epiworld

0.0-1

Generated by Doxygen 1.9.1

Chapter 1

Example: 00-hello-world

Output from the program:

```
_____________________________________________________________________
Running the model...
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| done.
 done.
_____________________________________________________________________
_____________________________________________________________________
SIMULATION STUDY
Name of the model   : (none)
Population size     : 10000
Agents' data        : (none)
Number of entities  : 0
Days (duration)     : 100 (of 100)
Number of viruses   : 1
Last run elapsed t  : 16.00ms
Last run speed      : 59.75 million agents x day / second
Rewiring            : off
Global events:
 (none)
Virus(es):
 - covid 19 (baseline prevalence: 50 seeds)
Tool(s):
 - vaccine (baseline prevalence: 50.00%)
Model parameters:
 (none)
Distribution of the population at time 100:
  - (0) Susceptible :  9950 -> 0
  - (1) Exposed     :    50 -> 0
  - (2) Recovered   :     0 -> 9399
  - (3) Removed     :     0 -> 601
Transition Probabilities:
 - Susceptible  0.87  0.13  0.00  0.00
 - Exposed      0.00  0.83  0.15  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 states (arbitrary number of them).

Run

1. Distribute the tool(s) and virus(es)
2. For each t in 1 -> Duration:
    - Update state for susceptible/infected/removed(?)
    - Mutate virus(es) (each individual)
    - Run Global events (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 = state, 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

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

epiworld: A fast and flexible ABM framework for epidemiological

simulations George G. Vega Yon, Ph.D. 2024-04-05

`epiworld` is a fast and flexible agent-based modeling (ABM) framework for epidemiological simulations. Designed in C++, it can simulate large populations with complex interactions. The framework is designed to be modular, allowing users to extend and modify the model to suit their needs quickly. Besides the C++ library, an R package (`epiworldR`), a ShinyApp (`epiworldRShiny`), and a Python library (`epiworldpy`) are available to interact with the model.

7.1 Features

Header-only: epiworld is a header-only template library, making it easy to integrate into existing projects. It is distributed as a collection of files and as `a single header file`.

A framework: epiworld is designed to be a flexible framework for building epidemiological simulations. It provides a set of core components that can be easily extended and modified to suit the user's needs.

Fast: epiworld is designed for speed. It is implemented in C++ and uses efficient data structures and algorithms to ensure simulations dash. Furthermore, epiworld is designed to take advantage of multi-core processors, allowing simulations to be run in parallel (see the `benchmark section` for more details).

Complex disease dynamics: epiworld supports complex disease dynamics, including the evolution of the disease over time. For instance, diseases can accumulate mutations.

Open-source: With funding from the Centers for Disease Control and Prevention [CDC], epiworld is open-source and available under the MIT license, meaning it is entirely free. The source code is available on `GitHub`.

7.2 Example use cases

As a framework, `epiworld` can simulate various epidemiological scenarios. This section provides some possible use cases for the package:

- **Geographically informed models**: With the ability to model complex interactions between agents, the library can simulate geographically informed models featuring multiple regions with different populations and individuals moving between regions.
- **Non-pharmaceutical interventions (NPIs)**: The library has been used to simulate NPIs featuring masking and social distancing. Furthermore, its architecture allows for deploying interventions that are reactive to the model's state, such as a mask mandate that only activates when the prevalence of the disease reaches a certain threshold.
- **Vaccination strategies**: The library can simulate different vaccination strategies, including prioritizing certain groups for vaccination, varying the vaccination rate, and modeling the impact of vaccine hesitancy.
- **Disease evolution**: The library can simulate the evolution of the disease over time, including the emergence of new variants and the impact of these variants on the spread of the disease.
- **Population comorbidities**: The library can model the impact of population comorbidities on the spread of the disease, including how different comorbidities affect the transmission and severity of the disease.

7.3 Appendix

7.3.1 Code example

The following code snippet shows a simple example of how to use epiworld to simulate an epidemic, particularly the Susceptible-Infected-Recovered model [SIR]:

```
#include "epiworld.hpp"
using namespace epiworld;
int main()
{
    // epiworld already comes with a couple
    // of models, like the SIR
    epimodels::ModelSIR<> hello(
        "COVID-19", // Name of the virus
        0.01,        // Initial prevalence
        0.9,        // Transmission probability
        0.3         // Recovery probability
        );
    // We can simulate agents using a small-world network
    // with 100,000 individuals, in this case
    hello.agents_smallworld(100000, 4L, false, .01);
    // Running the model and printing the results
    // Setting the number of days (100) and seed (122)
    hello.run(100, 122);
    hello.print();
    return 0;
}
```

The output could look something like the following:

