epiworld

0.0-1

Chapter 1

Example: 00-hello-world

Output from the program:

```
Running the model...
_____
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| done.
[epiworld-debug] DEBUGGING ON (compiled with EPI_DEBUG defined)
_____
SIMULATION STUDY
Population size     : 10000
Number of entitites : 0
Days (duration)     : 100 (of 100)
Number of variants  : 1
Last run elapsed t  : 40.00ms
Rewiring            : off
Virus(es):
 - covid 19 (baseline prevalence: 50 seeds)
Tool(s):
 - vaccine (baseline prevalence: 50.00%)
 - Immunity (covid 19) (originated in the model...)
Model parameters:
 (none)
Distribution of the population at time 100:
 - (0) Susceptible :  9950 -> 70
 - (1) Exposed     :    50 -> 70
 - (2) Recovered   :     0 -> 9271
 - (3) Removed     :     0 -> 589
Transition Probabilities:
 - Susceptible  0.95  0.05  0.00  0.00
 - Exposed      0.00  0.85  0.14  0.01
 - Recovered    0.00  0.00  1.00  0.00
 - Removed      0.00  0.00  0.00  1.00
```

Chapter 2

Benchmarking

Here we keep a list of scenarios where we compare epiworld with other ABM simulation engines. Although the comparison is made at the speed level, we also list features of capabilities and main differences between the engines.

Chapter 3

Contributor Code of Conduct

As contributors and maintainers of this project, we pledge to respect all people who contribute through reporting issues, posting feature requests, updating documentation, submitting pull requests or patches, and other activities.

We are committed to making participation in this project a harassment-free experience for everyone, regardless of level of experience, gender, gender identity and expression, sexual orientation, disability, personal appearance, body size, race, ethnicity, age, or religion.

Examples of unacceptable behavior by participants include the use of sexual language or imagery, derogatory comments or personal attacks, trolling, public or private harassment, insults, or other unprofessional conduct.

Project maintainers have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct. Project maintainers who do not follow the Code of Conduct may be removed from the project team.

Instances of abusive, harassing, or otherwise unacceptable behavior may be reported by opening an issue or contacting one or more of the project maintainers.

This Code of Conduct is adapted from the Contributor Covenant (http://contributor-covenant.org), version 1.0.0, available at http://contributor-covenant.org/version/1/0/0/

Chapter 4

epiworld c++ template library

4.1 Main features

This C++ template-header-only library provides a general framework for epidemiologic simulation. The main features of the library are:

1. Four key classes: `Model`, `Person`, `Tool`, and `Virus`.

2. The model features a social networks of `Persons`.

3. `Persons` can have multiple `Tools` as a defense system.

4. `Tools` can reduce contagion rate, transmissibility, death rates, and improve recovery rates.

5. `Viruses` can mutate (generating new variants).

6. `Models` can feature multiple states, e.g., `HEALTHY`, `SUSCEPTIBLE`, etc.

7. `Models` can have an arbitrary number of parameters.

8. **REALLY FAST** About 6.5 Million person/day simulations per second.

4.2 Algorithm

Setup

- Create viruses.

- Create tools (arbitrary).

- Set model parameters (arbitrary).

- Create global events (e.g., surveillance).

- Set up the population: small world network (default).

- Set up rewiring (optional).

- Set statuses (arbitrary number of them).

Run

1. Distribute the tool(s) and virus(es)

2. For each t in 1 -> Duration:

   - Update status for susceptible/infected/removed(?)
   - Mutate virus(es) (each individual)
   - Run global actions (e.g., surveillance)
   - Run rewiring algorithm

Along update:

- Contagion events are applied recorded.

- New variants are recorded.

- Optional user data is recorded.

4.3   Hello world (C++)

```
#include "include/epiworld/epiworld.hpp"
int main()
{
  // Creating a virus
  epiworld::Virus<> covid19("covid 19");
  covid19.set_infectiousness(.8);

  // Creating a tool
  epiworld::Tool<> vax("vaccine");
  vax.set_contagion_reduction(.95);
  // Creating a model
  epiworld::Model<> model;
  // Adding the tool and virus
  model.add_virus(covid19, .01);
  model.add_tool(vax, .5);
  // Generating a random pop
  model.population_from_adjlist(
    epiworld::rgraph_smallworld(1000, 5, .2)
  );
  // Initializing setting days and seed
  model.init(60, 123123);
  // Running the model
  model.run();
  model.print();
  return;
}
```

4.4   Surveillance simulation

- Incubation time of the disease $\sim$ `Gamma(3, 1)`

- Duration of the disease $\sim$`Gamma(12, 1)`

- Probability of becoming symptomatic: 0.9

- Prob. of transmission: 1.0.

- Vaccinated population: 25%

- Vaccine efficacy: .9.

- Vaccine reduction on transmission: 0.5.

- Surveillance program of x% of the population at random.

- Individuals who test positive become isolated.

4.4.1 Preliminary results

```
# With low surveillance
pop_size <- 20e3
pop_seed <- pop_size * .01
s_levels <- c(0.0001, 0.002)
system(sprintf("./07-surveillance.o %i %i 100 %.04f 2>&1", pop_seed, pop_size, s_levels[1]), intern = TRUE)
      |>
  cat(sep = "\n")
```

```
## Running the model...
## _____
## |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| done.
##
## _____
## SIMULATION STUDY
##
## Population size    : 20000
## Days (duration)    : 200 (of 200)
## Number of variants : 1
## Last run elapsed t : 505.00ms
## Rewiring           : off
##
## Virus(es):
##  - Covid19 (baseline prevalence: 100 seeds)
## Tool(s):
##  - Vaccine (baseline prevalence: 25.00%)
##
## Model parameters:
##  - Infect period        : 12.0000
##  - Latent period        : 3.0000
##  - Prob of symptoms      : 0.7000
##  - Prob of transmission  : 1.0000
##  - Prob. death           : 0.0010
##  - Prob. reinfect        : 0.1000
##  - Surveilance prob.     : 1.0e-04
##  - Vax efficacy          : 0.9000
##  - Vax redux transmision : 0.5000
##
## Distribution of the population at time 200:
##  - Total susceptible (S)          :    19900 -> 2106
##  - Total recovered (S)            :        0 -> 17369
##  - Total latent (I)               :      100 -> 109
##  - Total symptomatic (I)          :        0 -> 155
##  - Total symptomatic isolated (I) :        0 -> 2
##  - Total asymptomatic (I)         :        0 -> 72
##  - Total asymptomatic isolated (I) :       0 -> 0
##  - Total removed (R)              :        0 -> 187
##
## (S): Susceptible, (I): Infected, (R): Recovered
## _____
```

```
hist1 <- read.csv("07-surveillance_hist.txt", sep = " ")
surv1 <- read.csv("07-surveillance_user_data.txt", sep = " ")
# With high surveillance
system(sprintf("./07-surveillance.o %i %i 100 %.04f 2>&1", pop_seed, pop_size, s_levels[2]), intern = TRUE)
      |>
  cat(sep = "\n")
```

```
## Running the model...
## _____
## |||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| done.
##
## _____
## SIMULATION STUDY
##
## Population size    : 20000
## Days (duration)    : 200 (of 200)
## Number of variants : 1
## Last run elapsed t : 530.00ms
## Rewiring           : off
##
## Virus(es):
```

```
##  - Covid19 (baseline prevalence: 100 seeds)
## Tool(s):
##  - Vaccine (baseline prevalence: 25.00%)
##
## Model parameters:
##  - Infect period         : 12.0000
##  - Latent period         : 3.0000
##  - Prob of symptoms       : 0.7000
##  - Prob of transmission  : 1.0000
##  - Prob. death           : 0.0010
##  - Prob. reinfect        : 0.1000
##  - Surveilance prob.     : 0.0020
##  - Vax efficacy          : 0.9000
##  - Vax redux transmision : 0.5000
##
## Distribution of the population at time 200:
##  - Total susceptible (S)      :   19900 -> 2125
##  - Total recovered (S)        :       0 -> 17325
##  - Total latent (I)           :     100 -> 109
##  - Total symptomatic (I)      :       0 -> 155
##  - Total symptomatic isolated (I)  :    0 -> 8
##  - Total asymptomatic (I)     :       0 -> 76
##  - Total asymptomatic isolated (I) :    0 -> 1
##  - Total removed (R)          :       0 -> 201
##
## (S): Susceptible, (I): Infected, (R): Recovered
## _____
```

```
hist2 <- read.csv("07-surveillance_hist.txt", sep = " ")
surv2 <- read.csv("07-surveillance_user_data.txt", sep = " ")
hist_comb <- rbind(
  cbind(sim = as.character(s_levels[1]), hist1),
  cbind(sim = as.character(s_levels[2]), hist2)
  )
ggplot(hist_comb, aes(x = date, y = counts + 1, colour = status, linetype=sim)) +
  geom_line() +
  # scale_y_log10() +
  labs(y = "Counts (log)")
```

4.4.2 Cases detected

```
survdat <- rbind(
  with(surv1, rbind(
    data.frame(Id = as.character(s_levels[1]), Date = date, Type = "N Sampled",  n = nsampled),
    data.frame(Id = as.character(s_levels[1]), Date = date, Type = "N detected", n = ndetected),
    data.frame(Id = as.character(s_levels[1]), Date = date, Type = "N detected Asymp",  n =
      ndetected_asympt),
    data.frame(Id = as.character(s_levels[1]), Date = date, Type = "N Asymp",  n = nasymptomatic)
  )),
  with(surv2, rbind(
    data.frame(Id = as.character(s_levels[2]), Date = date, Type = "N Sampled",  n = nsampled),
    data.frame(Id = as.character(s_levels[2]), Date = date, Type = "N detected",  n = ndetected),
    data.frame(Id = as.character(s_levels[2]), Date = date, Type = "N detected Asymp",  n =
      ndetected_asympt),
    data.frame(Id = as.character(s_levels[2]), Date = date, Type = "N Asymp",  n = nasymptomatic)
  ))
)
ggplot(survdat, aes(x = Date, y = n + 1, colour = Type)) +
  geom_line() +
  facet_wrap(~Id) +
  scale_y_log10() +
  labs(y = "Counts (log)")
```

Chapter 5

MIT License

Copyright (c) 2021 George G. Vega Yon

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Chapter 6

model1

The dynamics of the simulation process are:

1. Discrete Markov process.

2. The simulation has the following parameters:

   a. New variant emergence at rate `X`. b. For each variant `k`:

   - Unvaccinated individuals become sick rate `C(k)`,
   - Mortality rate `D(k)`,
   - Recovery rate `H(k)`,
   - Vaccines have an efficaccy rate `E(v,k)` and pseudo vaccines (recovered) have efficacy rate `E(r,k)` < `E(v,k)`. In general, the probability of `i` acquiring the disease `k` from `j` will be equal to

   ``` P(i gets the disease from j | their states) = C(k) * (1 - E(i,k)) * (1 - E(j, k)) ```

   where `(i,j) in (u,v,r)`. Efficacy rate for unvaccinated is zero.

   - Vaccinated individuals have a reduced mortality rate `D(k,v)` > `D(k)`, and recovered individuals `D(k,r) in (D(k,v), D(k)]`
   - Vaccinated individuals have an increased recovery rate `H(k,v)` > `H(k)`, whereas recovered's rate `H(k,r) in [H(k), H(k,v))`.

   The sum of mortality and recovery rates is less than one since the difference represents no change.

   c. Each country vaccinates citizens at rate `V` function of `A` (availability) and `B` (citizens' acceptance rate.) d. In each country `i`, the entire population `N(i)` distributes between the following states:

   - Healthy unvaccinated (`N(i,t,u)`),
   - Healthy vaccinated (`N(i,t,v)`),
   - Deceased (`N(i,t,d)`),
   - Recovered (`N(i,t,r)`),
   - Unvaccinated and sick with variant (`N(i,t,s,k|u)`) `k`., and
   - Vaccinated and sick with variant (`N(i,t,s,k|v)`) `k`.

   Total sick are `N(i,t,k,s) = sum(g in {u,v}) N(i,t,k,s|g)`

   Globally, we keep track of the prevalence of new variants. Variants can disappear if no more individuals port the variant, i.e., the prevalence rate `P(k,t)=sum(i) N(i,s,k)` equals zero.

   d. Vaccines are manufactured at each country at rates `M(i)` and uniformly shared with other countries at rate `S(i)`. c. Population flows between each country pair `(i,j)` at a rate `F(i,j)`. Flows between countries do not change Population and are symmetric.

3. The simulation process is as follows:

  (a) Countries are initialized with a total population `N(i)`.

  (b) Variant zero initializes at a random location `i`, with an initial prevalence `P(k,t) = N(i,t,k)`.

  (c) For time `t` in `(0,T)` do:

  a. Unvaccinated individuals can become sick of variant `k` with probability:

  ``` Pr(h->s|i,t,k,u) ~ sum(g in {u,v}) (N(i,t - 1,s,k|g) + sum(j != i) F(i,j) * N(j,t-1,s,k|g)) * C(k) / (N(i) + sum(j != i) N(j)) ```

  b. Vaccinated individuals can become sick of variant `k` with probability: `Pr(v->s|i,t,k,v) ~ Pr(h->s|i,t,k) * (1 - E(v,k))`.

  b. Recovered individuals can become sick of variant `k` with probability: `Pr(v->s|i,t,k,r) ~ Pr(h->s|i,t,k) * (1 - E(r,k))`.

  c. Sick individuals with variant `k` die with probability `D(k)` or recover with probability `H(k)`, otherwise they stay infected; with the rates depending on their vaccination status `v` or `n`.

  d. Unvaccinated individuals vaccinate in country `i` with probability `P(u->v) ~ V(A(i,t), B(i))`.

  e. The country vaccine supply changes.

Chapter 7

EPI Simulator

7.1 Disease dynamics

Diseases continuously evolve in time. Changes in their genetic sequence make them more or less resistant to the particular version of the vaccine. Mutations also affect the transmissibility level and mortality rate of the disease. Using this approach allows making vaccination efficacy a function of compatibility between the variant and the vaccine.

When an individual becomes infected, the disease accumulates mutations in the new host. Ultimately, there is no single version of the disease present in the model, but rather an infinite number of them, each slightly different from the other.

7.2 Network dynamics

We can assume that the Population is organized in fully connected blocks for the first version of the model. Block sizes and the number of connections between blocks are Poisson random variables. Individuals interact with all the members of their blocks, and bridging individuals allow the disease to move across blocks.

7.3 Contagion dynamics

The transmission of the disease will be governed by the number of vaccinated, infected, and recovered within each block. Transmission between blocks will be treated in the same way, although individuals bridging the block will only interact with others within the block and their direct connections across the blocks.

7.4 Time dynamics

Time dynamics has two components, how biology evolves and how agents react.

The model develops as a continuous-time Markov process. Each block of individuals takes action at rates `L(i|N(i))` function of the local number of infections. This way, if

7.5 Updating agent's status

Like most other components, updating agents' states can be personalized. A naive approach allows agents to get infected with a single virus or stay as-is. The probability of this event is conditional on acquiring at most one virus. Since these are independent events, the conditional probability is computed as follows:

```
P(Variant k|at most 1) = P(at most 1| Variant k) * P(Variant k)/P(at most 1)
                       = P(only variant k)/P(variant k) * P(Variant k)/P(at most 1)
                       = P(only variant k)/P(at most 1)
```

Where

```
P(only variant k) = P(k) * Prod(m!=v) (1 - P(m))
P(at most 1) = P(None) + Sum(v in variants) P(v) * Prod(m != v) (1 - P(m))
P(None) = Prod(v in variants) (1 - P(v))
```

Furthermore, the (Variant, Person) pairs are treated independently.

7.5.1 Other parameters

- Who did you get the infection from.

- Omicron is 1.5 more infectious than delta.

- Surveillance:

  - Pull people to be tested at random.
  - Or at symptoms.
  - A mix of the two.

- Define a class for passing extra functions and datasets, for example, testing surveillance.

- Exposed people become infectious after k days.

- Network changesthe can be a function of an ERGM. Apply K steps throughout time.

- Add progress bar.

Chapter 8

Namespace Index

8.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Chapter 9

Hierarchical Index

9.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 10

Class Index

10.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 11

File Index

11.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 12

Namespace Documentation

12.1 epiworld::sampler Namespace Reference

Functions for sampling viruses.

Functions

- template<typename TSeq >
  std::function< void(Agent< TSeq > *, Model< TSeq > *)> make_update_susceptible (std::vector< epiworld_fast_uint > exclude={})

  Make a function to sample from neighbors.

- template<typename TSeq = int>
  std::function< Virus< TSeq > *(Agent< TSeq > *, Model< TSeq > *)> make_sample_virus_neighbors (std::vector< epiworld_fast_uint > exclude={})

  Make a function to sample from neighbors.

- template<typename TSeq = int>
  Virus< TSeq > * sample_virus_single (Agent< TSeq > *p, Model< TSeq > *m)

  Sample from neighbors pool of viruses (at most one)

12.1.1 Detailed Description

Functions for sampling viruses.

12.1.2 Function Documentation

12.1.2.1 make_sample_virus_neighbors()

```
template<typename TSeq = int>
std::function<Virus<TSeq>*(Agent<TSeq>*,Model<TSeq>*)> epiworld::sampler::make_sample_↩
virus_neighbors (
            std::vector< epiworld_fast_uint > exclude = {} )  [inline]
```

Make a function to sample from neighbors.

This is akin to the function default_update_susceptible, with the difference that it will create a function that supports excluding states from the sampling frame. For example, individuals who have acquired a virus can be excluded if in incubation state.

