

Usage

```
cercariae_uptake_with_human_larvae(humans, cercariae, miracidia, pars)
```

`collect_prevs` *Store prevalence and sac prevalence ———————————- when we run multiple simulations, we store them in an array. This function will store the prevalence and sac prevalence collect multiple prevalences within the population and store in appropriate arrays*

Description

Store prevalence and sac prevalence ———————————- when we run multiple simulations, we store them in an array. This function will store the prevalence and sac prevalence collect multiple prevalences within the population and store in appropriate arrays

Usage

```
collect_prevs(
  times,
  prev,
  sac_prev,
  high_burden,
  high_burden_sac,
  adult_prev,
  high_adult_burden,
  record,
  run
)
```

`count_eggs` *count number of eggs ———————————————- count the total number of eggs in the human population*

Description

count number of eggs ———————————————- count the total number of eggs in the human population

Usage

```
count_eggs(humans)
```

`create_contact_settings` *Contact settings create the age specific contact settings given the scenario This will create age dependent contact rates based on the scenario for simulation which is input. This is either "low adult", "moderate adult" or "high adult"*

Description

Contact settings create the age specific contact settings given the scenario This will create age dependent contact rates based on the scenario for simulation which is input. This is either "low adult", "moderate adult" or "high adult"

Usage

```
create_contact_settings(scenario)
```

`create_mda` *create MDA ———————————————— function to create a set of mda's which will be performed regularly first_mda_time specifies when this will first occur in years, last_mda_time is the final mda in this block regularity is how often to perform the mda in years. specify the proportion of pre SAC, SAC and adults at each of these time points also specify genders for these differect age groups, along with the effectiveness of mda function to create a set of mda's which will be performed regularly*

Description

create MDA ———————————————— function to create a set of mda's which will be performed regularly first_mda_time specifies when this will first occur in years, last_mda_time is the final mda in this block regularity is how often to perform the mda in years. specify the proportion of pre SAC, SAC and adults at each of these time points also specify genders for these differect age groups, along with the effectiveness of mda function to create a set of mda's which will be performed regularly

Usage

```
create_mda(
  pre_SAC_prop,
  SAC_prop,
  adult_prop,
  first_mda_time,
  last_mda_time,
  regularity,
  pre_SAC_gender,
  SAC_gender,
  adult_gender,
  mda_effectiveness
)
```

Arguments

pre_SAC_prop	is the proportion of pre SAC given treatment at each of the time points
SAC_prop	is the proportion of SAC given treatment at each of the time points
adult_prop	is the proportion of adults given treatment at each of the time points
first_mda_time	specifies when mda will first be administered in years
last_mda_time	is the final mda in this block
regularity	is how often to perform the mda in years

`create_population` *Initial population ——— This will create the initial human population with randomly chosen age, and gender. Predisposition is taken to be gamma distributed There is also a male and female adjustment to predisposition adjusting for gender specific behaviour In addition to this, it will create the initial miracidia environment vector*

Description

Initial population ——— This will create the initial human population with randomly chosen age, and gender. Predisposition is taken to be gamma distributed There is also a male and female adjustment to predisposition adjusting for gender specific behaviour In addition to this, it will create the initial miracidia environment vector

Usage

```
create_population(pars)
```

create_population_specified_ages *Initial human population with an age distribution ———— This will create the initial human population with an age distribution specified by the spec_ages variable Predisposition is taken to be gamma distributed. There is also a male and female adjustment to predisposition adjusting for gender specific behaviour In addition to this, it will create the initial miracidia environment vector*

Description

Initial human population with an age distribution ———— This will create the initial human population with an age distribution specified by the spec_ages variable Predisposition is taken to be gamma distributed. There is also a male and female adjustment to predisposition adjusting for gender specific behaviour In addition to this, it will create the initial miracidia environment vector