```
_________________________________________________________________
Running the model...
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| done.
 done.
_________________________________________________________________________
_________________________________________________________________________
SIMULATION STUDY
Name of the model   : Susceptible-Infected-Recovered (SIR)
Population size     : 100000
Agents' data        : (none)
Number of entities  : 0
Days (duration)     : 100 (of 100)
Number of viruses   : 1
Last run elapsed t  : 103.00ms
Last run speed      : 96.34 million agents x day / second
```

```
Rewiring            : off
Global events:
 (none)
Virus(es):
 - COVID-19 (baseline prevalence: 1.00%)
Tool(s):
 (none)
Model parameters:
 - Recovery rate     : 0.3000
 - Transmission rate : 0.9000
Distribution of the population at time 100:
  - (0) Susceptible :  99000 -> 2565
  - (1) Infected    :   1000 -> 366
  - (2) Recovered   :      0 -> 97069
Transition Probabilities:
 - Susceptible  0.96  0.04  0.00
 - Infected     0.00  0.70  0.30
 - Recovered    0.00  0.00  1.00
```

7.3.2 Speed benchmark

7.3.3 Implementation

Chapter 8

EPI Simulator

8.1 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.

8.2 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.

8.3 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.

8.4 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

8.5 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.

8.5.1 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 9

Namespace Index

9.1 Namespace List

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

Chapter 10

Hierarchical Index

10.1 Class Hierarchy

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

Chapter 11

Class Index

11.1 Class List

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

Chapter 12

File Index

12.1 File List

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

Chapter 13

Namespace Documentation

13.1 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)

13.1.1 Detailed Description

Functions for sampling viruses.

13.1.2 Function Documentation

13.1.2.1 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;

13.1.2.2 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;

13.1.2.3 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;

13.2 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)

13.2.1 Detailed Description

Functions for sampling viruses.

13.2.2 Function Documentation

13.2.2.1 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;

13.2.2.2 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;

13.2.2.3 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 14

Class Documentation

14.1 AdjList Class Reference

Public Member Functions

- AdjList (const std::vector< int > &source, const std::vector< int > &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< int, int > **operator()** (epiworld_fast_uint i) const
- void **print** (epiworld_fast_uint 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< int, int > > & **get_dat** ()
- bool is_directed () const

  `true` *if the network is directed.*

14.1.1 Constructor & Destructor Documentation

14.1.1.1 AdjList()

```
AdjList::AdjList (
            const std::vector< int > & source,
            const std::vector< int > & 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

14.1.2 Member Function Documentation

14.1.2.1 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

14.2 epiworld::AdjList Class Reference

Public Member Functions

- AdjList (const std::vector< int > &source, const std::vector< int > &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< int, int > **operator()** (epiworld_fast_uint i) const
- void **print** (epiworld_fast_uint 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< int, int > > & **get_dat** ()
- bool is_directed () const

  `true` *if the network is directed.*

14.2.1 Constructor & Destructor Documentation

14.2.1.1 AdjList()

```
AdjList::AdjList (
            const std::vector< int > & source,
            const std::vector< int > & 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

14.2.2 Member Function Documentation

14.2.2.1 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

14.3 Agent< TSeq > Class Template Reference

Agent (agents)

`#include <agent-bones.hpp>`

Public Member Functions

- **Agent** (Agent< TSeq > &&p)
- **Agent** (const Agent< TSeq > &p)
- Agent< TSeq > & **operator=** (const Agent< TSeq > &other_agent)
- int get_id () const

  Id of the individual.
- VirusPtr< TSeq > & **get_virus** ()
- 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_virus** ()
- void **add_neighbor** (Agent< TSeq > &p, bool check_source=true, bool check_target=true)
- void swap_neighbors (Agent< TSeq > &other, size_t n_this, size_t n_other)

  Swaps neighbors between the current agent and agent `other`
- std::vector< Agent< TSeq > * > **get_neighbors** ()
- size_t **get_n_neighbors** () const
- void **change_state** (Model< TSeq > *model, epiworld_fast_uint new_state, epiworld_fast_int queue=0)
- const epiworld_fast_uint & **get_state** () const
- void **reset** ()
- bool **has_tool** (epiworld_fast_uint t) const
- bool **has_tool** (std::string name) const
- bool **has_tool** (const Tool< TSeq > &t) const
- bool **has_virus** (epiworld_fast_uint t) const
- bool **has_virus** (std::string name) const
- bool **has_virus** (const Virus< TSeq > &v) const
- void **print** (Model< TSeq > *model, bool compressed=false) const
- Entities< TSeq > **get_entities** ()
- const Entities_const< TSeq > **get_entities** () const
- const Entity< TSeq > & **get_entity** (size_t i) const
- Entity< TSeq > & **get_entity** (size_t i)
- size_t **get_n_entities** () const
- bool **operator==** (const Agent< TSeq > &other) const
- bool **operator!=** (const Agent< TSeq > &other) 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*
state_new	*state after the change*
queue	

- void **add_tool** (ToolPtr< TSeq > tool, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **add_tool** (Tool< TSeq > tool, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **set_virus** (VirusPtr< TSeq > virus, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **set_virus** (Virus< TSeq > virus, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **add_entity** (Entity< TSeq > &entity, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (epiworld_fast_uint tool_idx, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (ToolPtr< TSeq > &tool, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_virus** (Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** (epiworld_fast_uint entity_idx, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** (Entity< TSeq > &entity, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void rm_agent_by_virus (Model< TSeq > *model, epiworld_fast_int state_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, Model< TSeq > *model)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v, Model< TSeq > *model)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v, Model< TSeq > *model)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v, Model< TSeq > *model)

- double & operator() (size_t j)

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

Friends

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

14.3.1 Detailed Description

template<typename TSeq>
class Agent< TSeq >

Agent (agents)

Template Parameters

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

14.3.2 Member Function Documentation

14.3.2.1 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&

14.3.2.2 swap_neighbors()