Template Parameters

TSeq	

Parameters

exclude	unsigned vector of states that need to be excluded from the sampling

Returns

> Virus<TSeq>∗ of the selected virus. If none selected (or none available,) returns a nullptr;

12.1.2.2 make_update_susceptible()

```
template<typename TSeq >
std::function<void(Agent<TSeq>*,Model<TSeq>*)> epiworld::sampler::make_update_susceptible (
            std::vector< epiworld_fast_uint > exclude = {} )  [inline]
```

Make a function to sample from neighbors.

This is akin to the function default_update_susceptible, with the difference that it will create a function that supports excluding states from the sampling frame. For example, individuals who have acquired a virus can be excluded if in incubation state.

Template Parameters

TSeq	

Parameters

exclude	unsigned vector of states that need to be excluded from the sampling

Returns

> Virus<TSeq>∗ of the selected virus. If none selected (or none available,) returns a nullptr;

12.1.2.3 sample_virus_single()

```
template<typename TSeq = int>
Virus<TSeq>* epiworld::sampler::sample_virus_single (
            Agent< TSeq > * p,
            Model< TSeq > * m )  [inline]
```

Sample from neighbors pool of viruses (at most one)

This function samples at most one virus from the pool of viruses from its neighbors. If no virus is selected, the function returns a `nullptr`, otherwise it returns a pointer to the selected virus.

This can be used to build a new update function (EPI_NEW_UPDATEFUN.)

Template Parameters

TSeq	

Parameters

p	Pointer to person
m	Pointer to the model

Returns

> Virus<TSeq>∗ of the selected virus. If none selected (or none available,) returns a nullptr;

12.2 sampler Namespace Reference

Functions for sampling viruses.

Functions

- template<typename TSeq >
  std::function< void(Agent< TSeq > ∗, Model< TSeq > ∗)> make_update_susceptible (std::vector< epiworld_fast_uint > exclude={})

  Make a function to sample from neighbors.

- template<typename TSeq = int>
  std::function< Virus< TSeq > ∗(Agent< TSeq > ∗, Model< TSeq > ∗)> make_sample_virus_neighbors (std::vector< epiworld_fast_uint > exclude={})

  Make a function to sample from neighbors.

- template<typename TSeq = int>
  Virus< TSeq > ∗ sample_virus_single (Agent< TSeq > ∗p, Model< TSeq > ∗m)

  Sample from neighbors pool of viruses (at most one)

12.2.1 Detailed Description

Functions for sampling viruses.

12.2.2 Function Documentation

12.2.2.1 make_sample_virus_neighbors()

```
template<typename TSeq = int>
std::function<Virus<TSeq>*(Agent<TSeq>*,Model<TSeq>*)> sampler::make_sample_virus_neighbors
(
            std::vector< epiworld_fast_uint > exclude = {} )  [inline]
```

Make a function to sample from neighbors.

This is akin to the function default_update_susceptible, with the difference that it will create a function that supports excluding states from the sampling frame. For example, individuals who have acquired a virus can be excluded if in incubation state.

Template Parameters

TSeq	

Parameters

exclude	unsigned vector of states that need to be excluded from the sampling

Returns

Virus<TSeq>∗ of the selected virus. If none selected (or none available,) returns a nullptr;

12.2.2.2 make_update_susceptible()

```
template<typename TSeq >
std::function<void(Agent<TSeq>*,Model<TSeq>*)> sampler::make_update_susceptible (
            std::vector< epiworld_fast_uint > exclude = {} )  [inline]
```

Make a function to sample from neighbors.

This is akin to the function default_update_susceptible, with the difference that it will create a function that supports excluding states from the sampling frame. For example, individuals who have acquired a virus can be excluded if in incubation state.

Template Parameters

TSeq	

Parameters

exclude	unsigned vector of states that need to be excluded from the sampling

Returns

Virus<TSeq>∗ of the selected virus. If none selected (or none available,) returns a nullptr;

12.2.2.3 sample_virus_single()

```
template<typename TSeq = int>
Virus<TSeq>* sampler::sample_virus_single (
            Agent< TSeq > * p,
            Model< TSeq > * m )  [inline]
```

Sample from neighbors pool of viruses (at most one)

This function samples at most one virus from the pool of viruses from its neighbors. If no virus is selected, the function returns a `nullptr`, otherwise it returns a pointer to the selected virus.

This can be used to build a new update function (EPI_NEW_UPDATEFUN.)

Template Parameters

TSeq	

Parameters

p	Pointer to person
m	Pointer to the model

Returns

Virus<TSeq>∗ of the selected virus. If none selected (or none available,) returns a nullptr;

Chapter 13

Class Documentation

13.1 Action< TSeq > Struct Template Reference

Action data for update an agent.

```
#include <config.hpp>
```

Public Member Functions

- Action (Agent< TSeq > ∗agent_, VirusPtr< TSeq > virus_, ToolPtr< TSeq > tool_, Entity< TSeq > ∗entity_, epiworld_fast_int new_status_, epiworld_fast_int queue_, ActionFun< TSeq > call_, int idx_agent↩ _, int idx_object_)

  Construct a new Action object.

Public Attributes

- Agent< TSeq > ∗ **agent**
- VirusPtr< TSeq > **virus**
- ToolPtr< TSeq > **tool**
- Entity< TSeq > ∗ **entity**
- epiworld_fast_int **new_status**
- epiworld_fast_int **queue**
- ActionFun< TSeq > **call**
- int **idx_agent**
- int **idx_object**

13.1.1 Detailed Description

template<**typename TSeq**>
struct Action< **TSeq** >

Action data for update an agent.

Template Parameters

TSeq	

13.1.2 Constructor & Destructor Documentation

13.1.2.1 Action()

```
template<typename TSeq >
Action< TSeq >::Action (
            Agent< TSeq > * agent_,
            VirusPtr< TSeq > virus_,
            ToolPtr< TSeq > tool_,
            Entity< TSeq > * entity_,
            epiworld_fast_int new_status_,
            epiworld_fast_int queue_,
            ActionFun< TSeq > call_,
            int idx_agent_,
            int idx_object_ )  [inline]
```

Construct a new Action object.

All the parameters are rather optional.

Parameters

*agent_*	Agent over who the action will happen
*virus_*	Virus to add
*tool_*	Tool to add
virus_idx	Index of virus to be removed (if needed)
tool_idx	Index of tool to be removed (if needed)
*new_↩ status_*	Next status
*queue_*	Efect on the queue
*call_*	The action call (if needed)
idx_agent↩ _	Location of agent in object.
idx_object↩ _	Location of object in agent.

The documentation for this struct was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/config.hpp

13.2 epiworld::Action< TSeq > Struct Template Reference

Action data for update an agent.

```
#include <epiworld.hpp>
```

Public Member Functions

- Action (Agent< TSeq > *agent_, VirusPtr< TSeq > virus_, ToolPtr< TSeq > tool_, Entity< TSeq > *entity_, epiworld_fast_int new_status_, epiworld_fast_int queue_, ActionFun< TSeq > call_, int idx_agent↩ _, int idx_object_)

  Construct a new Action object.

Public Attributes

- Agent< TSeq > * **agent**
- VirusPtr< TSeq > **virus**
- ToolPtr< TSeq > **tool**
- Entity< TSeq > * **entity**
- epiworld_fast_int **new_status**
- epiworld_fast_int **queue**
- ActionFun< TSeq > **call**
- int **idx_agent**
- int **idx_object**

13.2.1 Detailed Description

template<**typename TSeq**>
struct epiworld::Action< TSeq >

Action data for update an agent.

Template Parameters

TSeq	

13.2.2 Constructor & Destructor Documentation

13.2.2.1 Action()

```
template<typename TSeq >
epiworld::Action< TSeq >::Action (
            Agent< TSeq > * agent_,
```

```
VirusPtr< TSeq > virus_,
ToolPtr< TSeq > tool_,
Entity< TSeq > * entity_,
epiworld_fast_int new_status_,
epiworld_fast_int queue_,
ActionFun< TSeq > call_,
int idx_agent_,
int idx_object_ )  [inline]
```

Construct a new Action object.

All the parameters are rather optional.

Parameters

*agent_*	Agent over who the action will happen
*virus_*	Virus to add
*tool_*	Tool to add
virus_idx	Index of virus to be removed (if needed)
tool_idx	Index of tool to be removed (if needed)
*new_↩ status_*	Next status
*queue_*	Efect on the queue
*call_*	The action call (if needed)
idx_agent↩ _	Location of agent in object.
idx_object↩ _	Location of object in agent.

The documentation for this struct was generated from the following file:

- epiworld.hpp

13.3   AdjList Class Reference

Public Member Functions

- AdjList (const std::vector< epiworld_fast_uint > &source, const std::vector< epiworld_fast_uint > &target, int size, bool directed)

    Construct a new Adj List object.
- **AdjList** (AdjList &&a)
- **AdjList** (const AdjList &a)
- AdjList & **operator=** (const AdjList &a)
- void read_edgelist (std::string fn, int size, int skip=0, bool directed=true)

    Read an edgelist.
- std::map< unsigned int, unsigned int > **operator()** (unsigned int i) const
- void **print** (unsigned int limit=20u) const
- size_t vcount () const

    Number of vertices/nodes in the network.
- size_t ecount () const

    Number of edges/arcs/ties in the network.
- std::vector< std::map< unsigned int, unsigned int > > & **get_dat** ()
- bool is_directed () const

    `true` if the network is directed.

13.3.1 Constructor & Destructor Documentation

13.3.1.1 AdjList()

```
AdjList::AdjList (
            const std::vector< epiworld_fast_uint > & source,
            const std::vector< epiworld_fast_uint > & target,
            int size,
            bool directed ) [inline]
```

Construct a new Adj List object.

Ids in the network are assume to range from `0` to `size - 1`.

Parameters

source	Unsigned int vector with the source
target	Unsigned int vector with the target
size	Number of vertices in the network.
directed	Bool true if the network is directed

13.3.2 Member Function Documentation

13.3.2.1 read_edgelist()

```
void AdjList::read_edgelist (
            std::string fn,
            int size,
            int skip = 0,
            bool directed = true ) [inline]
```

Read an edgelist.

Ids in the network are assume to range from `0` to `size - 1`.

Parameters

fn	Path to the file
skip	Number of lines to skip (e.g., 1 if there's a header)
directed	`true` if the network is directed
size	Number of vertices in the network.

The documentation for this class was generated from the following files:

- include/epiworld/adjlist-bones.hpp
- include/epiworld/adjlist-meat.hpp

13.4 epiworld::AdjList Class Reference

Public Member Functions

- AdjList (const std::vector< epiworld_fast_uint > &source, const std::vector< epiworld_fast_uint > &target, int size, bool directed)

    Construct a new Adj List object.
- **AdjList** (AdjList &&a)
- **AdjList** (const AdjList &a)
- AdjList & **operator=** (const AdjList &a)
- void read_edgelist (std::string fn, int size, int skip=0, bool directed=true)

    Read an edgelist.
- std::map< unsigned int, unsigned int > **operator()** (unsigned int i) const
- void **print** (unsigned int limit=20u) const
- size_t vcount () const

    Number of vertices/nodes in the network.
- size_t ecount () const

    Number of edges/arcs/ties in the network.
- std::vector< std::map< unsigned int, unsigned int > > & **get_dat** ()
- bool is_directed () const

    `true` *if the network is directed.*

13.4.1 Constructor & Destructor Documentation

13.4.1.1 AdjList()

```
AdjList::AdjList (
            const std::vector< epiworld_fast_uint > & source,
            const std::vector< epiworld_fast_uint > & target,
            int size,
            bool directed )  [inline]
```

Construct a new Adj List object.

Ids in the network are assume to range from `0` to `size - 1`.

Parameters

source	Unsigned int vector with the source
target	Unsigned int vector with the target
size	Number of vertices in the network.
directed	Bool true if the network is directed

13.4.2 Member Function Documentation

13.4.2.1 read_edgelist()

```
void AdjList::read_edgelist (
            std::string fn,
            int size,
            int skip = 0,
            bool directed = true )  [inline]
```

Read an edgelist.

Ids in the network are assume to range from `0` to `size - 1`.

Parameters

fn	Path to the file
skip	Number of lines to skip (e.g., 1 if there's a header)
directed	`true` if the network is directed
size	Number of vertices in the network.

The documentation for this class was generated from the following file:

- epiworld.hpp

13.5 Agent< TSeq > Class Template Reference

Agent (agents)

```
#include <agent-bones.hpp>
```

Public Member Functions

- **Agent** (const Agent< TSeq > &p)
- int get_id () const
    - *Id of the individual.*
- std::mt19937 ∗ **get_rand_endgine** ()
- Model< TSeq > ∗ **get_model** ()
- VirusPtr< TSeq > & **get_virus** (int i)
- Viruses< TSeq > **get_viruses** ()
- const Viruses_const< TSeq > **get_viruses** () const
- size_t **get_n_viruses** () const noexcept
- ToolPtr< TSeq > & **get_tool** (int i)
- Tools< TSeq > **get_tools** ()
- const Tools_const< TSeq > **get_tools** () const
- size_t **get_n_tools** () const noexcept

- void **mutate_variant** ()
- void **add_neighbor** ([Agent](#)< TSeq > ∗p, bool check_source=true, bool check_target=true)
- std::vector< [Agent](#)< TSeq > ∗ > & **get_neighbors** ()
- void **change_status** (epiworld_fast_uint new_status, epiworld_fast_int queue=0)
- const epiworld_fast_uint & **get_status** () const
- void **reset** ()
- bool **has_tool** (unsigned int t) const
- bool **has_tool** (std::string name) const
- bool **has_virus** (unsigned int t) const
- bool **has_virus** (std::string name) const
- void **print** (bool compressed=false) const
- [Entities](#)< TSeq > **get_entities** ()
- const [Entities_const](#)< TSeq > **get_entities** () const

Add/Remove Virus/Tool

Any of these is ultimately reflected at the end of the iteration.

Parameters

tool	*[Tool](#) to add*
virus	*[Virus](#) to add*
status_new	*Status after the change*
queue	

- void **add_tool** (ToolPtr< TSeq > tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **add_tool** ([Tool](#)< TSeq > tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **add_virus** (VirusPtr< TSeq > virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **add_virus** ([Virus](#)< TSeq > virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **add_entity** ([Entity](#)< TSeq > &entity, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (epiworld_fast_uint tool_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (ToolPtr< TSeq > &tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_virus** (epiworld_fast_uint virus_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_virus** (VirusPtr< TSeq > &virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** (epiworld_fast_uint entity_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** ([Entity](#)< TSeq > &entity, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void [rm_agent_by_virus](#) (epiworld_fast_uint virus_idx, epiworld_fast_int status_new=-99, epiworld_fast↩_int queue=-99)

  [Agent](#) removed by virus.
- void [rm_agent_by_virus](#) (VirusPtr< TSeq > &virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)

  [Agent](#) removed by virus.

Get the rates (multipliers) for the agent

Parameters

v	*A pointer to a virus.*

Returns

> *epiworld_double*

- epiworld_double **get_susceptibility_reduction** (VirusPtr< TSeq > v)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v)

- double & operator() (size_t j)

  Access the j-th column of the agent.
- double & **operator[ ]** (size_t j)

Friends

- class **Model**< **TSeq** >
- class **Virus**< **TSeq** >
- class **Viruses**< **TSeq** >
- class **Viruses_const**< **TSeq** >
- class **Tool**< **TSeq** >
- class **Tools**< **TSeq** >
- class **Tools_const**< **TSeq** >
- class **Queue**< **TSeq** >
- class **Entities**< **TSeq** >
- class **AgentsSample**< **TSeq** >
- void **default_add_virus** (Action< TSeq > &a, Model< TSeq > ∗m)
- void **default_add_tool** (Action< TSeq > &a, Model< TSeq > ∗m)
- void **default_add_entity** (Action< TSeq > &a, Model< TSeq > ∗m)
- void **default_rm_virus** (Action< TSeq > &a, Model< TSeq > ∗m)
- void **default_rm_tool** (Action< TSeq > &a, Model< TSeq > ∗m)
- void **default_rm_entity** (Action< TSeq > &a, Model< TSeq > ∗m)

13.5.1 Detailed Description

template<**typename TSeq**>
class Agent< **TSeq** >

Agent (agents)

Template Parameters

TSeq	Sequence type (should match `TSeq` across the model)

13.5.2 Member Function Documentation

13.5.2.1 operator()()

```
template<typename TSeq >
double & Agent< TSeq >::operator() (
            size_t j )  [inline]
```

Access the j-th column of the agent.

If an external array has been specified, then these two functions can be used to access additional agent's features not included in the model.

The `operator[]` method is with no boundary check, whereas the `operator()` method checks boundaries. The former can result in a segfault.

Parameters

j	

Returns

> double&

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/agent-meat.hpp

13.6 epiworld::Agent< TSeq > Class Template Reference

Agent (agents)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Agent** (const Agent< TSeq > &p)
- int get_id () const
    Id of the individual.
- std::mt19937 ∗ **get_rand_endgine** ()
- Model< TSeq > ∗ **get_model** ()
- VirusPtr< TSeq > & **get_virus** (int i)
- Viruses< TSeq > **get_viruses** ()
- const Viruses_const< TSeq > **get_viruses** () const
- size_t **get_n_viruses** () const noexcept
- ToolPtr< TSeq > & **get_tool** (int i)
- Tools< TSeq > **get_tools** ()
- const Tools_const< TSeq > **get_tools** () const
- size_t **get_n_tools** () const noexcept
- void **mutate_variant** ()
- void **add_neighbor** (Agent< TSeq > ∗p, bool check_source=true, bool check_target=true)

- std::vector< [Agent]< TSeq > ∗ > & **get_neighbors** ()
- void **change_status** (epiworld_fast_uint new_status, epiworld_fast_int queue=0)
- const epiworld_fast_uint & **get_status** () const
- void **reset** ()
- bool **has_tool** (unsigned int t) const
- bool **has_tool** (std::string name) const
- bool **has_virus** (unsigned int t) const
- bool **has_virus** (std::string name) const
- void **print** (bool compressed=false) const
- [Entities]< TSeq > **get_entities** ()
- const [Entities_const]< TSeq > **get_entities** () const

Add/Remove Virus/Tool

Any of these is ultimately reflected at the end of the iteration.