Usage

```
create_population_specified_ages(pars)
```

death_of_human *Death of humans ————————————————*

Description

Death of humans ————————————————

Usage

```
death_of_human(humans)
```

egg_production *Number of eggs produced ————————————————- function to calculate the number of eggs produced this is done by choosing from a negative binomial distribution for each worms, where the mean and aggregation parameters are calculated as in the "Refined stratified-worm-burden models that incorporate specific biological features of human and snail hosts provide better estimates of Schistosoma diagnosis, transmission, and control" paper for julia the negative binomial describes the number of failures before the given number of successes in a collection of independent Bernoulli trials. we need to specify a probability of success, and a given number of successes, which are derived from the mean and aggregation in the function below inputs r - aggregation factor for NB distribution egg_production!(humans, pars) function to produce eggs for individuals, dependent on how many worms they have and the max fecundity and density dependent fecundity of the population*

Description

Number of eggs produced ———————————————————- function to calculate the number of eggs produced this is done by choosing from a negative binomial distribution for each worms, where the mean and aggregation parameters are calculated as in the "Refined stratified-worm-burden models that incorporate specific biological features of human and snail hosts provide better estimates of Schistosoma diagnosis, transmission, and control" paper for julia the negative binomial describes the number of failures before the given number of successes in a collection of independent Bernoulli trials. we need to specify a probability of success, and a given number of successes, which are derived from the mean and aggregation in the function below inputs r - aggregation factor for NB distribution egg_production!(humans, pars) function to produce eggs for individuals, dependent on how many worms they have and the max fecundity and density dependent fecundity of the population

Usage

```
egg_production(humans, pars)
```

`egg_production_increasing`

egg production increasing egg_production_increasing!(humans, pars) function to produce eggs for individuals, dependent on how many worms they have and the max fecundity and density dependent fecundity of the population

---

Description

egg production increasing egg_production_increasing!(humans, pars) function to produce eggs for individuals, dependent on how many worms they have and the max fecundity and density dependent fecundity of the population

Usage

```
egg_production_increasing(humans, pars)
```

`generate_ages_and_deaths`

age population and generating death ages ———————————— function to age population and generating death ages Step forward the population by a number of steps, where we will go through aging and removing individuals when they pass their age of death. This will generate an age distribution in the population which corresponds to the death_prob_by_age and ages_for_deaths parameters, which specify the probability of dying at each age.

Description

age population and generating death ages ———————————— function to age population and generating death ages Step forward the population by a number of steps, where we will go through aging and removing individuals when they pass their age of death. This will generate an age distribution in the population which corresponds to the death_prob_by_age and ages_for_deaths parameters, which specify the probability of dying at each age.

Usage

```
generate_ages_and_deaths(num_steps, humans, pars)
```

`generate_age_distribution` *generate a distribution for ages ———————————————- function to generate a distribution for ages based on a specified demography generate population numbers for each age in*

Description

generate a distribution for ages ———————————————- function to generate a distribution for ages based on a specified demography generate population numbers for each age in

Usage

```
generate_age_distribution(pars)
```

`get_death_age` *Age for death of an individual ———————————————— function to generate an age for death of an individual This will create the initial human population with randomly chosen age, and gender. Predisposition is taken to be gamma distributed There is also a male and female adjustment to predisposition adjusting for gender specific behaviour In addition to this, it will create the initial miracidia environment vector*

Description

Age for death of an individual ———————————————— function to generate an age for death of an individual This will create the initial human population with randomly chosen age, and gender. Predisposition is taken to be gamma distributed There is also a male and female adjustment to predisposition adjusting for gender specific behaviour In addition to this, it will create the initial miracidia environment vector

Usage

```
get_death_age(pars)
```

`get_prevalences` *get prevalences ———— calculate the desired prevalences in the human population, and store them in an out struct*

Description

get prevalences ———— calculate the desired prevalences in the human population, and store them in an out struct

Usage

```
get_prevalences(humans, time, pars)
```

`hello` *Hello, World!*

Description

Prints 'Hello, world!'.