```
template<typename TSeq >
void Agent< TSeq >::swap_neighbors (
            Agent< TSeq > & other,
            size_t n_this,
            size_t n_other )  [inline]
```

Swaps neighbors between the current agent and agent `other`

Parameters

other	
n_this	
n_other	

14.3.3 Friends And Related Function Documentation

14.3.3.1 default_rm_entity

```
template<typename TSeq >
void default_rm_entity (
            Event< TSeq > & a,
            Model< TSeq > * m )  [friend]
```

< Last entity of the agent

< Last agent of the entity

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

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

14.4 epiworld::Agent< TSeq > Class Template Reference

Agent (agents)

`#include <epiworld.hpp>`

Public Member Functions

- **Agent** (Agent< TSeq > &&p)
- **Agent** (const Agent< TSeq > &p)
- Agent< TSeq > & **operator=** (const Agent< TSeq > &other_agent)
- int get_id () const

  Id of the individual.
- VirusPtr< TSeq > & **get_virus** ()
- 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_virus** ()
- void **add_neighbor** (Agent< TSeq > &p, bool check_source=true, bool check_target=true)
- void swap_neighbors (Agent< TSeq > &other, size_t n_this, size_t n_other)

  Swaps neighbors between the current agent and agent `other`
- std::vector< Agent< TSeq > * > **get_neighbors** ()
- size_t **get_n_neighbors** () const
- void **change_state** (Model< TSeq > *model, epiworld_fast_uint new_state, epiworld_fast_int queue=0)
- const epiworld_fast_uint & **get_state** () const
- void **reset** ()
- bool **has_tool** (epiworld_fast_uint t) const
- bool **has_tool** (std::string name) const
- bool **has_tool** (const Tool< TSeq > &t) const
- bool **has_virus** (epiworld_fast_uint t) const
- bool **has_virus** (std::string name) const
- bool **has_virus** (const Virus< TSeq > &v) const
- void **print** (Model< TSeq > *model, bool compressed=false) const
- Entities< TSeq > **get_entities** ()
- const Entities_const< TSeq > **get_entities** () const
- const Entity< TSeq > & **get_entity** (size_t i) const
- Entity< TSeq > & **get_entity** (size_t i)
- size_t **get_n_entities** () const
- bool **operator==** (const Agent< TSeq > &other) const
- bool **operator!=** (const Agent< TSeq > &other) 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*
state_new	*state after the change*
queue	

- void **add_tool** (ToolPtr< TSeq > tool, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **add_tool** (Tool< TSeq > tool, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **set_virus** (VirusPtr< TSeq > virus, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **set_virus** (Virus< TSeq > virus, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **add_entity** (Entity< TSeq > &entity, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (epiworld_fast_uint tool_idx, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_tool** (ToolPtr< TSeq > &tool, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_virus** (Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** (epiworld_fast_uint entity_idx, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void **rm_entity** (Entity< TSeq > &entity, Model< TSeq > *model, epiworld_fast_int state_new=-99, epiworld_fast_int queue=-99)
- void rm_agent_by_virus (Model< TSeq > *model, epiworld_fast_int state_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, Model< TSeq > *model)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v, Model< TSeq > *model)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v, Model< TSeq > *model)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v, Model< TSeq > *model)

- double & operator() (size_t j)

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

Friends

- class **Model**< **TSeq** >
- class **Virus**< **TSeq** >
- class **Tool**< **TSeq** >
- class **Tools**< **TSeq** >
- class **Tools_const**< **TSeq** >
- class **Queue**< **TSeq** >
- class **Entities**< **TSeq** >

- class **AgentsSample**< **TSeq** >
- void **default_add_virus** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_add_tool** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_add_entity** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_virus** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_tool** (Event< TSeq > &a, Model< TSeq > *m)
- void default_rm_entity (Event< TSeq > &a, Model< TSeq > *m)
- void **default_change_state** (Event< TSeq > &a, Model< TSeq > *m)

14.4.1 Detailed Description

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

Agent (agents)

Template Parameters

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

14.4.2 Member Function Documentation

14.4.2.1 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&

14.4.2.2 swap_neighbors()