Parameters

tool	*Tool to add*
virus	*Virus to add*
status_new	*Status after the change*
queue	

- void **add_tool** (ToolPtr< TSeq > tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **add_tool** ([Tool]< TSeq > tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **add_virus** (VirusPtr< TSeq > virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **add_virus** ([Virus]< TSeq > virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **add_entity** ([Entity]< TSeq > &entity, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (epiworld_fast_uint tool_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (ToolPtr< TSeq > &tool, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_virus** (epiworld_fast_uint virus_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_virus** (VirusPtr< TSeq > &virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** (epiworld_fast_uint entity_idx, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** ([Entity]< TSeq > &entity, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)
- void [rm_agent_by_virus] (epiworld_fast_uint virus_idx, epiworld_fast_int status_new=-99, epiworld_fast←↩ _int queue=-99)

    [Agent] removed by virus.
- void [rm_agent_by_virus] (VirusPtr< TSeq > &virus, epiworld_fast_int status_new=-99, epiworld_fast_int queue=-99)

    [Agent] removed by virus.

Get the rates (multipliers) for the agent

Parameters

v	*A pointer to a virus.*

Returns

epiworld_double

- • epiworld_double **get_susceptibility_reduction** (VirusPtr< TSeq > v)
- • epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v)
- • epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v)
- • epiworld_double **get_death_reduction** (VirusPtr< TSeq > v)

- • double & operator() (size_t j)

    Access the j-th column of the agent.
- • double & **operator[ ]** (size_t j)

Friends

- • class **Model**< **TSeq** >
- • class **Virus**< **TSeq** >
- • class **Viruses**< **TSeq** >
- • class **Viruses_const**< **TSeq** >
- • class **Tool**< **TSeq** >
- • class **Tools**< **TSeq** >
- • class **Tools_const**< **TSeq** >
- • class **Queue**< **TSeq** >
- • class **Entities**< **TSeq** >
- • class **AgentsSample**< **TSeq** >
- • void **default_add_virus** (Action< TSeq > &a, Model< TSeq > ∗m)
- • void **default_add_tool** (Action< TSeq > &a, Model< TSeq > ∗m)
- • void **default_add_entity** (Action< TSeq > &a, Model< TSeq > ∗m)
- • void **default_rm_virus** (Action< TSeq > &a, Model< TSeq > ∗m)
- • void **default_rm_tool** (Action< TSeq > &a, Model< TSeq > ∗m)
- • void **default_rm_entity** (Action< TSeq > &a, Model< TSeq > ∗m)

13.6.1 Detailed Description

template<**typename TSeq**>
class epiworld::Agent< **TSeq** >

Agent (agents)

Template Parameters

TSeq	Sequence type (should match `TSeq` across the model)

13.6.2 Member Function Documentation

13.6.2.1 operator()()

```
template<typename TSeq >
double & Agent< TSeq >::operator() (
            size_t j ) [inline]
```

Access the j-th column of the agent.

If an external array has been specified, then these two functions can be used to access additional agent's features not included in the model.

The `operator[]` method is with no boundary check, whereas the `operator()` method checks boundaries. The former can result in a segfault.

Parameters

j	

Returns

double&

The documentation for this class was generated from the following file:

- epiworld.hpp

13.7 AgentsSample$<$ TSeq $>$ Class Template Reference

Sample of agents.

```
#include <agentssample-bones.hpp>
```

Public Member Functions

- AgentsSample ()=delete
    Default constructor.
- AgentsSample (const AgentsSample$<$ TSeq $>$ &a)=delete
    Copy constructor.
- AgentsSample (AgentsSample$<$ TSeq $>$ &&a)=delete
    Move constructor.
- **AgentsSample** (Model$<$ TSeq $>$ &model_, size_t n, bool truncate=false)
- **AgentsSample** (Entity$<$ TSeq $>$ &entity_, size_t n, bool truncate=false)
- **AgentsSample** (Agent$<$ TSeq $>$ &agent_, size_t n, bool truncate=false)
- std::vector$<$ Agent$<$ TSeq $>$ $*$ $>$::iterator **begin** ()
- std::vector$<$ Agent$<$ TSeq $>$ $*$ $>$::iterator **end** ()
- Agent$<$ TSeq $>$ $*$ **operator[]** (size_t n)
- Agent$<$ TSeq $>$ $*$ **operator()** (size_t n)
- const size_t **size** () const noexcept

13.7.1 Detailed Description

template$<$**typename TSeq**$>$
class AgentsSample$<$ **TSeq** $>$

Sample of agents.

This class allows sampling agents from Entity$<$TSeq$>$ and Model$<$TSeq$>$.

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- include/epiworld/agentssample-bones.hpp

13.8 epiworld::AgentsSample< TSeq > Class Template Reference

Sample of agents.

```
#include <epiworld.hpp>
```

Public Member Functions

- AgentsSample ()=delete
  Default constructor.
- AgentsSample (const AgentsSample< TSeq > &a)=delete
  Copy constructor.
- AgentsSample (AgentsSample< TSeq > &&a)=delete
  Move constructor.
- **AgentsSample** (Model< TSeq > &model_, size_t n, bool truncate=false)
- **AgentsSample** (Entity< TSeq > &entity_, size_t n, bool truncate=false)
- **AgentsSample** (Agent< TSeq > &agent_, size_t n, bool truncate=false)
- std::vector< Agent< TSeq > ∗ >::iterator **begin** ()
- std::vector< Agent< TSeq > ∗ >::iterator **end** ()
- Agent< TSeq > ∗ **operator[]** (size_t n)
- Agent< TSeq > ∗ **operator()** (size_t n)
- const size_t **size** () const noexcept

13.8.1 Detailed Description

template<**typename TSeq**>
class epiworld::AgentsSample< TSeq >

Sample of agents.

This class allows sampling agents from Entity<TSeq> and Model<TSeq>.

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- epiworld.hpp

13.9 DataBase$<$ TSeq $>$ Class Template Reference

Statistical data about the process.

```
#include <database-bones.hpp>
```

Public Member Functions

- **DataBase** (Model$<$ TSeq $>$ &m)
- void record_variant (Virus$<$ TSeq $>$ &v)

  Registering a new variant.
- void **record_tool** (Tool$<$ TSeq $>$ &t)
- void **set_seq_hasher** (std::function$<$ std::vector$<$ int $>$(TSeq)$>$ fun)
- void **set_model** (Model$<$ TSeq $>$ &m)
- Model$<$ TSeq $>$ ∗ **get_model** ()
- void **record** ()
- const std::vector$<$ TSeq $>$ & **get_sequence** () const
- const std::vector$<$ int $>$ & **get_nexposed** () const
- size_t **size** () const
- void **write_data** (std::string fn_variant_info, std::string fn_variant_hist, std::string fn_tool_info, std::string fn_tool_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition, std::string fn_↩ reproductive_number) const
- void **record_transmission** (int i, int j, int variant, int i_expo_date)
- size_t **get_n_variants** () const
- size_t **get_n_tools** () const
- void **reset** ()
- void **set_user_data** (std::vector$<$ std::string $>$ names)
- void **add_user_data** (std::vector$<$ epiworld_double $>$ x)
- void **add_user_data** (unsigned int j, epiworld_double x)
- UserData$<$ TSeq $>$ & **get_user_data** ()
- std::vector$<$ epiworld_double $>$ transition_probability (bool print=true) const

  Calculates the transition probabilities.

Get recorded information from the model

Parameters

what	*std::string, The status, e.g., 0, 1, 2, ...*

Returns

> In `get_today_total`, *the current counts of* `what`.
>
> In `get_today_variant`, *the current counts of* `what` *for each variant.*
>
> In `get_hist_total`, *the time series of* `what`
>
> In `get_hist_variant`, *the time series of what for each variant.*
>
> In `get_hist_total_date` *and* `get_hist_variant_date` *the corresponding dates*

- int **get_today_total** (std::string what) const
- int **get_today_total** (epiworld_fast_uint what) const
- void **get_today_total** (std::vector$<$ std::string $>$ ∗status=nullptr, std::vector$<$ int $>$ ∗counts=nullptr) const
- void **get_today_variant** (std::vector$<$ std::string $>$ &status, std::vector$<$ int $>$ &id, std::vector$<$ int $>$ &counts) const

- void **get_hist_total** (std::vector< int > *date, std::vector< std::string > *status, std::vector< int > *counts) const
- void **get_hist_variant** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &status, std::vector< int > &counts) const

- MapVec_type< int, int > [reproductive_number](#) () const

  Computes the reproductive number of each case.
- void **reproductive_number** (std::string fn) const

Friends

- class **Model**< **TSeq** >
- void **default_add_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_add_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > *m)
- void **default_rm_tool** ([Action](#)< TSeq > &a, [Model](#)< TSeq > *m)

13.9.1 Detailed Description

template<**typename TSeq**>
class DataBase< **TSeq** >

Statistical data about the process.

Template Parameters

TSeq	

13.9.2 Member Function Documentation

13.9.2.1 record_variant()

```
template<typename TSeq >
void DataBase< TSeq >::record_variant (
            Virus< TSeq > & v )  [inline]
```

Registering a new variant.

Parameters

v	Pointer to the new variant. Since variants are originated in the agent, the numbers simply move around. From the parent variant to the new variant. And the total number of infected does not change.

13.9.2.2 reproductive_number()

```
template<typename TSeq >
MapVec_type< int, int > DataBase< TSeq >::reproductive_number  [inline]
```

Computes the reproductive number of each case.

By definition, whereas it computes R0 (basic reproductive number) or Rt/R (the effective reproductive number) will depend on whether the virus is allowed to circulate naïvely or not, respectively.

Parameters

fn	File where to write out the reproductive number.

13.9.2.3 transition_probability()

```
template<typename TSeq >
std::vector< epiworld_double > DataBase< TSeq >::transition_probability (
            bool print = true ) const  [inline]
```

Calculates the transition probabilities.

Returns

std::vector< epiworld_double >

The documentation for this class was generated from the following files:

- include/epiworld/database-bones.hpp
- include/epiworld/database-meat.hpp

13.10 epiworld::DataBase< TSeq > Class Template Reference

Statistical data about the process.

```
#include <epiworld.hpp>
```

Public Member Functions

- **DataBase** ([Model](TSeq)< TSeq > &m)
- void [record_variant](Virus) ([Virus](TSeq)< TSeq > &v)

    Registering a new variant.

- void **record_tool** ([Tool](TSeq)< TSeq > &t)
- void **set_seq_hasher** (std::function< std::vector< int >(TSeq)> fun)
- void **set_model** ([Model](TSeq)< TSeq > &m)
- [Model](TSeq)< TSeq > ∗ **get_model** ()
- void **record** ()
- const std::vector< TSeq > & **get_sequence** () const
- const std::vector< int > & **get_nexposed** () const
- size_t **size** () const
- void **write_data** (std::string fn_variant_info, std::string fn_variant_hist, std::string fn_tool_info, std::string fn_tool_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition, std::string fn_↩ reproductive_number) const
- void **record_transmission** (int i, int j, int variant, int i_expo_date)
- size_t **get_n_variants** () const
- size_t **get_n_tools** () const
- void **reset** ()
- void **set_user_data** (std::vector< std::string > names)
- void **add_user_data** (std::vector< epiworld_double > x)
- void **add_user_data** (unsigned int j, epiworld_double x)
- [UserData](TSeq)< TSeq > & **get_user_data** ()
- std::vector< epiworld_double > [transition_probability](bool) (bool print=true) const

    Calculates the transition probabilities.

Get recorded information from the model

Parameters

what	*std::string, The status, e.g., 0, 1, 2, ...*

Returns

> In `get_today_total`, *the current counts of* `what`.
> In `get_today_variant`, *the current counts of* `what` *for each variant.*
> In `get_hist_total`, *the time series of* `what`
> In `get_hist_variant`, *the time series of what for each variant.*
> In `get_hist_total_date` *and* `get_hist_variant_date` *the corresponding dates*

- int **get_today_total** (std::string what) const
- int **get_today_total** (epiworld_fast_uint what) const
- void **get_today_total** (std::vector< std::string > ∗status=nullptr, std::vector< int > ∗counts=nullptr) const
- void **get_today_variant** (std::vector< std::string > &status, std::vector< int > &id, std::vector< int > &counts) const
- void **get_hist_total** (std::vector< int > ∗date, std::vector< std::string > ∗status, std::vector< int > ∗counts) const
- void **get_hist_variant** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &status, std::vector< int > &counts) const

- MapVec_type< int, int > [reproductive_number](TSeq) () const

    Computes the reproductive number of each case.

- void **reproductive_number** (std::string fn) const

Friends

- class **Model**< **TSeq** >
- void **default_add_virus** ([Action](< TSeq > &a, [Model](< TSeq > *m)
- void **default_add_tool** ([Action](< TSeq > &a, [Model](< TSeq > *m)
- void **default_rm_virus** ([Action](< TSeq > &a, [Model](< TSeq > *m)
- void **default_rm_tool** ([Action](< TSeq > &a, [Model](< TSeq > *m)

13.10.1 Detailed Description

template<**typename TSeq**>
class epiworld::DataBase< **TSeq** >

Statistical data about the process.

Template Parameters

TSeq	

13.10.2 Member Function Documentation

13.10.2.1 record_variant()

```
template<typename TSeq >
void DataBase< TSeq >::record_variant (
            Virus< TSeq > & v )  [inline]
```

Registering a new variant.

Parameters

v	Pointer to the new variant. Since variants are originated in the agent, the numbers simply move around. From the parent variant to the new variant. And the total number of infected does not change.

13.10.2.2 reproductive_number()

```
template<typename TSeq >
MapVec_type< int, int > DataBase< TSeq >::reproductive_number  [inline]
```

Computes the reproductive number of each case.

By definition, whereas it computes R0 (basic reproductive number) or Rt/R (the effective reproductive number) will depend on whether the virus is allowed to circulate naïvely or not, respectively.

| *fn* | File where to write out the reproductive number. |

13.10.2.3 transition_probability()

```
template<typename TSeq >
std::vector< epiworld_double > DataBase< TSeq >::transition_probability (
            bool print = true ) const  [inline]
```

Calculates the transition probabilities.

Returns

std::vector< epiworld_double >

The documentation for this class was generated from the following file:

- epiworld.hpp

13.11 Entities< TSeq > Class Template Reference

Set of Entities (useful for building iterators)

```
#include <entities-bones.hpp>
```

Public Member Functions

- **Entities** (Agent< TSeq > &p)
- std::vector< Entity< TSeq > ∗ >::iterator **begin** ()
- std::vector< Entity< TSeq > ∗ >::iterator **end** ()
- Entity< TSeq > ∗& **operator()** (size_t i)
- Entity< TSeq > ∗& **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Entity**< **TSeq** >
- class **Agent**< **TSeq** >

13.11.1 Detailed Description

template<**typename TSeq**>
class Entities< **TSeq** >

Set of Entities (useful for building iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/entities-bones.hpp

13.12 epiworld::Entities< TSeq > Class Template Reference

Set of Entities (useful for building iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Entities** (Agent< TSeq > &p)
- std::vector< Entity< TSeq > ∗ >::iterator **begin** ()
- std::vector< Entity< TSeq > ∗ >::iterator **end** ()
- Entity< TSeq > ∗& **operator()** (size_t i)
- Entity< TSeq > ∗& **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Entity**< **TSeq** >
- class **Agent**< **TSeq** >

13.12.1 Detailed Description

template<**typename TSeq**>
class epiworld::Entities< **TSeq** >

Set of Entities (useful for building iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- epiworld.hpp

13.13   Entities_const< TSeq > Class Template Reference

Set of [Entities](#) (const) (useful for iterators)

```
#include <entities-bones.hpp>
```

Public Member Functions

- **Entities_const** (const [Agent](#)< TSeq > &p)
- std::vector< [Entity](#)< TSeq > ∗ >::const_iterator **begin** ()
- std::vector< [Entity](#)< TSeq > ∗ >::const_iterator **end** ()
- const [Entity](#)< TSeq > ∗& **operator()** (size_t i)
- const [Entity](#)< TSeq > ∗& **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Virus< TSeq >**
- class **Agent< TSeq >**

13.13.1   Detailed Description

template<**typename TSeq**>
class Entities_const< TSeq >

Set of [Entities](#) (const) (useful for iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- include/epiworld/entities-bones.hpp

13.14   epiworld::Entities_const< TSeq > Class Template Reference

Set of [Entities](#) (const) (useful for iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Entities_const** (const [Agent](#)< TSeq > &p)
- std::vector< [Entity](#)< TSeq > ∗ >::const_iterator **begin** ()
- std::vector< [Entity](#)< TSeq > ∗ >::const_iterator **end** ()
- const [Entity](#)< TSeq > ∗& **operator()** (size_t i)
- const [Entity](#)< TSeq > ∗& **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Virus**< **TSeq** >
- class **Agent**< **TSeq** >

13.14.1 Detailed Description

template<**typename TSeq**>
class epiworld::Entities_const< **TSeq** >

Set of [Entities](#) (const) (useful for iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- epiworld.hpp