Usage

```
hello()
```

Examples

```
hello()
```

`Human` *Human*

Description

This function contains the information about a human individual. This contains age, the pre determined age of death, community they are in, their gender, predisposition to picking up cercariae, the number of larvae, female and male worms and eggs in the individual along with a count of total lifetime eggs. Also it has their age dependent contact rate, adherence and access to interventions.

Usage

```
Human(
  age,
  death_age,
  gender,
  predisposition,
  female_worms,
  male_worms,
  eggs,
  vac_status,
  age_contact_rate,
  adherence,
  access,
  community,
  relative_contact_rate,
  uptake_rate,
  acquired_immunity,
  total_worms,
  larvae,
  last_uptake
)
```

`human_larvae_maturity` *Human larvae maturity ———— This will mature the human larvae into worms after a chosen number of days, which is specified by the human_larvae_maturity_time parameter in the pars struct*

Description

Human larvae maturity ———— This will mature the human larvae into worms after a chosen number of days, which is specified by the human_larvae_maturity_time parameter in the pars struct

Usage

```
human_larvae_maturity(humans, pars)
```

`kato_katz` *kato_katz eggs ————- calculate number of eggs using kato katz method. Gamma_k is a gamma distribution with shape and scale defined by pars.kato_katz_par*

Description

kato_katz eggs ————- calculate number of eggs using kato katz method. Gamma_k is a gamma distribution with shape and scale defined by pars.kato_katz_par

Usage

```
kato_katz(eggs, gamma_k)
```

`load_population_from_file` *load_population_from_file(filename) load the environmental variables saved in the specified file*

Description

load_population_from_file(filename) load the environmental variables saved in the specified file

Usage

```
load_population_from_file(filename)
```

`make_age_contact_rate_array` *Age dependent contact rate ———————————- function to get age dependent contact rate. the contact rates are taken from the "What is required in terms of mass drug administration to interrupt the transmission of schistosome parasites in regions of endemic infection?" paper at some point we may change this to be an input from a file instead*

Description

Age dependent contact rate ———————————- function to get age dependent contact rate. the contact rates are taken from the "What is required in terms of mass drug administration to interrupt the transmission of schistosome parasites in regions of endemic infection?" paper at some point we may change this to be an input from a file instead

Usage

```
make_age_contact_rate_array(pars, scenario, input_ages, input_contact_rates)
```

`mda` *Mass drug administration ——————————— function for mass drug administration currently there is no correlation between individuals chosen each time mda(humans, mda_coverage, min_age_mda, max_age_mda, mda_effectiveness, mda_gender) administer mda in the population. This includes choosing individuals between specified ages, having a certain level of coverage and taking access and adherence into consideration*

Description

Mass drug administration ——————————— function for mass drug administration currently there is no correlation between individuals chosen each time mda(humans, mda_coverage, min_age_mda, max_age_mda, mda_effectiveness, mda_gender) administer mda in the population. This includes choosing individuals between specified ages, having a certain level of coverage and taking access and adherence into consideration

Usage

```
mda(
  humans,
  mda_coverage,
  min_age_mda,
  max_age_mda,
  mda_effectiveness,
  mda_gender
)
```

`mda_information` *mda_information This function contains the information for the mda, storing the coverage, minimum and maximum age targeted, gender, drug efficacy and the time for the mda to be done*

Description

mda_information This function contains the information for the mda, storing the coverage, minimum and maximum age targeted, gender, drug efficacy and the time for the mda to be done

Usage

```
mda_information(
  mda_information,
  coverage,
  min_age,
  max_age,
  gender,
  effectiveness,
  time
)
```

`miracidia_death` *Kill miracidia in the environment ———————————— Kill a chosen proportion of miracidia in the environment governed by the miracidia_survival parameter in the pars struct*

Description

Kill miracidia in the environment ———————————— Kill a chosen proportion of miracidia in the environment governed by the miracidia_survival parameter in the pars struct

Usage

```
miracidia_death(miracidia, pars)
```

`miracidia_production` *Miracidia production ——————————————————- function miracidia_production!(humans) release eggs from individuals into the environment as miracidia. Release is relative to the contact rate with the environment for each individual.*

Description

Miracidia production ——————————————————- function miracidia_production!(humans) release eggs from individuals into the environment as miracidia. Release is relative to the contact rate with the environment for each individual.