```
template<typename TSeq >
void Agent< TSeq >::swap_neighbors (
             Agent< TSeq > & other,
             size_t n_this,
             size_t n_other )  [inline]
```

Swaps neighbors between the current agent and agent `other`

Parameters

other	
n_this	
n_other	

14.4.3 Friends And Related Function Documentation

14.4.3.1 default_rm_entity

```
template<typename TSeq >
void default_rm_entity (
             Event< TSeq > & a,
             Model< TSeq > * m )  [friend]
```

< Last entity of the agent

< Last agent of the entity

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

- epiworld.hpp

14.5 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, std::vector< size_t > states_={}, bool truncate=false)
- **AgentsSample** (Model< TSeq > *model, Entity< TSeq > &entity_, size_t n, std::vector< size_t > states_={}, bool truncate=false)
- AgentsSample (Model< TSeq > *model, Agent< TSeq > &agent_, size_t n, std::vector< size_t > states_={}, bool truncate=false)

  Sample from the agent's entities.
- std::vector< Agent< TSeq > * >::iterator **begin** ()
- std::vector< Agent< TSeq > * >::iterator **end** ()
- Agent< TSeq > * **operator[ ]** (size_t n)
- Agent< TSeq > * **operator()** (size_t n)
- size_t **size** () const noexcept

14.5.1 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	

14.5.2 Constructor & Destructor Documentation

14.5.2.1 AgentsSample()

```
template<typename TSeq >
AgentsSample< TSeq >::AgentsSample (
            Model< TSeq > * model,
            Agent< TSeq > & agent_,
            size_t n,
            std::vector< size_t > states_ = {},
            bool truncate = false )  [inline]
```

Sample from the agent's entities.

For example, how many individuals the agent contacts in a given point in time.

Template Parameters

TSeq	

Parameters

*agent_*	
n	Sample size
truncate	If the agent has fewer than `n` connections, then truncate = true will automatically reduce the number of possible samples. Otherwise, if false, then it returns an error.

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

- include/epiworld/agentssample-bones.hpp

14.6 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, std::vector< size_t > states_={}, bool truncate=false)
- **AgentsSample** (Model< TSeq > *model, Entity< TSeq > &entity_, size_t n, std::vector< size_t > states_={}, bool truncate=false)
- AgentsSample (Model< TSeq > *model, Agent< TSeq > &agent_, size_t n, std::vector< size_t > states_={}, bool truncate=false)
  Sample from the agent's entities.
- std::vector< Agent< TSeq > * >::iterator **begin** ()
- std::vector< Agent< TSeq > * >::iterator **end** ()
- Agent< TSeq > * **operator[ ]** (size_t n)
- Agent< TSeq > * **operator()** (size_t n)
- size_t **size** () const noexcept

14.6.1 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	

14.6.2 Constructor & Destructor Documentation

14.6.2.1 AgentsSample()