13.15 Entity< TSeq > Class Template Reference

Public Member Functions

- **Entity** (std::string name)
- void **add_agent** ([Agent](#)< TSeq > &p)
- void **add_agent** ([Agent](#)< TSeq > ∗p)
- void **rm_agent** (size_t idx)
- size_t **size** () const noexcept
- void **set_location** (std::vector< epiworld_double > loc)
- std::vector< epiworld_double > & **get_location** ()
- std::vector< [Agent](#)< TSeq > ∗ >::iterator **begin** ()
- std::vector< [Agent](#)< TSeq > ∗ >::iterator **end** ()
- std::vector< [Agent](#)< TSeq > ∗ >::const_iterator **begin** () const
- std::vector< [Agent](#)< TSeq > ∗ >::const_iterator **end** () const
- [Agent](#)< TSeq > ∗ **operator[ ]** (size_t i)
- int **get_id** () const noexcept
- const std::string & **get_name** () const noexcept
- void **set_status** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_status** (epiworld_fast_int ∗init, epiworld_fast_int ∗post)
- void **get_queue** (epiworld_fast_int ∗init, epiworld_fast_int ∗post)

Friends

- class **Agent**< **TSeq** >
- class **AgentsSample**< **TSeq** >
- class **Model**< **TSeq** >
- void **default_add_entity** ([Action](#)< TSeq > &a, [Model](#)< TSeq > ∗m)
- void **default_rm_entity** ([Action](#)< TSeq > &a, [Model](#)< TSeq > ∗m)

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/entity-bones.hpp
- include/epiworld/entity-meat.hpp

13.16 epiworld::Entity< TSeq > Class Template Reference

Public Member Functions

- **Entity** (std::string name)
- void **add_agent** ([Agent](#)< TSeq > &p)
- void **add_agent** ([Agent](#)< TSeq > ∗p)
- void **rm_agent** (size_t idx)
- size_t **size** () const noexcept
- void **set_location** (std::vector< epiworld_double > loc)
- std::vector< epiworld_double > & **get_location** ()
- std::vector< [Agent](#)< TSeq > ∗ >::iterator **begin** ()
- std::vector< [Agent](#)< TSeq > ∗ >::iterator **end** ()
- std::vector< [Agent](#)< TSeq > ∗ >::const_iterator **begin** () const
- std::vector< [Agent](#)< TSeq > ∗ >::const_iterator **end** () const
- [Agent](#)< TSeq > ∗ **operator[]** (size_t i)
- int **get_id** () const noexcept
- const std::string & **get_name** () const noexcept
- void **set_status** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_status** (epiworld_fast_int ∗init, epiworld_fast_int ∗post)
- void **get_queue** (epiworld_fast_int ∗init, epiworld_fast_int ∗post)

Friends

- class **Agent**< **TSeq** >
- class **AgentsSample**< **TSeq** >
- class **Model**< **TSeq** >
- void **default_add_entity** ([Action](#)< TSeq > &a, [Model](#)< TSeq > ∗m)
- void **default_rm_entity** ([Action](#)< TSeq > &a, [Model](#)< TSeq > ∗m)

The documentation for this class was generated from the following file:

- epiworld.hpp

13.17 epiworld::LFMCMC< TData > Class Template Reference

Likelihood-Free Markov Chain Monte Carlo.

```
#include <epiworld.hpp>
```

Public Member Functions

- void **run** (std::vector< epiworld_double > param_init, size_t n_samples_, epiworld_double epsilon_)
- **LFMCMC** (TData &observed_data_)
- void **set_observed_data** (TData &observed_data_)
- void **set_proposal_fun** (LFMCMCProposalFun< TData > fun)
- void **set_simulation_fun** (LFMCMCSimFun< TData > fun)
- void **set_summary_fun** (LFMCMCSummaryFun< TData > fun)
- void **set_kernel_fun** (LFMCMCKernelFun< TData > fun)
- size_t **get_n_samples** () const
- size_t **get_n_statistics** () const
- size_t **get_n_parameters** () const
- epiworld_double **get_epsilon** () const
- const std::vector< epiworld_double > & **get_params_now** ()
- const std::vector< epiworld_double > & **get_params_prev** ()
- const std::vector< epiworld_double > & **get_params_init** ()
- const std::vector< epiworld_double > & **get_statistics_obs** ()
- const std::vector< epiworld_double > & **get_statistics_hist** ()
- const std::vector< bool > & **get_statistics_accepted** ()
- const std::vector< epiworld_double > & **get_posterior_lf_prob** ()
- const std::vector< epiworld_double > & **get_drawn_prob** ()
- std::vector< TData > ∗ **get_sampled_data** ()
- void **set_par_names** (std::vector< std::string > names)
- void **set_stats_names** (std::vector< std::string > names)
- void **print** ()

Random number generation

Parameters

eng	

- void **set_rand_engine** (std::mt19937 &eng)
- std::mt19937 ∗ **get_rand_endgine** ()
- void **seed** (unsigned int s)
- void **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- epiworld_double **runif** ()
- epiworld_double **rnorm** ()
- epiworld_double **rgamma** ()
- epiworld_double **runif** (epiworld_double lb, epiworld_double ub)
- epiworld_double **rnorm** (epiworld_double mean, epiworld_double sd)
- epiworld_double **rgamma** (epiworld_double alpha, epiworld_double beta)

13.17.1 Detailed Description

template<**typename TData**>
class epiworld::LFMCMC< **TData** >

Likelihood-Free Markov Chain Monte Carlo.

Template Parameters

| *TData* | Type of data that is generated |

The documentation for this class was generated from the following file:

- epiworld.hpp

13.18   LFMCMC< TData > Class Template Reference

Likelihood-Free Markov Chain Monte Carlo.

```
#include <lfmcmc-bones.hpp>
```

Public Member Functions

- void **run** (std::vector< epiworld_double > param_init, size_t n_samples_, epiworld_double epsilon_)
- **LFMCMC** (TData &observed_data_)
- void **set_observed_data** (TData &observed_data_)
- void **set_proposal_fun** (LFMCMCProposalFun< TData > fun)
- void **set_simulation_fun** (LFMCMCSimFun< TData > fun)
- void **set_summary_fun** (LFMCMCSummaryFun< TData > fun)
- void **set_kernel_fun** (LFMCMCKernelFun< TData > fun)
- size_t **get_n_samples** () const
- size_t **get_n_statistics** () const
- size_t **get_n_parameters** () const
- epiworld_double **get_epsilon** () const
- const std::vector< epiworld_double > & **get_params_now** ()
- const std::vector< epiworld_double > & **get_params_prev** ()
- const std::vector< epiworld_double > & **get_params_init** ()
- const std::vector< epiworld_double > & **get_statistics_obs** ()
- const std::vector< epiworld_double > & **get_statistics_hist** ()
- const std::vector< bool > & **get_statistics_accepted** ()
- const std::vector< epiworld_double > & **get_posterior_lf_prob** ()
- const std::vector< epiworld_double > & **get_drawn_prob** ()
- std::vector< TData > ∗ **get_sampled_data** ()
- void **set_par_names** (std::vector< std::string > names)
- void **set_stats_names** (std::vector< std::string > names)
- void **print** ()

Random number generation

Parameters

eng	

- void **set_rand_engine** (std::mt19937 &eng)
- std::mt19937 ∗ **get_rand_endgine** ()
- void **seed** (unsigned int s)
- void **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- epiworld_double **runif** ()
- epiworld_double **rnorm** ()
- epiworld_double **rgamma** ()
- epiworld_double **runif** (epiworld_double lb, epiworld_double ub)
- epiworld_double **rnorm** (epiworld_double mean, epiworld_double sd)
- epiworld_double **rgamma** (epiworld_double alpha, epiworld_double beta)

13.18.1   Detailed Description

template<**typename TData**>
class LFMCMC< **TData** >

Likelihood-Free Markov Chain Monte Carlo.

Template Parameters

TData	Type of data that is generated

The documentation for this class was generated from the following files:

- include/epiworld/math/lfmcmc/lfmcmc-bones.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat-print.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat.hpp

13.19   epiworld::Model< TSeq > Class Template Reference

Core class of epiworld.

```
#include <epiworld.hpp>
```

Public Member Functions

- [DataBase](#)< TSeq > & **get_db** ()
- epiworld_double & **operator()** (std::string pname)
- size_t **size** () const
- size_t **get_n_variants** () const
- size_t **get_n_tools** () const
- unsigned int **get_ndays** () const
- unsigned int **get_n_replicates** () const
- void **set_ndays** (unsigned int ndays)
- bool **get_verbose** () const

- void **verbose_off** ()
- void **verbose_on** ()
- int today () const

    The current time of the model.

- void write_data (std::string fn_variant_info, std::string fn_variant_hist, std::string fn_tool_info, std::string fn↩
    _tool_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition, std::string fn_↩
    reproductive_number) const

    Wrapper of `DataBase::write_data`

- std::map< std::string, epiworld_double > & **params** ()
- void reset ()

    Reset the model.

- void **print** () const
- Model< TSeq > && **clone** () const
- void **get_elapsed** (std::string unit="auto", epiworld_double ∗last_elapsed=nullptr, epiworld_double ∗total_↩
    elapsed=nullptr, std::string ∗unit_abbr=nullptr, bool print=true) const
- void add_global_action (std::function< void(Model< TSeq > ∗)> fun, int date=-99)

    Set a global action.

- void **run_global_actions** ()
- void **clear_status_set** ()
- const std::vector< VirusPtr< TSeq > > & **get_viruses** () const
- const std::vector< ToolPtr< TSeq > > & **get_tools** () const
- void set_agents_data (double ∗data_, size_t ncols_)

    Set the agents data object.

- void set_name (std::string name)

    Set the name object.

Set the backup object

`backup` can be used to restore the entire object after a run. This can be useful if the user wishes to have
individuals start with the same network from the beginning.

- void **set_backup** ()
- void **restore_backup** ()

Random number generation

Parameters

eng	*Random number generator*
s	*Seed*

- void **set_rand_engine** (std::mt19937 &eng)
- std::mt19937 ∗ **get_rand_endgine** ()
- void **seed** (unsigned int s)
- void **set_rand_norm** (epiworld_double mean, epiworld_double sd)
- void **set_rand_unif** (epiworld_double a, epiworld_double b)
- void **set_rand_exp** (epiworld_double lambda)
- void **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- void **set_rand_lognormal** (epiworld_double mean, epiworld_double shape)
- epiworld_double **runif** ()
- epiworld_double **runif** (epiworld_double a, epiworld_double b)
- epiworld_double **rnorm** ()
- epiworld_double **rnorm** (epiworld_double mean, epiworld_double sd)
- epiworld_double **rgamma** ()
- epiworld_double **rgamma** (epiworld_double alpha, epiworld_double beta)

- epiworld_double **rexp** ()
- epiworld_double **rexp** (epiworld_double lambda)
- epiworld_double **rlognormal** ()
- epiworld_double **rlognormal** (epiworld_double mean, epiworld_double shape)

Add Virus/Tool to the model

This is done before the model has been initialized.

Parameters

v	*Virus to be added*
t	*Tool to be added*
preval	*Initial prevalence (initial state.) It can be specified as a proportion (between zero and one,) or an integer indicating number of individuals.*

- void **add_virus** (Virus< TSeq > v, epiworld_double preval)
- void **add_virus_n** (Virus< TSeq > v, unsigned int preval)
- void **add_virus_fun** (Virus< TSeq > v, VirusToAgentFun< TSeq > fun)
- void **add_tool** (Tool< TSeq > t, epiworld_double preval)
- void **add_tool_n** (Tool< TSeq > t, unsigned int preval)
- void **add_tool_fun** (Tool< TSeq > t, ToolToAgentFun< TSeq > fun)
- void **add_entity** (Entity< TSeq > e, epiworld_double preval)
- void **add_entity_n** (Entity< TSeq > e, unsigned int preval)
- void **add_entity_fun** (Entity< TSeq > e, EntityToAgentFun< TSeq > fun)

Accessing population of the model

Parameters

fn	*std::string Filename of the edgelist file.*
skip	*int Number of lines to skip in `fn`.*
directed	*bool Whether the graph is directed or not.*
size	*Size of the network.*
al	*AdjList to read into the model.*

- void **agents_from_adjlist** (std::string fn, int size, int skip=0, bool directed=false)
- void **agents_from_adjlist** (AdjList al)
- bool **is_directed** () const
- std::vector< Agent< TSeq > > ∗ **get_agents** ()
- void **agents_smallworld** (unsigned int n=1000, unsigned int k=5, bool d=false, epiworld_double p=.01)
- void **agents_empty_graph** (unsigned int n=1000)

Functions to run the model

Parameters

seed	*Seed to be used for Pseudo-RNG.*
ndays	*Number of days (steps) of the simulation.*
fun	*In the case of `run_multiple`, a function that is called after each experiment.*

- void **init** (unsigned int ndays, unsigned int seed)
- void **update_status** ()

- void **mutate_variant** ()
- void **next** ()
- void run ()
  - *Runs the simulation (after initialization)*
- void run_multiple (unsigned int nexperiments, std::function$<$ void(size_t, Model$<$ TSeq $>$ $*$)$>$ fun=make_save_run$<$ TSeq $>$(), bool reset=true, bool verbose=true)

Rewire the network preserving the degree sequence.

This implementation assumes an undirected network, thus if {(i,j), (k,l)} -> {(i,l), (k,j)}, the reciprocal is also true, i.e., {(j,i), (l,k)} -> {(j,k), (l,i)}.

Parameters

proportion	*Proportion of ties to be rewired.*

Returns

> *A rewired version of the network.*

- void **set_rewire_fun** (std::function$<$ void(std::vector$<$ Agent$<$ TSeq $>>$ $*$, Model$<$ TSeq $>$ $*$, epiworld_double)$>$ fun)
- void **set_rewire_prop** (epiworld_double prop)
- epiworld_double **get_rewire_prop** () const
- void **rewire** ()

Export the network data in edgelist form

Parameters

fn	*std::string. File name.*
source	*Integer vector*
target	*Integer vector*

When passing the source and target, the function will write the edgelist on those.

- void **write_edgelist** (std::string fn) const
- void **write_edgelist** (std::vector$<$ unsigned int $>$ &source, std::vector$<$ unsigned int $>$ &target) const

Manage status (states) in the model

The functions `get_status` *return the current values for the statuses included in the model.*

Parameters

lab	`std::string` *Name of the status.*

Returns

> `add_status*` *returns nothing.*
>
> `get_status_*` *returns a vector of pairs with the statuses and their labels.*

- void **add_status** (std::string lab, UpdateFun$<$ TSeq $>$ fun=nullptr)
- const std::vector$<$ std::string $>$ & **get_status** () const
- const std::vector$<$ UpdateFun$<$ TSeq $>$ $>$ & **get_status_fun** () const
- void **print_status_codes** () const

Set the user data object

Parameters

names	*string vector with the names of the variables.*

- void [set_user_data](std::vector< std::string > names)
    [@
- void **add_user_data** (unsigned int j, epiworld_double x)
- void **add_user_data** (std::vector< epiworld_double > x)
- [UserData](std)< TSeq > & **get_user_data** ()

Queuing system

When queueing is on, the model will keep track of which agents are either in risk of exposure or exposed. This then is used at each step to act only on the aforementioned agents.

- void [queuing_on](())
    Activates the queuing system (default.)
- void [queuing_off](())
    Deactivates the queuing system.
- bool [is_queuing_on](()) const
    Query if the queuing system is on.
- [Queue](< TSeq >) & [get_queue](())
    Retrieve the Queue object.

Get the susceptibility reduction object

Parameters

v	

Returns

    epiworld_double

- void **set_susceptibility_reduction_mixer** (MixerFun< TSeq > fun)
- void **set_transmission_reduction_mixer** (MixerFun< TSeq > fun)
- void **set_recovery_enhancer_mixer** (MixerFun< TSeq > fun)
- void **set_death_reduction_mixer** (MixerFun< TSeq > fun)

Friends

- class **Agent< TSeq >**
- class **AgentsSample< TSeq >**
- class **DataBase< TSeq >**
- class **Queue< TSeq >**

Tool Mixers

These functions combine the effects tools have to deliver a single effect. For example, wearing a mask, been vaccinated, and the immune system combine together to jointly reduce the susceptibility for a given virus.

- std::vector< epiworld_double > **array_double_tmp**
- std::vector< [Virus](< TSeq > *) > **array_virus_tmp**

- **Model** ()
- **Model** (const Model< TSeq > &m)
- **Model** (Model< TSeq > &&m)
- Model< TSeq > & **operator=** (const Model< TSeq > &m)
- void **clone_population** (std::vector< Agent< TSeq > > &p, bool &d, Model< TSeq > ∗m=nullptr) const
- void **clone_population** (const Model< TSeq > &m)

Setting and accessing parameters from the model

Tools can incorporate parameters included in the model. Internally, parameters in the tool are stored as pointers to an std::map<> of parameters in the model. Using the `unsigned int` method directly fetches the parameters in the order these were added to the tool. Accessing parameters via the `std::string` method involves searching the parameter directly in the std::map<> member of the model (so it is not recommended.)

The `par()` function members are aliases for `get_param()`.

In the case of the function `read_params`, users can pass a file listing parameters to be included in the model. Each line in the file should have the following structure:
```
[name of parameter 1]: [value in double]
[name of parameter 2]: [value in double]
...
```

The only condition for parameter names is that these do not include a colon.

Parameters

initial_val	
pname	Name of the parameter to add or to fetch
fn	Path to the file containing parameters

Returns

> The current value of the parameter in the model.