Usage

```
miracidia_production(humans)
```

`out` *out This function contains the different outputs we are interested in recording. This is the overall population burden, with categories for low, moderate and heavy burdens, along with separate categories for the school age children and adults. Along with these, the time of each result is recorded, so we can subsequently see the prevalence of the outbreak over time.*

Description

out This function contains the different outputs we are interested in recording. This is the overall population burden, with categories for low, moderate and heavy burdens, along with separate categories for the school age children and adults. Along with these, the time of each result is recorded, so we can subsequently see the prevalence of the outbreak over time.

Usage

```
out(
  population_burden,
  sac_burden,
  adult_burden,
  pop_prev,
  sac_prev,
  adult_prev,
  sac_pop,
  adult_pop,
  final_ages,
  recorded_eggs,
  time
)
```

Parameters *Parameters*

Description

This function takes in all the parameters for the model

Usage

```
Parameters(
  N = 500,
  time_step = 20,
  N_communities = 1,
  community_probs = 1,
  community_contact_rate = 1,
  density_dependent_fecundity = 7e-04,
  average_worm_lifespan = 5.7,
  max_age = 100,
  initial_worms = 10,
  initial_miracidia = 1e+08,
  initial_miracidia_days = 1,
  init_env_cercariae = 1e+08,
  worm_stages = 1,
  contact_rate = 0.18,
  max_fec_contact_rate_product = 0.8,
  max_fecundity = 20,
  age_contact_rates = input_contact_rates/sum(input_contact_rates),
  ages_for_contacts = c(4, 9, 15, 100),
  contact_rate_by_age_array = rep(0, times = max_age + 1),
  mda_adherence = 1,
  mda_access = 1,
  female_factor = 1,
  male_factor = 1,
  miracidia_maturity = 24,
  birth_rate = 28 * time_step/(1000 * 365),
  human_cercariae_prop = 1,
  predis_aggregation = 0.24,
  cercariae_survival = 0.05,
  miracidia_survival = 0.05,
 death_prob_by_age = c(0.0656, 0.0093, 0.003, 0.0023, 0.0027, 0.0038, 0.0044, 0.0048,
   0.0053, 0.0065, 0.0088, 0.0106, 0.0144, 0.021, 0.0333, 0.0529, 0.0851, 0.1366,
    0.2183, 0.2998, 0.3698, 1),
 ages_for_death = c(1, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80,
    85, 90, 95, 100, 110),
 r = 0.03,
 vaccine_effectiveness = 0.86,
 drug_effectiveness = 0.86,
 spec_ages = c(7639, 7082, 6524, 5674, 4725, 4147, 3928, 3362, 2636, 1970, 1468, 1166,
    943, 718, 455, 244),
 ages_per_index = 5,
 record_frequency = 1/24,
```

```
  use_kato_katz = 0,
  kato_katz_par = 0.87,
  heavy_burden_threshold = 50,
  rate_acquired_immunity = 0,
  M0 = 20,
  human_larvae_maturity_time = 30,
  egg_sample_size = 1/100,
  input_ages = c(4, 9, 15, 100),
  input_contact_rates = c(0.01, 1.2, 1, 0.02),
  scenario = "high adult"
)
```