- epiworld_double ∗ **p0**
- epiworld_double ∗ **p1**
- epiworld_double ∗ **p2**
- epiworld_double ∗ **p3**
- epiworld_double ∗ **p4**
- epiworld_double ∗ **p5**
- epiworld_double ∗ **p6**
- epiworld_double ∗ **p7**
- epiworld_double ∗ **p8**
- epiworld_double ∗ **p9**
- epiworld_double ∗ **p10**
- epiworld_double ∗ **p11**
- epiworld_double ∗ **p12**
- epiworld_double ∗ **p13**
- epiworld_double ∗ **p14**
- epiworld_double ∗ **p15**
- epiworld_double ∗ **p16**
- epiworld_double ∗ **p17**
- epiworld_double ∗ **p18**

- epiworld_double $*$ **p19**
- epiworld_double $*$ **p20**
- epiworld_double $*$ **p21**
- epiworld_double $*$ **p22**
- epiworld_double $*$ **p23**
- epiworld_double $*$ **p24**
- epiworld_double $*$ **p25**
- epiworld_double $*$ **p26**
- epiworld_double $*$ **p27**
- epiworld_double $*$ **p28**
- epiworld_double $*$ **p29**
- epiworld_double $*$ **p30**
- epiworld_double $*$ **p31**
- epiworld_double $*$ **p32**
- epiworld_double $*$ **p33**
- epiworld_double $*$ **p34**
- epiworld_double $*$ **p35**
- epiworld_double $*$ **p36**
- epiworld_double $*$ **p37**
- epiworld_double $*$ **p38**
- epiworld_double $*$ **p39**
- unsigned int **npar_used** = 0u
- epiworld_double **add_param** (epiworld_double initial_val, std::string pname)
- void **read_params** (std::string fn)
- epiworld_double **get_param** (unsigned int k)
- epiworld_double **get_param** (std::string pname)
- epiworld_double **par** (unsigned int k)
- epiworld_double **par** (std::string pname)

13.19.1   Detailed Description

template$<$**typename TSeq**$>$
class epiworld::Model$<$ **TSeq** $>$

Core class of epiworld.

The model class provides the wrapper that puts together Agent, Virus, and Tools.

Template Parameters

TSeq	Type of sequence. In principle, users can build models in which virus and human sequence is represented as numeric vectors (if needed.)

13.19.2   Member Function Documentation

13.19.2.1   add_global_action()

```
template<typename TSeq >
void Model< TSeq >::add_global_action (
```

```
        std::function< void(Model< TSeq > *)> fun,
        int date = -99 )  [inline]
```

Set a global action.

Parameters

fun	A function to be called on the prescribed dates
date	Integer indicating when the function is called (see details)

When date is less than zero, then the function is called at the end of every day. Otherwise, the function will be called only at the end of the indicated date.

13.19.2.2 reset()

```
template<typename TSeq >
void Model< TSeq >::reset  [inline]
```

Reset the model.

Resetting the model will:

- clear the database

- restore the population (if `set_backup()` was called before)

- re-distribute tools

- re-distribute viruses

- set the date to 0

13.19.2.3 run_multiple()

```
template<typename TSeq >
void Model< TSeq >::run_multiple (
        unsigned int nexperiments,
        std::function< void(size_t, Model< TSeq > *)> fun = make_save_run<TSeq>(),
        bool reset = true,
        bool verbose = true )  [inline]
```

Parameters

nexperiments	Multiple runs of the simulation

13.19.2.4 set_agents_data()

```
template<typename TSeq >
void Model< TSeq >::set_agents_data (
            double * data_,
            size_t ncols_ )  [inline]
```

Set the agents data object.

The data should be an array with the data stored in a column major order, i.e., by column.

Parameters

data↩	Pointer to the first element of an array of size `size() *`
_	`ncols_.`
ncols↩	Number of features included in the data.
_	

13.19.2.5 set_name()

```
template<typename TSeq >
void Model< TSeq >::set_name (
            std::string name )  [inline]
```

Set the name object.

Parameters

name	

13.19.2.6 write_data()

```
template<typename TSeq >
void Model< TSeq >::write_data (
            std::string fn_variant_info,
            std::string fn_variant_hist,
            std::string fn_tool_info,
            std::string fn_tool_hist,
            std::string fn_total_hist,
            std::string fn_transmission,
            std::string fn_transition,
            std::string fn_reproductive_number ) const  [inline]
```

Wrapper of `DataBase::write_data`

Parameters

fn_variant_info	Filename. Information about the variant.
fn_variant_hist	Filename. History of the variant.
fn_tool_info	Filename. Information about the tool.
fn_tool_hist	Filename. History of the tool.
fn_total_hist	Filename. Aggregated history (status)
fn_transmission	Filename. Transmission history.
fn_transition	Filename. Markov transition history.
fn_reproductive_number	Filename. Case by case reproductive number

The documentation for this class was generated from the following file:

- epiworld.hpp

13.20 Model< TSeq > Class Template Reference

Core class of epiworld.

```
#include <model-bones.hpp>
```

Public Member Functions

- DataBase< TSeq > & **get_db** ()
- epiworld_double & **operator()** (std::string pname)
- size_t **size** () const
- size_t **get_n_variants** () const
- size_t **get_n_tools** () const
- unsigned int **get_ndays** () const
- unsigned int **get_n_replicates** () const
- void **set_ndays** (unsigned int ndays)
- bool **get_verbose** () const
- void **verbose_off** ()
- void **verbose_on** ()
- int today () const

    The current time of the model.
- void write_data (std::string fn_variant_info, std::string fn_variant_hist, std::string fn_tool_info, std::string fn↵_tool_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition, std::string fn_↵reproductive_number) const

    Wrapper of `DataBase::write_data`
- std::map< std::string, epiworld_double > & **params** ()
- void reset ()

    Reset the model.
- void **print** () const
- Model< TSeq > && **clone** () const
- void **get_elapsed** (std::string unit="auto", epiworld_double ∗last_elapsed=nullptr, epiworld_double ∗total_↵elapsed=nullptr, std::string ∗unit_abbr=nullptr, bool print=true) const
- void add_global_action (std::function< void(Model< TSeq > ∗)> fun, int date=-99)

    Set a global action.

- • void **run_global_actions** ()
- • void **clear_status_set** ()
- • const std::vector< VirusPtr< TSeq > > & **get_viruses** () const
- • const std::vector< ToolPtr< TSeq > > & **get_tools** () const
- • void set_agents_data (double ∗data_, size_t ncols_)
    Set the agents data object.
- • void set_name (std::string name)
    Set the name object.

Set the backup object

backup can be used to restore the entire object after a run. This can be useful if the user wishes to have individuals start with the same network from the beginning.

- • void **set_backup** ()
- • void **restore_backup** ()

Random number generation

Parameters

eng	*Random number generator*
s	*Seed*

- • void **set_rand_engine** (std::mt19937 &eng)
- • std::mt19937 ∗ **get_rand_endgine** ()
- • void **seed** (unsigned int s)
- • void **set_rand_norm** (epiworld_double mean, epiworld_double sd)
- • void **set_rand_unif** (epiworld_double a, epiworld_double b)
- • void **set_rand_exp** (epiworld_double lambda)
- • void **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- • void **set_rand_lognormal** (epiworld_double mean, epiworld_double shape)
- • epiworld_double **runif** ()
- • epiworld_double **runif** (epiworld_double a, epiworld_double b)
- • epiworld_double **rnorm** ()
- • epiworld_double **rnorm** (epiworld_double mean, epiworld_double sd)
- • epiworld_double **rgamma** ()
- • epiworld_double **rgamma** (epiworld_double alpha, epiworld_double beta)
- • epiworld_double **rexp** ()
- • epiworld_double **rexp** (epiworld_double lambda)
- • epiworld_double **rlognormal** ()
- • epiworld_double **rlognormal** (epiworld_double mean, epiworld_double shape)

Add Virus/Tool to the model

This is done before the model has been initialized.

Parameters

v	*Virus to be added*
t	*Tool to be added*
preval	*Initial prevalence (initial state.) It can be specified as a proportion (between zero and one,) or an integer indicating number of individuals.*

- • void **add_virus** (Virus< TSeq > v, epiworld_double preval)
- • void **add_virus_n** (Virus< TSeq > v, unsigned int preval)

- void **add_virus_fun** (Virus< TSeq > v, VirusToAgentFun< TSeq > fun)
- void **add_tool** (Tool< TSeq > t, epiworld_double preval)
- void **add_tool_n** (Tool< TSeq > t, unsigned int preval)
- void **add_tool_fun** (Tool< TSeq > t, ToolToAgentFun< TSeq > fun)
- void **add_entity** (Entity< TSeq > e, epiworld_double preval)
- void **add_entity_n** (Entity< TSeq > e, unsigned int preval)
- void **add_entity_fun** (Entity< TSeq > e, EntityToAgentFun< TSeq > fun)

Accessing population of the model

Parameters

fn	*std::string Filename of the edgelist file.*
skip	*int Number of lines to skip in `fn`.*
directed	*bool Whether the graph is directed or not.*
size	*Size of the network.*
al	*AdjList to read into the model.*

- void **agents_from_adjlist** (std::string fn, int size, int skip=0, bool directed=false)
- void **agents_from_adjlist** (AdjList al)
- bool **is_directed** () const
- std::vector< Agent< TSeq > > ∗ **get_agents** ()
- void **agents_smallworld** (unsigned int n=1000, unsigned int k=5, bool d=false, epiworld_double p=.01)
- void **agents_empty_graph** (unsigned int n=1000)

Functions to run the model

Parameters

seed	*Seed to be used for Pseudo-RNG.*
ndays	*Number of days (steps) of the simulation.*
fun	*In the case of `run_multiple`, a function that is called after each experiment.*

- void **init** (unsigned int ndays, unsigned int seed)
- void **update_status** ()
- void **mutate_variant** ()
- void **next** ()
- void [run](run) ()

    Runs the simulation (after initialization)
- void run_multiple (unsigned int nexperiments, std::function< void(size_t, Model< TSeq > ∗)> fun=make_save_run< TSeq >(), bool reset=true, bool verbose=true)

Rewire the network preserving the degree sequence.

This implementation assumes an undirected network, thus if {(i,j), (k,l)} -> {(i,l), (k,j)}, the reciprocal is also true, i.e., {(j,i), (l,k)} -> {(j,k), (l,i)}.

Parameters

proportion	*Proportion of ties to be rewired.*

Returns

> *A rewired version of the network.*

- void **set_rewire_fun** (std::function< void(std::vector< Agent< TSeq >> *, Model< TSeq > *, epiworld_double)> fun)
- void **set_rewire_prop** (epiworld_double prop)
- epiworld_double **get_rewire_prop** () const
- void **rewire** ()

Export the network data in edgelist form

Parameters

fn	*std::string. File name.*
source	*Integer vector*
target	*Integer vector*

When passing the source and target, the function will write the edgelist on those.

- void **write_edgelist** (std::string fn) const
- void **write_edgelist** (std::vector< unsigned int > &source, std::vector< unsigned int > &target) const

Manage status (states) in the model

The functions `get_status` *return the current values for the statuses included in the model.*

Parameters

lab	`std::string` *Name of the status.*

Returns

> `add_status*` *returns nothing.*
>
> `get_status_*` *returns a vector of pairs with the statuses and their labels.*

- void **add_status** (std::string lab, UpdateFun< TSeq > fun=nullptr)
- const std::vector< std::string > & **get_status** () const
- const std::vector< UpdateFun< TSeq > > & **get_status_fun** () const
- void **print_status_codes** () const

Set the user data object

Parameters

names	*string vector with the names of the variables.*

- void set_user_data (std::vector< std::string > names)
    [@
- void **add_user_data** (unsigned int j, epiworld_double x)
- void **add_user_data** (std::vector< epiworld_double > x)
- UserData< TSeq > & **get_user_data** ()

Queuing system

When queueing is on, the model will keep track of which agents are either in risk of exposure or exposed. This then is used at each step to act only on the aforementioned agents.

- void queuing_on ()

    Activates the queuing system (default.)
- void queuing_off ()

    Deactivates the queuing system.
- bool is_queuing_on () const

    Query if the queuing system is on.
- Queue< TSeq > & get_queue ()

    Retrieve the Queue object.

Get the susceptibility reduction object

Parameters

v	

Returns

   epiworld_double

- void **set_susceptibility_reduction_mixer** (MixerFun< TSeq > fun)
- void **set_transmission_reduction_mixer** (MixerFun< TSeq > fun)
- void **set_recovery_enhancer_mixer** (MixerFun< TSeq > fun)
- void **set_death_reduction_mixer** (MixerFun< TSeq > fun)

Friends

- class **Agent**< **TSeq** >
- class **AgentsSample**< **TSeq** >
- class **DataBase**< **TSeq** >
- class **Queue**< **TSeq** >

Tool Mixers

These functions combine the effects tools have to deliver a single effect. For example, wearing a mask, been vaccinated, and the immune system combine together to jointly reduce the susceptibility for a given virus.

- std::vector< epiworld_double > **array_double_tmp**
- std::vector< Virus< TSeq > ∗ > **array_virus_tmp**
- **Model** ()
- **Model** (const Model< TSeq > &m)
- **Model** (Model< TSeq > &&m)
- Model< TSeq > & **operator=** (const Model< TSeq > &m)
- void **clone_population** (std::vector< Agent< TSeq > > &p, bool &d, Model< TSeq > ∗m=nullptr) const
- void **clone_population** (const Model< TSeq > &m)

Setting and accessing parameters from the model

Tools can incorporate parameters included in the model. Internally, parameters in the tool are stored as pointers to an std::map<> of parameters in the model. Using the `unsigned int` method directly fetches the parameters in the order these were added to the tool. Accessing parameters via the `std::string` method involves searching the parameter directly in the std::map<> member of the model (so it is not recommended.)

The `par()` function members are aliases for `get_param()`.

In the case of the function `read_params`, users can pass a file listing parameters to be included in the model. Each line in the file should have the following structure:

```
[name of parameter 1]: [value in double]
[name of parameter 2]: [value in double]
...
```

The only condition for parameter names is that these do not include a colon.

Parameters

initial_val	
pname	Name of the parameter to add or to fetch
fn	Path to the file containing parameters

Returns

> The current value of the parameter in the model.

- epiworld_double ∗ **p0**
- epiworld_double ∗ **p1**
- epiworld_double ∗ **p2**
- epiworld_double ∗ **p3**
- epiworld_double ∗ **p4**
- epiworld_double ∗ **p5**
- epiworld_double ∗ **p6**
- epiworld_double ∗ **p7**
- epiworld_double ∗ **p8**
- epiworld_double ∗ **p9**
- epiworld_double ∗ **p10**
- epiworld_double ∗ **p11**
- epiworld_double ∗ **p12**
- epiworld_double ∗ **p13**
- epiworld_double ∗ **p14**
- epiworld_double ∗ **p15**
- epiworld_double ∗ **p16**
- epiworld_double ∗ **p17**
- epiworld_double ∗ **p18**
- epiworld_double ∗ **p19**
- epiworld_double ∗ **p20**
- epiworld_double ∗ **p21**
- epiworld_double ∗ **p22**
- epiworld_double ∗ **p23**
- epiworld_double ∗ **p24**
- epiworld_double ∗ **p25**
- epiworld_double ∗ **p26**

- epiworld_double ∗ **p27**
- epiworld_double ∗ **p28**
- epiworld_double ∗ **p29**
- epiworld_double ∗ **p30**
- epiworld_double ∗ **p31**
- epiworld_double ∗ **p32**
- epiworld_double ∗ **p33**
- epiworld_double ∗ **p34**
- epiworld_double ∗ **p35**
- epiworld_double ∗ **p36**
- epiworld_double ∗ **p37**
- epiworld_double ∗ **p38**
- epiworld_double ∗ **p39**
- unsigned int **npar_used** = 0u
- epiworld_double **add_param** (epiworld_double initial_val, std::string pname)
- void **read_params** (std::string fn)
- epiworld_double **get_param** (unsigned int k)
- epiworld_double **get_param** (std::string pname)
- epiworld_double **par** (unsigned int k)
- epiworld_double **par** (std::string pname)

13.20.1 Detailed Description

template<**typename TSeq**>
class Model< **TSeq** >

Core class of epiworld.

The model class provides the wrapper that puts together Agent, Virus, and Tools.

Template Parameters

TSeq	Type of sequence. In principle, users can build models in which virus and human sequence is represented as numeric vectors (if needed.)

13.20.2 Member Function Documentation

13.20.2.1 add_global_action()

```
template<typename TSeq >
void Model< TSeq >::add_global_action (
            std::function< void(Model< TSeq > *)> fun,
            int date = -99 )  [inline]
```

Set a global action.

Parameters

fun	A function to be called on the prescribed dates
date	Integer indicating when the function is called (see details)

When date is less than zero, then the function is called at the end of every day. Otherwise, the function will be called only at the end of the indicated date.

13.20.2.2 reset()

```
template<typename TSeq >
void Model< TSeq >::reset  [inline]
```

Reset the model.

Resetting the model will:

- clear the database

- restore the population (if `set_backup()` was called before)

- re-distribute tools

- re-distribute viruses

- set the date to 0

13.20.2.3 run_multiple()

```
template<typename TSeq >
void Model< TSeq >::run_multiple (
            unsigned int nexperiments,
            std::function< void(size_t, Model< TSeq > *)> fun = make_save_run<TSeq>(),
            bool reset = true,
            bool verbose = true )  [inline]
```

Parameters

nexperiments	Multiple runs of the simulation

13.20.2.4 set_agents_data()

```
template<typename TSeq >
void Model< TSeq >::set_agents_data (
            double * data_,
            size_t ncols_ )  [inline]
```

Set the agents data object.

The data should be an array with the data stored in a column major order, i.e., by column.

Parameters

data↩_	Pointer to the first element of an array of size `size() *` `ncols_`.
ncols↩_	Number of features included in the data.

13.20.2.5 set_name()

```
template<typename TSeq >
void Model< TSeq >::set_name (
            std::string name )  [inline]
```

Set the name object.

Parameters

name	

13.20.2.6 write_data()

```
template<typename TSeq >
void Model< TSeq >::write_data (
            std::string fn_variant_info,
            std::string fn_variant_hist,
            std::string fn_tool_info,
            std::string fn_tool_hist,
            std::string fn_total_hist,
            std::string fn_transmission,
            std::string fn_transition,
            std::string fn_reproductive_number ) const  [inline]
```

Wrapper of `DataBase::write_data`

Parameters

fn_variant_info	Filename. Information about the variant.
fn_variant_hist	Filename. History of the variant.
fn_tool_info	Filename. Information about the tool.
fn_tool_hist	Filename. History of the tool.
fn_total_hist	Filename. Aggregated history (status)
fn_transmission	Filename. Transmission history.
fn_transition	Filename. Markov transition history.
fn_reproductive_number	Filename. Case by case reproductive number

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/model-bones.hpp
- include/epiworld/model-meat-print.hpp
- include/epiworld/model-meat.hpp

13.21 epiworld::epimodels::ModelSEIR< TSeq > Class Template Reference

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for epiworld::epimodels::ModelSEIR< TSeq >:

Collaboration diagram for epiworld::epimodels::ModelSEIR< TSeq >:
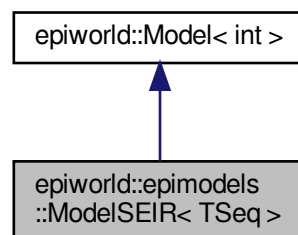
Public Member Functions

- **ModelSEIR** (ModelSEIR< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_↩ double infectiousness, epiworld_double incubation_days, epiworld_double recovery)
- **ModelSEIR** (std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_↩ double incubation_days, epiworld_double recovery)

Public Attributes

- epiworld::UpdateFun< TSeq > **update_exposed_seir**
- epiworld::UpdateFun< TSeq > **update_infected_seir**

13.21.1 Detailed Description

template<**typename TSeq = int**>
class epiworld::epimodels::ModelSEIR< **TSeq** >

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery rate of the immune system

13.21.2 Member Data Documentation

13.21.2.1 update_exposed_seir

```
template<typename TSeq = int>
epiworld::UpdateFun<TSeq> epiworld::epimodels::ModelSEIR< TSeq >::update_exposed_seir
```

Initial value:
```
= [](
        epiworld::Agent<TSeq> * p,
        epiworld::Model<TSeq> * m
    ) -> void {

        if (m->runif() < 1.0/(*m->p1))
            p->change_status(ModelSEIR<TSeq>::INFECTED);
        return;
    }
```

13.21.2.2 update_infected_seir

```
template<typename TSeq = int>
epiworld::UpdateFun<TSeq> epiworld::epimodels::ModelSEIR< TSeq >::update_infected_seir
```

Initial value:
```
= [](
        epiworld::Agent<TSeq> * p,
        epiworld::Model<TSeq> * m
    ) -> void {

        if (m->runif() < (*m->p2))
            p->rm_virus(0);
        return;
    }
```

The documentation for this class was generated from the following file:

- epiworld.hpp

13.22 ModelSEIR< TSeq > Class Template Reference

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

```
#include <seir.hpp>
```

Inheritance diagram for ModelSEIR< TSeq >:

Collaboration diagram for ModelSEIR< TSeq >:

Public Member Functions

- **ModelSEIR** (ModelSEIR< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_↩ double infectiousness, epiworld_double incubation_days, epiworld_double recovery)
- **ModelSEIR** (std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_↩ double incubation_days, epiworld_double recovery)

Public Attributes

- epiworld::UpdateFun< TSeq > **update_exposed_seir**
- epiworld::UpdateFun< TSeq > **update_infected_seir**

13.22.1 Detailed Description

template<**typename TSeq = int**>
class ModelSEIR< TSeq >

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery rate of the immune system

13.22.2 Member Data Documentation

13.22.2.1 update_exposed_seir

```
template<typename TSeq = int>
epiworld::UpdateFun<TSeq> ModelSEIR< TSeq >::update_exposed_seir
```

Initial value:
```
= [](
        epiworld::Agent<TSeq> * p,
        epiworld::Model<TSeq> * m
    ) -> void {

        if (m->runif() < 1.0/(*m->p1))
            p->change_status(ModelSEIR<TSeq>::INFECTED);
        return;
    }
```

13.22.2.2 update_infected_seir

```
template<typename TSeq = int>
epiworld::UpdateFun<TSeq> ModelSEIR< TSeq >::update_infected_seir
```

Initial value:
```
= [](
        epiworld::Agent<TSeq> * p,
        epiworld::Model<TSeq> * m
    ) -> void {

        if (m->runif() < (*m->p2))
            p->rm_virus(0);
        return;
    }
```
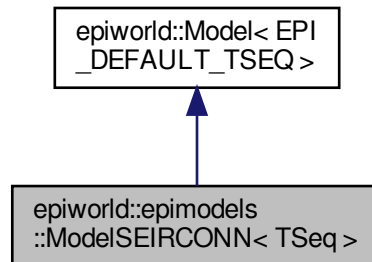
The documentation for this class was generated from the following file:

- include/epiworld/models/seir.hpp

13.23 epiworld::epimodels::ModelSEIRCONN< TSeq > Class Template Reference

Inheritance diagram for epiworld::epimodels::ModelSEIRCONN< TSeq >:

Collaboration diagram for epiworld::epimodels::ModelSEIRCONN< TSeq >:

Public Member Functions

- ModelSEIRCONN (ModelSEIRCONN< TSeq > &model, std::string vname, unsigned int n, epiworld_↩
  double prevalence, epiworld_double reproductive_number, epiworld_double prob_transmission, epiworld↩
  _double incubation_days, epiworld_double prob_recovery)

    Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRCONN** (std::string vname, unsigned int n, epiworld_double prevalence, epiworld_double
  reproductive_number, epiworld_double prob_transmission, epiworld_double incubation_days, epiworld↩
  _double prob_recovery)

Public Attributes

- std::vector< epiworld::Agent<> * > **tracked_agents_infected** = {}
- std::vector< epiworld::Agent<> * > **tracked_agents_infected_next** = {}
- bool **tracked_started** = false
- int **tracked_ninfected** = 0
- int **tracked_ninfected_next** = 0

13.23.1 Constructor & Destructor Documentation

13.23.1.1 ModelSEIRCONN()

```
template<typename TSeq >
ModelSEIRCONN< TSeq >::ModelSEIRCONN (
            ModelSEIRCONN< TSeq > & model,
            std::string vname,
            unsigned int n,
            epiworld_double prevalence,
            epiworld_double reproductive_number,
            epiworld_double prob_transmission,
            epiworld_double incubation_days,
            epiworld_double prob_recovery )  [inline]
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
prevalence	Initial prevalence (proportion)
reproductive_number	Reproductive number (beta)
prob_transmission	Probability of transmission
prob_recovery	Probability of recovery

The documentation for this class was generated from the following file:

- epiworld.hpp

13.24 ModelSEIRCONN< TSeq > Class Template Reference

Inheritance diagram for ModelSEIRCONN< TSeq >:

Collaboration diagram for ModelSEIRCONN< TSeq >:

Public Member Functions

- ModelSEIRCONN (ModelSEIRCONN< TSeq > &model, std::string vname, unsigned int n, epiworld_↩
  double prevalence, epiworld_double reproductive_number, epiworld_double prob_transmission, epiworld↩
  _double incubation_days, epiworld_double prob_recovery)

    Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

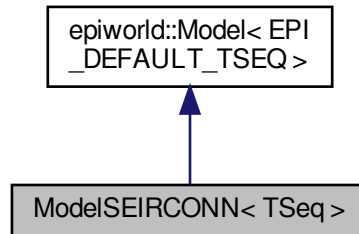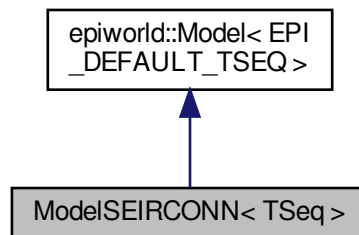- **ModelSEIRCONN** (std::string vname, unsigned int n, epiworld_double prevalence, epiworld_double
  reproductive_number, epiworld_double prob_transmission, epiworld_double incubation_days, epiworld↩
  _double prob_recovery)

Public Attributes

- std::vector< epiworld::Agent<> * > **tracked_agents_infected** = {}
- std::vector< epiworld::Agent<> * > **tracked_agents_infected_next** = {}
- bool **tracked_started** = false
- int **tracked_ninfected** = 0
- int **tracked_ninfected_next** = 0

13.24.1 Constructor & Destructor Documentation

13.24.1.1 ModelSEIRCONN()

```
template<typename TSeq >
ModelSEIRCONN< TSeq >::ModelSEIRCONN (
            ModelSEIRCONN< TSeq > & model,
            std::string vname,
            unsigned int n,
            epiworld_double prevalence,
            epiworld_double reproductive_number,
            epiworld_double prob_transmission,
            epiworld_double incubation_days,
            epiworld_double prob_recovery ) [inline]
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
prevalence	Initial prevalence (proportion)
reproductive_number	Reproductive number (beta)
prob_transmission	Probability of transmission
prob_recovery	Probability of recovery

The documentation for this class was generated from the following file:

- include/epiworld/models/seirconnected.hpp

13.25 ModelSEIRCONNLogit< TSeq > Class Template Reference

Inheritance diagram for ModelSEIRCONNLogit< TSeq >:
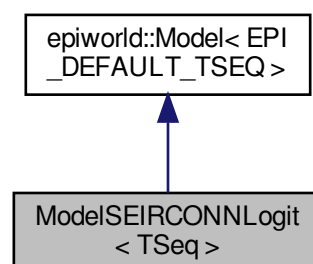
Collaboration diagram for ModelSEIRCONNLogit< TSeq >:
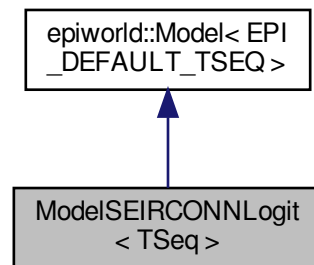
Public Member Functions

- ModelSEIRCONNLogit (ModelSEIRCONNLogit< TSeq > &model, std::string vname, unsigned int n, epiworld_double prevalence, epiworld_double reproductive_number, epiworld_double prob_transmission, epiworld_double incubation_days, epiworld_double prob_recovery, double ∗covars, std::vector< double > logit_params)

  Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

- **ModelSEIRCONNLogit** (std::string vname, unsigned int n, epiworld_double prevalence, epiworld_double reproductive_number, epiworld_double prob_transmission, epiworld_double incubation_days, epiworld_↩ double prob_recovery double ∗covars, std::vector< double > logit_params)

Public Attributes

- std::vector< epiworld::Agent<> ∗ > **tracked_agents_infected** = {}
- std::vector< epiworld::Agent<> ∗ > **tracked_agents_infected_next** = {}
- bool **tracked_started** = false
- int **tracked_ninfected** = 0
- int **tracked_ninfected_next** = 0

13.25.1 Constructor & Destructor Documentation

13.25.1.1 ModelSEIRCONNLogit()

```
template<typename TSeq >
ModelSEIRCONNLogit< TSeq >::ModelSEIRCONNLogit (
            ModelSEIRCONNLogit< TSeq > & model,
            std::string vname,
            unsigned int n,
            epiworld_double prevalence,
            epiworld_double reproductive_number,
```

```
        epiworld_double prob_transmission,
        epiworld_double incubation_days,
        epiworld_double prob_recovery,
        double * covars,
        std::vector< double > logit_params )  [inline]
```

Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
prevalence	Initial prevalence (proportion)
reproductive_number	Reproductive number (beta)
prob_transmission	Probability of transmission
prob_recovery	Probability of recovery

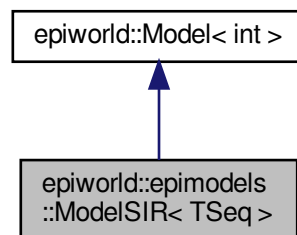The documentation for this class was generated from the following file:

- include/epiworld/models/seirconnected_logit.hpp

13.26 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference
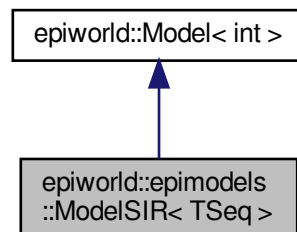
Template for a Susceptible-Infected-Removed (SIR) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for epiworld::epimodels::ModelSIR< TSeq >:

Collaboration diagram for epiworld::epimodels::ModelSIR< TSeq >:

Public Member Functions

- **ModelSIR** (ModelSIR< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_double recovery)
- **ModelSIR** (std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_double recovery)

Additional Inherited Members

13.26.1 Detailed Description

template<**typename TSeq = int**>
class epiworld::epimodels::ModelSIR< **TSeq** >

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery rate of the immune system

The documentation for this class was generated from the following file:
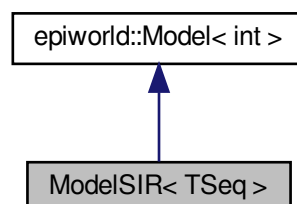
- epiworld.hpp

13.27 ModelSIR< TSeq > Class Template Reference

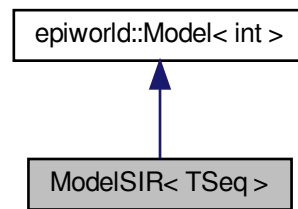Template for a Susceptible-Infected-Removed (SIR) model.

```
#include <sir.hpp>
```

Inheritance diagram for ModelSIR< TSeq >:

Collaboration diagram for ModelSIR< TSeq >:

Public Member Functions

- **ModelSIR** (ModelSIR< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_double recovery)
- **ModelSIR** (std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_double recovery)

Additional Inherited Members

13.27.1   Detailed Description

template<**typename TSeq = int**>
class ModelSIR< **TSeq** >

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery rate of the immune system

The documentation for this class was generated from the following file:

- include/epiworld/models/sir.hpp

13.28 epiworld::epimodels::ModelSIRCONN< TSeq > Class Template Reference

Inheritance diagram for epiworld::epimodels::ModelSIRCONN< TSeq >:

Collaboration diagram for epiworld::epimodels::ModelSIRCONN< TSeq >:

Public Member Functions

- ModelSIRCONN (ModelSIRCONN< TSeq > &model, std::string vname, unsigned int n, epiworld_double prevalence, epiworld_double reproductive_number, epiworld_double prob_transmission, epiworld_double prob_recovery)

    Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRCONN** (std::string vname, unsigned int n, epiworld_double prevalence, epiworld_double reproductive_number, epiworld_double prob_transmission, epiworld_double prob_recovery)

Public Attributes

- std::vector< [epiworld::Agent](TSeq)< TSeq > ∗ > **tracked_agents_infected** = {}
- std::vector< [epiworld::Agent](TSeq)< TSeq > ∗ > **tracked_agents_infected_next** = {}
- bool **tracked_started** = false
- int **tracked_ninfected** = 0
- int **tracked_ninfected_next** = 0
- epiworld_double **tracked_current_infect_prob** = 0.0

13.28.1 Constructor & Destructor Documentation

13.28.1.1 ModelSIRCONN()

```
template<typename TSeq >
ModelSIRCONN< TSeq >::ModelSIRCONN (
            ModelSIRCONN< TSeq > & model,
            std::string vname,
            unsigned int n,
            epiworld_double prevalence,
            epiworld_double reproductive_number,
            epiworld_double prob_transmission,
            epiworld_double prob_recovery )  [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
prevalence	Initial prevalence (proportion)
reproductive_number	Reproductive number (beta)
prob_transmission	Probability of transmission
prob_recovery	Probability of recovery

The documentation for this class was generated from the following file:

- epiworld.hpp

13.29 ModelSIRCONN< TSeq > Class Template Reference

Inheritance diagram for ModelSIRCONN< TSeq >:

Collaboration diagram for ModelSIRCONN< TSeq >:

Public Member Functions

- ModelSIRCONN (ModelSIRCONN< TSeq > &model, std::string vname, unsigned int n, epiworld_double prevalence, epiworld_double reproductive_number, epiworld_double prob_transmission, epiworld_double prob_recovery)

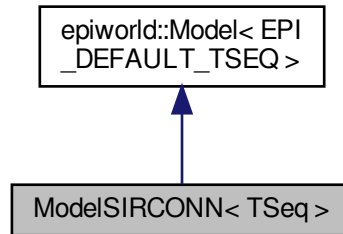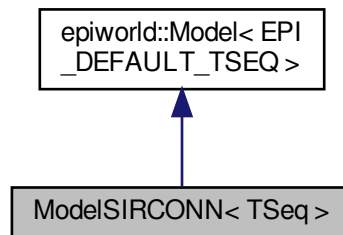   *Template for a Susceptible-Infected-Removed (SIR) model.*

- **ModelSIRCONN** (std::string vname, unsigned int n, epiworld_double prevalence, epiworld_double reproductive_number, epiworld_double prob_transmission, epiworld_double prob_recovery)

Public Attributes

- std::vector< epiworld::Agent< TSeq > ∗ > **tracked_agents_infected** = {}
- std::vector< epiworld::Agent< TSeq > ∗ > **tracked_agents_infected_next** = {}
- bool **tracked_started** = false
- int **tracked_ninfected** = 0
- int **tracked_ninfected_next** = 0
- epiworld_double **tracked_current_infect_prob** = 0.0

13.29.1 Constructor & Destructor Documentation

13.29.1.1 ModelSIRCONN()

```
template<typename TSeq >
ModelSIRCONN< TSeq >::ModelSIRCONN (
            ModelSIRCONN< TSeq > & model,
            std::string vname,
            unsigned int n,
            epiworld_double prevalence,
            epiworld_double reproductive_number,
            epiworld_double prob_transmission,
            epiworld_double prob_recovery )  [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
prevalence	Initial prevalence (proportion)
reproductive_number	Reproductive number (beta)
prob_transmission	Probability of transmission
prob_recovery	Probability of recovery