Arguments

N	human population size
time_step	length of time step (in days)
N_communities	number of communities in the population sharing the same environmental source
community_probs	probability of being in each community
community_contact_rate	contact rate with the environment for each of the commununity
density_dependent_fecundity	decrease in egg production per worm due to high density of worms
average_worm_lifespan	average expectancy of a worm
max_age	maximum age of individual
initial_worms	initial no. of worms
initial_miracidia	initial no. of miracidia in the environment
initial_miracidia_days	no.of days miracidia will age into cercariae larvae
init_env_cercariae	initial no of cercaria in the environment
worm_stages	number of stages in the worm. Having 1 stage will result to a Gamma distribution
contact_rate	global contact rate for the uptake of larvae from the environment
max_fec_contact_rate_product	product of max fecundity and the contact rate in the population. Setting this to a desired value is often a good way to ensure that the epidemic stays within a reasonable range, as when the max fecundity increases, if the contact rate doesn't decrease appropriately, then the behaviour of the outbreak can be unrealistically difficult to control.
max_fecundity	expected no. of eggs from a single worm
age_contact_rates	contact rate for the uptake of larvae from the environment for the chosen age groups
ages_for_contacts	age groups for specifying contact rates

`contact_rate_by_age_array`
: <- rep(0,times=max_age+1) array holding contact rate for each age

`mda_adherence`
: proportion of people who adhere to the MDA

`mda_access`
: proportion of people who have access to the MDA

`female_factor`
: factor for altering the contact rate for females, if we choose to have gender-specific behavior which affects contact rate

`male_factor`
: factor for altering the contact rate for males, if we choose to have gender-specific behavior which affects contact rate

`miracidia_maturity`
: no of days after which miracidias will mature to cercariae

`birth_rate`
: rate of birth of humans

`human_cercariae_prop`
: proportion of cercariae which are able to infect humans

`predis_aggregation`
: aggregation parameter for Poisson distributed egg production

`cercariae_survival`
: proportion of cercariae that survive from one time point to the next

`miracidia_survival`
: proportion of miracidia that survive from one time point to the next

`death_prob_by_age`
: probability of dying each year, specified by age

`ages_for_death`
: age ranges for death probailities

`r`
: aggregation parameter for negative binomially distributed egg production

`vaccine_effectiveness`
: efficacy of a vaccine if one is used

`drug_effectiveness`
: efficacy of a drug given during MDA

`spec_ages`
: number of individuals by age group

`ages_per_index`
: how many different ages we include in the spec_ages parameter

`record_frequency`
: how often we should record the prevalence in the population dusing simulation

`use_kato_katz`
: if 1, use Kato-Katz for egg count, if 0, do not use KK

`kato_katz_par`
: parameter for Gamma distribution if KK is used

`heavy_burden_threshold`
: number of eggs at which an individual is said to have a heavy infection

`rate_acquired_immunity`
: rate at which immunity will be acquired for individuals. This will be multiplied by the cumulative nymber of worms people have had during their life to decide the level of immunity acquired

`M0`
: if a particular formula of egg production is used, this parameter is required and is a proxy for mean worm burden

`human_larvae_maturity_time`
: length of time (in days) after which a cercariae uptake by a human will mature into a worm

`egg_sample_size`
: the proportion of eggs which are sampled from each individual every time we check their burden (between 0 and 1). 1= all eggs in the person are sampled. Typical value fpr a urine sample may be ~1/100

input_ages	input ages for constructing contact array
input_contact_rates	input contact rates
scenario	can be one of "low adult", "moderate adult" or high adult"

`run_repeated_sims_no_births_deaths`

repeat simulations ——— repeat simulations where we allow mdas and vaccination, but keep the population the same by adding a birth for every death run multiple simulations where aging of the population is not included and larvae are uptaken by humans as worms

---

Description

repeat simulations ——— repeat simulations where we allow mdas and vaccination, but keep the population the same by adding a birth for every death run multiple simulations where aging of the population is not included and larvae are uptaken by humans as worms

Usage

```
run_repeated_sims_no_births_deaths(
  filename,
  num_time_steps,
  mda_info,
  vaccine_info,
  num_repeats
)
```

`run_repeated_sims_no_births_deaths_human_larvae`

repeat simulations ——— run multiple simulations where aging of the population is not included and larvae are uptaken by humans as larvae

---

Description

repeat simulations ——— run multiple simulations where aging of the population is not included and larvae are uptaken by humans as larvae

Usage

```
run_repeated_sims_no_births_deaths_human_larvae(
  filename,
  num_time_steps,
  mda_info,
  vaccine_info,
  num_repeats
)
```

`run_repeated_sims_no_births_deaths_increasing`
repeat simulations ——— repeat simulations where we allow mdas and vaccination, but keep the population the same by adding a birth for every death run multiple simulations where aging of the population is not included and larvae are uptaken by humans as worms, and egg production is monotonically increasing

---

Description

repeat simulations ——— repeat simulations where we allow mdas and vaccination, but keep the population the same by adding a birth for every death run multiple simulations where aging of the population is not included and larvae are uptaken by humans as worms, and egg production is monotonically increasing

Usage

```
run_repeated_sims_no_births_deaths_increasing(
  filename,
  num_time_steps,
  mda_info,
  vaccine_info,
  num_repeats
)
```

`run_repeated_sims_no_population_change_human_larvae`
repeat simulations ——— repeat simulations where we allow mdas and vaccination, but keep the population the same by adding a birth for every death run multiple simulations where the population is aged, but each death is replaced by a birth and larvae are uptaken by humans as larvae

---

Description

repeat simulations ——— repeat simulations where we allow mdas and vaccination, but keep the population the same by adding a birth for every death run multiple simulations where the population is aged, but each death is replaced by a birth and larvae are uptaken by humans as larvae

Usage

```
run_repeated_sims_no_population_change_human_larvae(
  filename,
  num_time_steps,
  mda_info,
  vaccine_info,
  num_repeats
)
```

`run_repeated_sims_no_population_change_increasing` *repeated simulations ———————————————— run_repeated_sims_no_population_change(filename, num_time_steps, mda_info, vaccine_info, num_repeats) run multiple simulations where aging of the population is not included, larvae are uptaken by humans as worms*

Description

repeated simulations ————————————————- run_repeated_sims_no_population_change(filename, num_time_steps, mda_info, vaccine_info, num_repeats) run multiple simulations where aging of the population is not included, larvae are uptaken by humans as worms

Usage

```
run_repeated_sims_no_population_change_increasing(
  filename,
  num_time_steps,
  mda_info,
  vaccine_info,
  num_repeats
)
```

`save_population_to_file` *save population to file ————————————————- save the enironment variables in a specified file*

Description

save population to file ————————————————- save the enironment variables in a specified file

Usage

```
save_population_to_file(filename, humans, miracidia, cercariae, pars)
```

specified_age_distribution

construct the set of ages ——— function to construct the set of ages, with size N specified_age_distribution(pars) output ages according to a specified age distribution

Description

construct the set of ages ——— function to construct the set of ages, with size N specified_age_distribution(pars) output ages according to a specified age distribution

Usage

```
specified_age_distribution(pars)
```

update_contact_rate

update contact rates ———- function to update the contact rate of individuals in the population. This is necessary as over time when people age, they will move through different age groups which have different contact rates

Description

update contact rates ———- function to update the contact rate of individuals in the population. This is necessary as over time when people age, they will move through different age groups which have different contact rates

Usage

```
update_contact_rate(humans, pars)
```

update_env_constant_population

update env constant population ——— update_env_constant_population(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we include deaths and for each death an individual is immediately born. Interventions are included in this function and larvae are immediately uptaken as worms

Description

update env constant population ——— update_env_constant_population(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we include deaths and for each death an individual is immediately born. Interventions are included in this function and larvae are immediately uptaken as worms

Usage

```
update_env_constant_population(
  num_time_steps,
  humans,
  miracidia,
  cercariae,
  pars,
  mda_info,
  vaccine_info
)
```

---

update_env_constant_population_human_larvae

update env_sonstant population ———————————— update_env_constant_population_human_larvae(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we include deaths and for each death an individual is immediately born. Interventions are included in this function and larvae are uptaken as larvae in the humans.