The documentation for this class was generated from the following file:

- include/epiworld/models/sirconnected.hpp

13.30 epiworld::epimodels::ModelSIS< TSeq > Class Template Reference
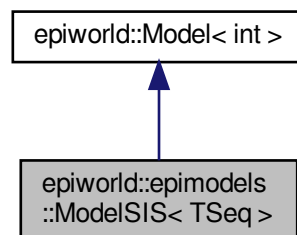
Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#include <epiworld.hpp>
```

Inheritance diagram for epiworld::epimodels::ModelSIS< TSeq >:

Collaboration diagram for epiworld::epimodels::ModelSIS< TSeq >:

Public Member Functions

- **ModelSIS** (ModelSIS< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_double recovery)
- **ModelSIS** (std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_double recovery)

Additional Inherited Members

13.30.1 Detailed Description

template<**typename TSeq = int**>
class epiworld::epimodels::ModelSIS< **TSeq** >

Template for a Susceptible-Infected-Susceptible (SIS) model.

Parameters

vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery rate of the immune system

The documentation for this class was generated from the following file:

- epiworld.hpp

13.31 ModelSIS< TSeq > Class Template Reference

Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#include <sis.hpp>
```

Inheritance diagram for ModelSIS< TSeq >:

Collaboration diagram for ModelSIS< TSeq >:

Public Member Functions

- **ModelSIS** (ModelSIS< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_double recovery)
- **ModelSIS** (std::string vname, epiworld_double prevalence, epiworld_double infectiousness, epiworld_double recovery)

Additional Inherited Members

13.31.1 Detailed Description

template<**typename TSeq = int**>
class ModelSIS< **TSeq** >

Template for a Susceptible-Infected-Susceptible (SIS) model.
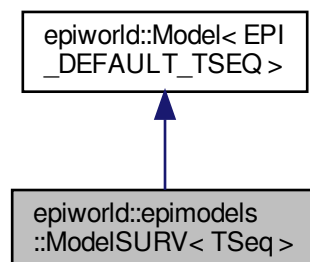
Parameters

vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery rate of the immune system

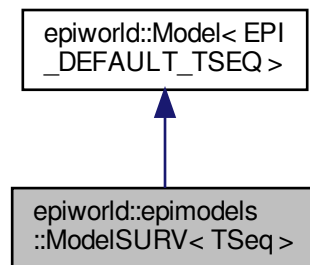The documentation for this class was generated from the following file:

- include/epiworld/models/sis.hpp

13.32 epiworld::epimodels::ModelSURV< TSeq > Class Template Reference

Inheritance diagram for epiworld::epimodels::ModelSURV< TSeq >:

Collaboration diagram for epiworld::epimodels::ModelSURV< TSeq >:

Public Member Functions

- ModelSURV (ModelSURV< TSeq > &model, std::string vname, unsigned int prevalence=50, epiworld_←
  double efficacy_vax=0.9, epiworld_double latent_period=3u, epiworld_double infect_period=6u, epiworld←
  _double prob_symptoms=0.6, epiworld_double prop_vaccinated=0.25, epiworld_double prop_vax_redux←
  _transm=0.5, epiworld_double prop_vax_redux_infect=0.5, epiworld_double surveillance_prob=0.001,
  epiworld_double prob_transmission=1.0, epiworld_double prob_death=0.001, epiworld_double prob_←
  noreinfect=0.9)

  Template for a Susceptible-Infected-Removed (SIR) model.

- **ModelSURV** (std::string vname, unsigned int prevalence=50, epiworld_double efficacy_vax=0.9, epiworld←
  _double latent_period=3u, epiworld_double infect_period=6u, epiworld_double prob_symptoms=0.←
  6, epiworld_double prop_vaccinated=0.25, epiworld_double prop_vax_redux_transm=0.5, epiworld_←
  double prop_vax_redux_infect=0.5, epiworld_double surveillance_prob=0.001, epiworld_double prob_←
  transmission=1.0, epiworld_double prob_death=0.001, epiworld_double prob_noreinfect=0.9)

Additional Inherited Members

13.32.1 Constructor & Destructor Documentation

13.32.1.1 ModelSURV()

```
template<typename TSeq >
ModelSURV< TSeq >::ModelSURV (
            ModelSURV< TSeq > & model,
            std::string vname,
            unsigned int prevalence = 50,
            epiworld_double efficacy_vax = 0.9,
            epiworld_double latent_period = 3u,
            epiworld_double infect_period = 6u,
            epiworld_double prob_symptoms = 0.6,
            epiworld_double prop_vaccinated = 0.25,
```

```
        epiworld_double prop_vax_redux_transm = 0.5,
        epiworld_double prop_vax_redux_infect = 0.5,
        epiworld_double surveillance_prob = 0.001,
        epiworld_double prob_transmission = 1.0,
        epiworld_double prob_death = 0.001,
        epiworld_double prob_noreinfect = 0.9 )  [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

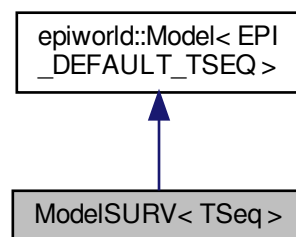**Parameters**

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_susceptibility_reduction	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery rate of the immune system

The documentation for this class was generated from the following file:
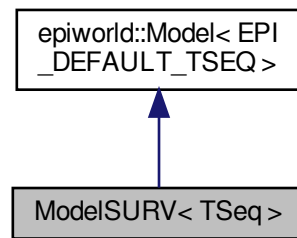
- epiworld.hpp

13.33 ModelSURV< TSeq > Class Template Reference

Inheritance diagram for ModelSURV< TSeq >:

Collaboration diagram for ModelSURV< TSeq >:

Public Member Functions

- **ModelSURV** (ModelSURV< TSeq > &model, std::string vname, unsigned int prevalence=50, epiworld_↩
  double efficacy_vax=0.9, epiworld_double latent_period=3u, epiworld_double infect_period=6u, epiworld↩
  _double prob_symptoms=0.6, epiworld_double prop_vaccinated=0.25, epiworld_double prop_vax_redux↩
  _transm=0.5, epiworld_double prop_vax_redux_infect=0.5, epiworld_double surveillance_prob=0.001,
  epiworld_double prob_transmission=1.0, epiworld_double prob_death=0.001, epiworld_double prob_↩
  noreinfect=0.9)

  Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSURV** (std::string vname, unsigned int prevalence=50, epiworld_double efficacy_vax=0.9, epiworld↩
  _double latent_period=3u, epiworld_double infect_period=6u, epiworld_double prob_symptoms=0.↩
  6, epiworld_double prop_vaccinated=0.25, epiworld_double prop_vax_redux_transm=0.5, epiworld_↩
  double prop_vax_redux_infect=0.5, epiworld_double surveillance_prob=0.001, epiworld_double prob_↩
  transmission=1.0, epiworld_double prob_death=0.001, epiworld_double prob_noreinfect=0.9)

Additional Inherited Members

13.33.1   Constructor & Destructor Documentation

13.33.1.1   ModelSURV()

```
template<typename TSeq >
ModelSURV< TSeq >::ModelSURV (
            ModelSURV< TSeq > & model,
            std::string vname,
            unsigned int prevalence = 50,
            epiworld_double efficacy_vax = 0.9,
            epiworld_double latent_period = 3u,
            epiworld_double infect_period = 6u,
            epiworld_double prob_symptoms = 0.6,
            epiworld_double prop_vaccinated = 0.25,
            epiworld_double prop_vax_redux_transm = 0.5,
```

```
        epiworld_double prop_vax_redux_infect = 0.5,
        epiworld_double surveillance_prob = 0.001,
        epiworld_double prob_transmission = 1.0,
        epiworld_double prob_death = 0.001,
        epiworld_double prob_noreinfect = 0.9 )  [inline]
```

Template for a Susceptible-Infected-Removed (SIR) model.

Parameters

model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_susceptibility_reduction	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery rate of the immune system

The documentation for this class was generated from the following file:

- include/epiworld/models/surveillance.hpp

13.34 Network< Nettype, Nodetype, Edgetype > Class Template Reference

Public Member Functions

- **NType** ()
- Edgetype **operator()** (int i, int j)
- bool **is_directed** () const
- size_t **vcount** () const
- size_t **ecount** () const
- void **add_edge** (int i, int j)
- void **rm_edge** (int i, int j)

The documentation for this class was generated from the following file:

- include/epiworld/network-bones.hpp

13.35 epiworld::PersonTools< TSeq > Class Template Reference

The documentation for this class was generated from the following file:

- epiworld.hpp

13.36 PersonTools< TSeq > Class Template Reference

The documentation for this class was generated from the following file:

- include/epiworld/config.hpp

13.37 epiworld::Progress Class Reference

A simple progress bar.

```
#include <epiworld.hpp>
```

Public Member Functions

- **Progress** (int n_, int width_)
- void **start** ()
- void **next** ()
- void **end** ()

13.37.1 Detailed Description

A simple progress bar.

The documentation for this class was generated from the following file:

- epiworld.hpp

13.38 Progress Class Reference

A simple progress bar.

```
#include <progress.hpp>
```

Public Member Functions

- **Progress** (int n_, int width_)
- void **start** ()
- void **next** ()
- void **end** ()

13.38.1 Detailed Description

A simple progress bar.

The documentation for this class was generated from the following file:

- include/epiworld/progress.hpp

13.39 epiworld::Queue< TSeq > Class Template Reference

Controls which agents are verified at each step.

```
#include <epiworld.hpp>
```

Public Member Functions

- void **operator+=** (Agent< TSeq > ∗p)
- void **operator-=** (Agent< TSeq > ∗p)
- epiworld_fast_int & **operator[ ]** (unsigned int i)
- void **set_model** (Model< TSeq > ∗m)

13.39.1 Detailed Description

template<**typename TSeq**>
class epiworld::Queue< **TSeq** >

Controls which agents are verified at each step.

The idea is that only agents who are either in an infected state or have an infected neighbor should be checked. Otherwise it makes no sense (no chance to recover or capture the disease).

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- epiworld.hpp

13.40 Queue< TSeq > Class Template Reference

Controls which agents are verified at each step.

```
#include <queue-bones.hpp>
```

Public Member Functions

- void **operator+=** (Agent< TSeq > ∗p)
- void **operator-=** (Agent< TSeq > ∗p)
- epiworld_fast_int & **operator[ ]** (unsigned int i)
- void **set_model** (Model< TSeq > ∗m)

13.40.1 Detailed Description

template<**typename TSeq**>
class Queue< **TSeq** >

Controls which agents are verified at each step.

The idea is that only agents who are either in an infected state or have an infected neighbor should be checked. Otherwise it makes no sense (no chance to recover or capture the disease).

Template Parameters

TSeq	

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/queue-bones.hpp

13.41 epiworld::QueueValues Class Reference

Static Public Attributes

- static const int **NoOne** = 0
- static const int **OnlySelf** = 1
- static const int **Everyone** = 2

The documentation for this class was generated from the following file:

- epiworld.hpp

13.42 QueueValues Class Reference

Static Public Attributes

- static const int **NoOne** = 0
- static const int **OnlySelf** = 1
- static const int **Everyone** = 2

The documentation for this class was generated from the following file:

- include/epiworld/epiworld-macros.hpp

13.43 RandGraph Class Reference

Public Member Functions

- **RandGraph** (int N_)
- void **init** (int s)
- void **set_rand_engine** (std::mt19937 &e)
- epiworld_double **runif** ()

The documentation for this class was generated from the following file:

- include/epiworld/random_graph.hpp

13.44 epiworld::SAMPLETYPE Class Reference

Static Public Attributes

- static const int **MODEL** = 0
- static const int **ENTITY** = 1
- static const int **AGENT** = 2

The documentation for this class was generated from the following file:

- epiworld.hpp

13.45 SAMPLETYPE Class Reference

Static Public Attributes

- static const int **MODEL** = 0
- static const int **ENTITY** = 1
- static const int **AGENT** = 2

The documentation for this class was generated from the following file:

- include/epiworld/agentssample-bones.hpp

13.46 epiworld::Tool< TSeq > Class Template Reference

Tools for defending the agent against the virus.

```
#include <epiworld.hpp>
```

Public Member Functions

- **Tool** (std::string name="unknown tool")
- void **set_sequence** (TSeq d)
- void **set_sequence_unique** (TSeq d)
- void **set_sequence** (std::shared_ptr< TSeq > d)
- std::shared_ptr< TSeq > **get_sequence** ()
- TSeq & **get_sequence_unique** ()
- void **set_name** (std::string name)
- std::string **get_name** () const
- [Agent]< TSeq > ∗ **get_agent** ()
- int **get_id** () const
- void **set_id** (int id)
- void **set_date** (int d)
- int **get_date** () const
- void **set_status** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_status** (epiworld_fast_int ∗init, epiworld_fast_int ∗post)
- void **get_queue** (epiworld_fast_int ∗init, epiworld_fast_int ∗post)

Get and set the tool functions

Parameters

v	The virus over which to operate
fun	the function to be used

Returns

> *epiworld_double*

- epiworld_double **get_susceptibility_reduction** (VirusPtr< TSeq > v)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v)
- void **set_susceptibility_reduction_fun** (ToolFun< TSeq > fun)
- void **set_transmission_reduction_fun** (ToolFun< TSeq > fun)
- void **set_recovery_enhancer_fun** (ToolFun< TSeq > fun)
- void **set_death_reduction_fun** (ToolFun< TSeq > fun)
- void **set_susceptibility_reduction** (epiworld_double ∗prob)
- void **set_transmission_reduction** (epiworld_double ∗prob)
- void **set_recovery_enhancer** (epiworld_double ∗prob)
- void **set_death_reduction** (epiworld_double ∗prob)
- void **set_susceptibility_reduction** (epiworld_double prob)
- void **set_transmission_reduction** (epiworld_double prob)
- void **set_recovery_enhancer** (epiworld_double prob)
- void **set_death_reduction** (epiworld_double prob)

Friends

- class **Agent**< **TSeq** >
- class **Model**< **TSeq** >
- void **default_add_tool** ([Action]< TSeq > &a, [Model]< TSeq > ∗m)
- void **default_rm_tool** ([Action]< TSeq > &a, [Model]< TSeq > ∗m)

13.46.1 Detailed Description

template<**typename TSeq**>
class epiworld::Tool< **TSeq** >

[Tools](#) for defending the agent against the virus.

Template Parameters

TSeq	Type of sequence

The documentation for this class was generated from the following file:

- epiworld.hpp

13.47 Tool< TSeq > Class Template Reference

[Tools](#) for defending the agent against the virus.

```
#include <tool-bones.hpp>
```

Public Member Functions

- **Tool** (std::string name="unknown tool")
- void **set_sequence** (TSeq d)
- void **set_sequence_unique** (TSeq d)
- void **set_sequence** (std::shared_ptr< TSeq > d)
- std::shared_ptr< TSeq > **get_sequence** ()
- TSeq & **get_sequence_unique** ()
- void **set_name** (std::string name)
- std::string **get_name** () const
- [Agent](#)< TSeq > ∗ **get_agent** ()
- int **get_id** () const
- void **set_id** (int id)
- void **set_date** (int d)
- int **get_date** () const
- void **set_status** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_status** (epiworld_fast_int ∗init, epiworld_fast_int ∗post)
- void **get_queue** (epiworld_fast_int ∗init, epiworld_fast_int ∗post)

Get and set the tool functions

Parameters

v	*The virus over which to operate*
fun	*the function to be used*

Returns

    epiworld_double

- epiworld_double **get_susceptibility_reduction** (VirusPtr< TSeq > v)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v)
- void **set_susceptibility_reduction_fun** (ToolFun< TSeq > fun)
- void **set_transmission_reduction_fun** (ToolFun< TSeq > fun)
- void **set_recovery_enhancer_fun** (ToolFun< TSeq > fun)
- void **set_death_reduction_fun** (ToolFun< TSeq > fun)
- void **set_susceptibility_reduction** (epiworld_double ∗prob)
- void **set_transmission_reduction** (epiworld_double ∗prob)
- void **set_recovery_enhancer** (epiworld_double ∗prob)
- void **set_death_reduction** (epiworld_double ∗prob)
- void **set_susceptibility_reduction** (epiworld_double prob)
- void **set_transmission_reduction** (epiworld_double prob)
- void **set_recovery_enhancer** (epiworld_double prob)
- void **set_death_reduction** (epiworld_double prob)

Friends

- class **Agent**< **TSeq** >
- class **Model**< **TSeq** >
- void **default_add_tool** (Action< TSeq > &a, Model< TSeq > ∗m)
- void **default_rm_tool** (Action< TSeq > &a, Model< TSeq > ∗m)

13.47.1 Detailed Description

template<**typename TSeq**>
class Tool< **TSeq** >

Tools for defending the agent against the virus.

Template Parameters

TSeq	Type of sequence

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/tool-bones.hpp
- include/epiworld/tool-meat.hpp

13.48 epiworld::Tools< TSeq > Class Template Reference

Set of tools (useful for building iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Tools** (Agent< TSeq > &p)
- std::vector< ToolPtr< TSeq > >::iterator **begin** ()
- std::vector< ToolPtr< TSeq > >::iterator **end** ()
- ToolPtr< TSeq > & **operator()** (size_t i)
- ToolPtr< TSeq > & **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Tool**< **TSeq** >
- class **Agent**< **TSeq** >

13.48.1 Detailed Description

template<**typename TSeq**>
class epiworld::Tools< **TSeq** >

Set of tools (useful for building iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- epiworld.hpp

13.49 Tools< TSeq > Class Template Reference

Set of tools (useful for building iterators)

```
#include <tools-bones.hpp>
```

Public Member Functions

- **Tools** (Agent< TSeq > &p)
- std::vector< ToolPtr< TSeq > >::iterator **begin** ()
- std::vector< ToolPtr< TSeq > >::iterator **end** ()
- ToolPtr< TSeq > & **operator()** (size_t i)
- ToolPtr< TSeq > & **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Tool**< **TSeq** >
- class **Agent**< **TSeq** >

13.49.1 Detailed Description

template<**typename TSeq**>
class Tools< **TSeq** >

Set of tools (useful for building iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/tools-bones.hpp

13.50 epiworld::Tools_const< TSeq > Class Template Reference

Set of [Tools](#) (const) (useful for iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Tools_const** (const [Agent](#)< TSeq > &p)
- std::vector< ToolPtr< TSeq > >::const_iterator **begin** () const
- std::vector< ToolPtr< TSeq > >::const_iterator **end** () const
- const ToolPtr< TSeq > & **operator()** (size_t i)
- const ToolPtr< TSeq > & **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Tool**< **TSeq** >
- class **Agent**< **TSeq** >

13.50.1 Detailed Description

template<**typename TSeq**>
class epiworld::Tools_const< **TSeq** >

Set of [Tools](#) (const) (useful for iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- epiworld.hpp

13.51 Tools_const< TSeq > Class Template Reference

Set of [Tools](const) (const) (useful for iterators)

```
#include <tools-bones.hpp>
```

Public Member Functions

- **Tools_const** (const [Agent](const)< TSeq > &p)
- std::vector< ToolPtr< TSeq > >::const_iterator **begin** () const
- std::vector< ToolPtr< TSeq > >::const_iterator **end** () const
- const ToolPtr< TSeq > & **operator()** (size_t i)
- const ToolPtr< TSeq > & **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Tool**< **TSeq** >
- class **Agent**< **TSeq** >

13.51.1 Detailed Description

template<**typename TSeq**>
class Tools_const< **TSeq** >

Set of [Tools](const) (const) (useful for iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/tools-bones.hpp

13.52 epiworld::UserData< TSeq > Class Template Reference

Personalized data by the user.