Description

update env_sonstant population ———————————— update_env_constant_population_human_larvae(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we include deaths and for each death an individual is immediately born. Interventions are included in this function and larvae are uptaken as larvae in the humans.

Usage

```
update_env_constant_population_human_larvae(
  num_time_steps,
  humans,
  miracidia,
  cercariae,
  pars,
  mda_info,
  vaccine_info
)
```

---

update_env_constant_population_increasing

update env constant population increasing ————————————- update_env_constant_population_increasing(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we include deaths and for each death an individual is immediately born. Interventions are included in this function and larvae are uptaken immediately as worms and egg production follows a monotonically increasing function

---

Description

update env constant population increasing ———————————- update_env_constant_population_increasing(num_t humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we include deaths and for each death an individual is immediately born. Interventions are included in this function and larvae are uptaken immediately as worms and egg production follows a monotonically increasing function

Usage

```
update_env_constant_population_increasing(
  num_time_steps,
  humans,
  miracidia,
  cercariae,
  pars,
  mda_info,
  vaccine_info
)
```

`update_env_no_births_deaths`

update env births and deaths ———————————— update_env_no_births_deaths(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we do not include births or deaths and individuals do not age Interventions are included in this function and larvae are uptaken immediately as worms

---

Description

update env births and deaths ———————————— update_env_no_births_deaths(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we do not include births or deaths and individuals do not age Interventions are included in this function and larvae are uptaken immediately as worms

Usage

```
update_env_no_births_deaths(
  num_time_steps,
  humans,
  miracidia,
  cercariae,
  pars,
  mda_info,
  vaccine_info
)
```

update_env_no_births_deaths_human_larvae

update env_no_births_deaths_human larvae ———————— – update_env_no_births_deaths_human_larvae(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we do not include births or deaths and individuals do not age Interventions are included in this function and larvae are uptaken as larvae in humans

Description

update env_no_births_deaths_human larvae ———————— update_env_no_births_deaths_human_larvae(nu humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we do not include births or deaths and individuals do not age Interventions are included in this function and larvae are uptaken as larvae in humans

Usage

```
update_env_no_births_deaths_human_larvae(
  num_time_steps,
  humans,
  miracidia,
  cercariae,
  pars,
  mda_info,
  vaccine_info
)
```

update_env_no_births_deaths_increasing

update env no. births and deaths increasing ———————— ——— update_env_no_births_deaths_increasing(num_time_steps, humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we do not include births or deaths and individuals do not age Interventions are included in this function and larvae are uptaken as immediately as worms and egg production is monotonically increasing

Description

update env no. births and deaths increasing ———————— update_env_no_births_deaths_increasing(num_time humans, miracidia, cercariae, pars, mda_info, vaccine_info) update the population for a given length of time. Here we do not include births or deaths and individuals do not age Interventions are included in this function and larvae are uptaken as immediately as worms and egg production is monotonically increasing

Usage

```
update_env_no_births_deaths_increasing(
  num_time_steps,
  humans,
  miracidia,
  cercariae,
  pars,
  mda_info,
  vaccine_info
)
```

`update_env_to_equilibrium`

update environment to equilibrium ———————————— update_env_to_equilibrium(num_time_steps, humans, miracidia, cercariae, pars) update the population for a given length of time. Here we do not age the population or include birth, deaths or interventions.

Description

update environment to equilibrium ———————————— update_env_to_equilibrium(num_time_steps, humans, miracidia, cercariae, pars) update the population for a given length of time. Here we do not age the population or include birth, deaths or interventions.

Usage

```
update_env_to_equilibrium(num_time_steps, humans, miracidia, cercariae, pars)
```

`update_env_to_equilibrium_human_larvae`

update the pop for a given length of time update_env_to_equilibrium_human_larvae(num_time_steps, humans, miracidia, cercariae, pars) update the population for a given length of time. Here we do not age the population or include birth, deaths or interventions and for this function larvae are uptaken from the environment into a larvae category in the humans, rather than immediately becoming worms

---

Description

update the pop for a given length of time update_env_to_equilibrium_human_larvae(num_time_steps, humans, miracidia, cercariae, pars) update the population for a given length of time. Here we do not age the population or include birth, deaths or interventions and for this function larvae are uptaken from the environment into a larvae category in the humans, rather than immediately becoming worms

Usage

```
update_env_to_equilibrium_human_larvae(
  num_time_steps,
  humans,
  miracidia,
  cercariae,
  pars
)
```

---

update_env_to_equilibrium_increasing *update environment to equilibrium ——— update_env_to_equilibrium_increasing(num_time_steps, humans, miracidia, cercariae, pars) update the population for a given length of time. Here we do not age the population or include birth, deaths or interventions and for this function larvae are uptaken from the environment immediately to worms and eggs are produced using a monotonically increasing function*

Description

update environment to equilibrium ——— update_env_to_equilibrium_increasing(num_time_s humans, miracidia, cercariae, pars) update the population for a given length of time. Here we do not age the population or include birth, deaths or interventions and for this function larvae are uptaken from the environment immediately to worms and eggs are produced using a monotonically increasing function

Usage

```
update_env_to_equilibrium_increasing(
  num_time_steps,
  humans,
  miracidia,
  cercariae,
  pars
)
```

---

update_mda *Update MDA ——— function to update the mda information update when the next mda will take place*

Description

Update MDA ——— function to update the mda information update when the next mda will take place

Usage

```
update_mda(mda_info, mda_round)
```

vaccine_information	*vaccine_information This function contains the information for the vaccine, storing the coverage, minimum and maximum age targeted, gender, drug efficacy and the time for the vaccine to be done along with how long the vaccine provides protection for*

Description

vaccine_information This function contains the information for the vaccine, storing the coverage, minimum and maximum age targeted, gender, drug efficacy and the time for the vaccine to be done along with how long the vaccine provides protection for

Usage

```
vaccine_information(coverage, min_age, max_age, gender, duration, time)
```

vac_decay	*add vaccination to population ——————————- function to add vaccination to population update vaccine information ——————————- function to update vaccine information =# vaccine decay ——————————— decrease vaccination status for each person by 1 each day*

Description

add vaccination to population ——————————- function to add vaccination to population update vaccine information ——————————- function to update vaccine information =# vaccine decay ——————————— decrease vaccination status for each person by 1 each day

Usage

```
vac_decay(humans, pars)
```

worm_maturity	*Kill worms within human host ——————————— Worms die have a specified mean life span, and hence a rate of deaths per day . the number of deaths, or maturing from each stage is dependent on the number of worm stages and rate of aging through stages and dying. This p is multiplied by the time scale of the simulation, so if 2 days pass between consecutive time points, twice as many worms age and die for human in humans println(human.eggs) end function to kill worms within human hosts, and if there is more than one stage for worm life, to update how many worms are in each stage*

Description

Kill worms within human host ———————————————— Worms die have a specified mean life span, and hence a rate of deaths per day . the number of deaths, or maturing from each stage is dependent on the number of worm stages and rate of aging through stages and dying. This p is multiplied by the time scale of the simulation, so if 2 days pass between consecutive time points, twice as many worms age and die for human in humans println(human.eggs) end function to kill worms within human hosts, and if there is more than one stage for worm life, to update how many worms are in each stage

Usage

```
worm_maturity(humans, pars)
```

Index