```
#include <epiworld.hpp>
```

Public Member Functions

- **UserData** (Model< TSeq > &m)
- UserData (std::vector< std::string > names)
    Construct a new User Data object.
- std::vector< std::string > & **get_names** ()
- std::vector< int > & **get_dates** ()
- std::vector< epiworld_double > & **get_data** ()
- void **get_all** (std::vector< std::string > *names=nullptr, std::vector< int > *date=nullptr, std::vector< epiworld_double > *data=nullptr)
- unsigned int **nrow** () const
- unsigned int **ncol** () const
- void **write** (std::string fn)
- void **print** () const

Append data

Parameters

x	A vector of length `ncol()` (if vector), otherwise a `epiworld_double`.
j	Index of the data point, from 0 to `ncol() - 1`.

- void **add** (std::vector< epiworld_double > x)
- void **add** (unsigned int j, epiworld_double x)

Access data

Parameters

i	Row (0 through ndays - 1.)
j	Column (0 through `ncols()`).

Returns

> epiworld_double&

- epiworld_double & **operator()** (unsigned int i, unsigned int j)
- epiworld_double & **operator()** (unsigned int i, std::string name)

Friends

- class **Model< TSeq >**
- class **DataBase< TSeq >**

13.52.1 Detailed Description

template<**typename TSeq**>
class epiworld::UserData< TSeq >

Personalized data by the user.

Template Parameters

TSeq	

13.52.2 Constructor & Destructor Documentation

13.52.2.1 UserData()

```
template<typename TSeq >
UserData< TSeq >::UserData (
            std::vector< std::string > names )  [inline]
```

Construct a new User Data object.

Parameters

names	A vector of names. The length of the vector sets the number of columns to record.

The documentation for this class was generated from the following file:

- epiworld.hpp

13.53 UserData< TSeq > Class Template Reference

Personalized data by the user.

```
#include <userdata-bones.hpp>
```

Public Member Functions

- **UserData** (Model< TSeq > &m)
- UserData (std::vector< std::string > names)
    Construct a new User Data object.
- std::vector< std::string > & **get_names** ()
- std::vector< int > & **get_dates** ()
- std::vector< epiworld_double > & **get_data** ()
- void **get_all** (std::vector< std::string > ∗names=nullptr, std::vector< int > ∗date=nullptr, std::vector< epiworld_double > ∗data=nullptr)
- unsigned int **nrow** () const
- unsigned int **ncol** () const
- void **write** (std::string fn)
- void **print** () const

Append data

Parameters

x	A vector of length `ncol()` (if vector), otherwise a `epiworld_double`.
j	Index of the data point, from 0 to `ncol() - 1`.

- void **add** (std::vector< epiworld_double > x)
- void **add** (unsigned int j, epiworld_double x)

Access data

Parameters

i	Row (0 through ndays - 1.)
j	Column (0 through `ncols()`).

Returns

epiworld_double&

- epiworld_double & **operator()** (unsigned int i, unsigned int j)
- epiworld_double & **operator()** (unsigned int i, std::string name)

Friends

- class **Model**< **TSeq** >
- class **DataBase**< **TSeq** >

13.53.1 Detailed Description

template<**typename TSeq**>
class UserData< **TSeq** >

Personalized data by the user.

Template Parameters

TSeq	

13.53.2 Constructor & Destructor Documentation

13.53.2.1 UserData()

```
template<typename TSeq >
UserData< TSeq >::UserData (
            std::vector< std::string > names )  [inline]
```

Construct a new User Data object.

Parameters

names	A vector of names. The length of the vector sets the number of columns to record.

The documentation for this class was generated from the following files:

- include/epiworld/database-bones.hpp
- include/epiworld/userdata-bones.hpp
- include/epiworld/userdata-meat.hpp

13.54 epiworld::vecHasher< T > Struct Template Reference

Vector hasher.

```
#include <epiworld.hpp>
```

Public Member Functions

- std::size_t **operator()** (std::vector< T > const &dat) const noexcept

13.54.1 Detailed Description

template<**typename T**>
struct epiworld::vecHasher< **T** >

Vector hasher.

Template Parameters

T	

The documentation for this struct was generated from the following file:

- epiworld.hpp

13.55 vecHasher< T > Struct Template Reference

Vector hasher.

```
#include <misc.hpp>
```

Public Member Functions

- std::size_t **operator()** (std::vector< T > const &dat) const noexcept

13.55.1 Detailed Description

template<**typename T**>
struct vecHasher< **T** >

Vector hasher.

Template Parameters

T	

The documentation for this struct was generated from the following file:

- include/epiworld/misc.hpp

13.56 epiworld::Virus< TSeq > Class Template Reference

[Virus](#).

```
#include <epiworld.hpp>
```

Public Member Functions

- **Virus** (std::string name="unknown virus")
- void **mutate** ()
- void **set_mutation** (MutFun< TSeq > fun)
- const TSeq ∗ **get_sequence** ()
- void **set_sequence** (TSeq sequence)
- [Agent](#)< TSeq > ∗ **get_agent** ()
- void **set_agent** ([Agent](#)< TSeq > ∗p, epiworld_fast_uint idx)
- [Model](#)< TSeq > ∗ **get_model** ()
- void **set_date** (int d)
- int **get_date** () const
- void **set_id** (int idx)
- int **get_id** () const
- void **set_name** (std::string name)
- std::string **get_name** () const
- std::vector< epiworld_double > & **get_data** ()

Get and set the tool functions

Parameters

v	*The virus over which to operate*
fun	*the function to be used*

Returns

> *epiworld_double*

- • epiworld_double **get_prob_infecting** ()
- • epiworld_double **get_prob_recovery** ()
- • epiworld_double **get_prob_death** ()
- • void **post_recovery** ()
- • void **set_post_recovery** (PostRecoveryFun< TSeq > fun)
- • void **set_post_immunity** (epiworld_double prob)
- • void **set_post_immunity** (epiworld_double ∗prob)
- • void **set_prob_infecting_fun** (VirusFun< TSeq > fun)
- • void **set_prob_recovery_fun** (VirusFun< TSeq > fun)
- • void **set_prob_death_fun** (VirusFun< TSeq > fun)
- • void **set_prob_infecting** (epiworld_double ∗prob)
- • void **set_prob_recovery** (epiworld_double ∗prob)
- • void **set_prob_death** (epiworld_double ∗prob)
- • void **set_prob_infecting** (epiworld_double prob)
- • void **set_prob_recovery** (epiworld_double prob)
- • void **set_prob_death** (epiworld_double prob)

Get and set the status and queue

After applied, viruses can change the status and affect the queue of agents. These function sets the default values, which are retrieved when adding or removing a virus does not specify a change in status or in queue.

Parameters

init	*After the virus/tool is added to the agent.*
end	*After the virus/tool is removed.*
removed	*After the agent (Agent) is removed.*

- • void **set_status** (epiworld_fast_int init, epiworld_fast_int end, epiworld_fast_int removed=-99)
- • void **set_queue** (epiworld_fast_int init, epiworld_fast_int end, epiworld_fast_int removed=-99)
- • void **get_status** (epiworld_fast_int ∗init, epiworld_fast_int ∗end, epiworld_fast_int ∗removed=nullptr)
- • void **get_queue** (epiworld_fast_int ∗init, epiworld_fast_int ∗end, epiworld_fast_int ∗removed=nullptr)

Friends

- • class **Agent**< **TSeq** >
- • class **Model**< **TSeq** >
- • class **DataBase**< **TSeq** >
- • void **default_add_virus** (Action< TSeq > &a, Model< TSeq > ∗m)
- • void **default_rm_virus** (Action< TSeq > &a, Model< TSeq > ∗m)

13.56.1 Detailed Description

template<**typename TSeq**>
class epiworld::Virus< **TSeq** >

Virus.

Template Parameters

TSeq	

Raw transmisibility of a virus should be a function of its genetic sequence. Nonetheless, transmisibility can be reduced as a result of having one or more tools to fight the virus. Because of this, transmisibility should be a function of the agent.

The documentation for this class was generated from the following file:

- epiworld.hpp

13.57 Virus< TSeq > Class Template Reference

Virus.

```
#include <virus-bones.hpp>
```

Public Member Functions

- **Virus** (std::string name="unknown virus")
- void **mutate** ()
- void **set_mutation** (MutFun< TSeq > fun)
- const TSeq ∗ **get_sequence** ()
- void **set_sequence** (TSeq sequence)
- Agent< TSeq > ∗ **get_agent** ()
- void **set_agent** (Agent< TSeq > ∗p, epiworld_fast_uint idx)
- Model< TSeq > ∗ **get_model** ()
- void **set_date** (int d)
- int **get_date** () const
- void **set_id** (int idx)
- int **get_id** () const
- void **set_name** (std::string name)
- std::string **get_name** () const
- std::vector< epiworld_double > & **get_data** ()

Get and set the tool functions

Parameters

v	*The virus over which to operate*
fun	*the function to be used*

Returns

epiworld_double

- epiworld_double **get_prob_infecting** ()
- epiworld_double **get_prob_recovery** ()
- epiworld_double **get_prob_death** ()
- void **post_recovery** ()
- void **set_post_recovery** (PostRecoveryFun< TSeq > fun)
- void **set_post_immunity** (epiworld_double prob)
- void **set_post_immunity** (epiworld_double ∗prob)
- void **set_prob_infecting_fun** (VirusFun< TSeq > fun)
- void **set_prob_recovery_fun** (VirusFun< TSeq > fun)

- void **set_prob_death_fun** (VirusFun< TSeq > fun)
- void **set_prob_infecting** (epiworld_double ∗prob)
- void **set_prob_recovery** (epiworld_double ∗prob)
- void **set_prob_death** (epiworld_double ∗prob)
- void **set_prob_infecting** (epiworld_double prob)
- void **set_prob_recovery** (epiworld_double prob)
- void **set_prob_death** (epiworld_double prob)

Get and set the status and queue

After applied, viruses can change the status and affect the queue of agents. These function sets the default values, which are retrieved when adding or removing a virus does not specify a change in status or in queue.

Parameters

init	*After the virus/tool is added to the agent.*
end	*After the virus/tool is removed.*
removed	*After the agent ([Agent](#)) is removed.*

- void **set_status** (epiworld_fast_int init, epiworld_fast_int end, epiworld_fast_int removed=-99)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int end, epiworld_fast_int removed=-99)
- void **get_status** (epiworld_fast_int ∗init, epiworld_fast_int ∗end, epiworld_fast_int ∗removed=nullptr)
- void **get_queue** (epiworld_fast_int ∗init, epiworld_fast_int ∗end, epiworld_fast_int ∗removed=nullptr)

Friends

- class **Agent**< **TSeq** >
- class **Model**< **TSeq** >
- class **DataBase**< **TSeq** >
- void **default_add_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > ∗m)
- void **default_rm_virus** ([Action](#)< TSeq > &a, [Model](#)< TSeq > ∗m)

13.57.1 Detailed Description

template<**typename TSeq**>
class Virus< **TSeq** >

[Virus](#).

Template Parameters

TSeq	

Raw transmisibility of a virus should be a function of its genetic sequence. Nonetheless, transmisibility can be reduced as a result of having one or more tools to fight the virus. Because of this, transmisibility should be a function of the agent.

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/virus-bones.hpp
- include/epiworld/virus-meat.hpp

13.58 epiworld::Viruses< TSeq > Class Template Reference

Set of viruses (useful for building iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Viruses** ([Agent](#)< TSeq > &p)
- std::vector< VIRUSPTR >::iterator **begin** ()
- std::vector< VIRUSPTR >::iterator **end** ()
- VIRUSPTR & **operator()** (size_t i)
- VIRUSPTR & **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Virus**< **TSeq** >
- class **Agent**< **TSeq** >

13.58.1 Detailed Description

template<**typename TSeq**>
class epiworld::Viruses< **TSeq** >

Set of viruses (useful for building iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- epiworld.hpp

13.59 Viruses< TSeq > Class Template Reference

Set of viruses (useful for building iterators)

```
#include <viruses-bones.hpp>
```

Public Member Functions

- **Viruses** ([Agent](#)< TSeq > &p)
- std::vector< VIRUSPTR >::iterator **begin** ()
- std::vector< VIRUSPTR >::iterator **end** ()
- VIRUSPTR & **operator()** (size_t i)
- VIRUSPTR & **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Virus**< **TSeq** >
- class **Agent**< **TSeq** >

13.59.1 Detailed Description

template<**typename TSeq**>
class Viruses< **TSeq** >

Set of viruses (useful for building iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/viruses-bones.hpp

13.60 epiworld::Viruses_const< TSeq > Class Template Reference

Set of [Viruses](#) (const) (useful for iterators)

```
#include <epiworld.hpp>
```

Public Member Functions

- **Viruses_const** (const [Agent](#)< TSeq > &p)
- std::vector< VIRUSPTR >::const_iterator **begin** () const
- std::vector< VIRUSPTR >::const_iterator **end** () const
- const VIRUSPTR & **operator()** (size_t i)
- const VIRUSPTR & **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Virus**< **TSeq** >
- class **Agent**< **TSeq** >

13.60.1 Detailed Description

template<**typename TSeq**>
class epiworld::Viruses_const< **TSeq** >

Set of [Viruses](#) (const) (useful for iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following file:

- epiworld.hpp

13.61 Viruses_const< TSeq > Class Template Reference

Set of Viruses (const) (useful for iterators)

```
#include <viruses-bones.hpp>
```

Public Member Functions

- **Viruses_const** (const Agent< TSeq > &p)
- std::vector< VIRUSPTR >::const_iterator **begin** () const
- std::vector< VIRUSPTR >::const_iterator **end** () const
- const VIRUSPTR & **operator()** (size_t i)
- const VIRUSPTR & **operator[ ]** (size_t i)
- size_t **size** () const noexcept

Friends

- class **Virus**< **TSeq** >
- class **Agent**< **TSeq** >

13.61.1 Detailed Description

template< **typename TSeq**>
class Viruses_const< **TSeq** >

Set of Viruses (const) (useful for iterators)

Template Parameters

TSeq	

The documentation for this class was generated from the following files:

- include/epiworld/agent-bones.hpp
- include/epiworld/viruses-bones.hpp

Chapter 14

File Documentation

14.1 include/epiworld/agent-meat-status.hpp File Reference

Sampling functions are getting big, so we keep them in a separate file.

```
#include "agent-meat-virus-sampling.hpp"
```
Include dependency graph for agent-meat-status.hpp:
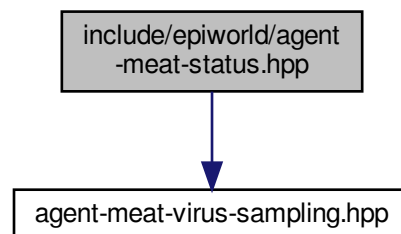
This graph shows which files directly or indirectly include this file:
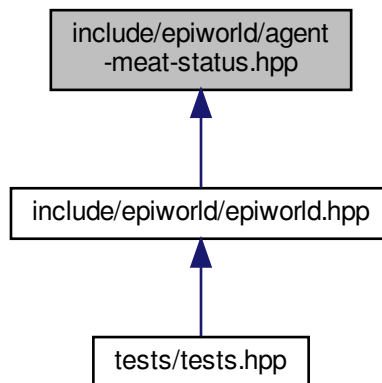
Functions

- template<typename TSeq = EPI_DEFAULT_TSEQ>
  void **default_update_susceptible** (Agent< TSeq > ∗p, Model< TSeq > ∗m)
- template<typename TSeq = EPI_DEFAULT_TSEQ>
  void **default_update_exposed** (Agent< TSeq > ∗p, Model< TSeq > ∗m)

14.1.1 Detailed Description

Sampling functions are getting big, so we keep them in a separate file.

Author

George G. Vega Yon (g.vegayon en gmail)

Version

0.1

Date

2022-06-15

Copyright

Copyright (c) 2022

Index