```
template<typename TSeq >
AgentsSample< TSeq >::AgentsSample (
            Model< TSeq > * model,
            Agent< TSeq > & agent_,
            size_t n,
            std::vector< size_t > states_ = {},
            bool truncate = false )  [inline]
```

Sample from the agent's entities.

For example, how many individuals the agent contacts in a given point in time.

Template Parameters

TSeq	

Parameters

*agent_*	
n	Sample size
truncate	If the agent has fewer than n connections, then truncate = true will automatically reduce the number of possible samples. Otherwise, if false, then it returns an error.

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

- epiworld.hpp

14.7 DataBase< TSeq > Class Template Reference

Statistical data about the process.

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

Public Member Functions

- **DataBase** (Model< TSeq > &m)
- **DataBase** (const DataBase< TSeq > &db)
- void record_virus (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 **reset** ()
- 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_virus_info, std::string fn_virus_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, std::string fn_generation_time) const
- void **record_transmission** (int i, int j, int virus, int i_expo_date)
- size_t **get_n_viruses** () const
- size_t **get_n_tools** () const
- void **set_user_data** (std::vector< std::string > names)
- void **add_user_data** (std::vector< epiworld_double > x)
- void **add_user_data** (epiworld_fast_uint j, epiworld_double x)
- UserData< TSeq > & **get_user_data** ()
- std::vector< epiworld_double > transition_probability (bool print=true) const

  Calculates the transition probabilities.
- bool operator== (const DataBase< TSeq > &other) const
- bool **operator!=** (const DataBase< TSeq > &other) const
- bool operator== (const DataBase< std::vector< int >> &other) const
- bool operator== (const DataBase< std::vector< int >> &other) const

Get recorded information from the model

Parameters

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

Returns

In `get_today_total`, the current counts of `what`.
In `get_today_virus`, the current counts of `what` for each virus.
In `get_hist_total`, the time series of `what`
In `get_hist_virus`, the time series of what for each virus.
In `get_hist_total_date` and `get_hist_virus_date` the corresponding date

- 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 > *state=nullptr, std::vector< int > *counts=nullptr) const
- void **get_today_virus** (std::vector< std::string > &state, std::vector< int > &id, std::vector< int > &counts) const
- void **get_hist_total** (std::vector< int > *date, std::vector< std::string > *state, std::vector< int > *counts) const
- void **get_hist_virus** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &state, std::vector< int > &counts) const
- void **get_hist_tool** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &state, std::vector< int > &counts) const

- void **get_hist_transition_matrix** (std::vector< std::string > &state_from, std::vector< std::string > &state_to, std::vector< int > &date, std::vector< int > &counts, bool skip_zeros) const

- void get_transmissions (std::vector< int > &date, std::vector< int > &source, std::vector< int > &target, std::vector< int > &virus, std::vector< int > &source_exposure_date) const

  Get the transmissions object.
- void **get_transmissions** (int *date, int *source, int *target, int *virus, int *source_exposure_date) const

- MapVec_type< int, int > reproductive_number () const

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

- void generation_time (std::vector< int > &agent_id, std::vector< int > &virus_id, std::vector< int > &time, std::vector< int > &gentime) const
- void **generation_time** (std::string fn) const

Friends

- class **Model**< **TSeq** >
- void **default_add_virus** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_add_tool** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_virus** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_tool** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_change_state** (Event< TSeq > &a, Model< TSeq > *m)

14.7.1 Detailed Description

template<typename TSeq>
class DataBase< TSeq >

Statistical data about the process.

Template Parameters

TSeq	

14.7.2 Member Function Documentation

14.7.2.1 generation_time()

```
template<typename TSeq >
void DataBase< TSeq >::generation_time (
            std::vector< int > & agent_id,
            std::vector< int > & virus_id,
            std::vector< int > & time,
            std::vector< int > & gentime ) const  [inline]
```

Calculates the generating time

Parameters

agent_id,virus_id,time,gentime	vectors where to save the values agent_id

14.7.2.2 get_transmissions()

```
template<typename TSeq >
void DataBase< TSeq >::get_transmissions (
            std::vector< int > & date,
            std::vector< int > & source,
            std::vector< int > & target,
            std::vector< int > & virus,
            std::vector< int > & source_exposure_date ) const  [inline]
```

Get the transmissions object.

Parameters

date	
source	
target	
virus	
source_exposure_date	

14.7.2.3 operator==() [1/3]

```
bool DataBase< std::vector< int > >::operator== (
            const DataBase< std::vector< int >> & other ) const  [inline]
```

< Date of the transmission eve,

< Id of the sour,

< Id of the targ,

< Id of the varia,

< Date when the source acquired the varia,

14.7.2.4 operator==() [2/3]

```
bool DataBase< std::vector< int > >::operator== (
            const DataBase< std::vector< int >> & other ) const  [inline]
```

< Date of the transmission eve,

< Id of the sour,

< Id of the targ,

< Id of the varia,

< Date when the source acquired the varia,

14.7.2.5 operator==() [3/3]

```
template<typename TSeq >
bool DataBase< TSeq >::operator== (
            const DataBase< TSeq > & other ) const  [inline]
```

< Date of the transmission eve

< Id of the sour

< Id of the targ

< Id of the varia

< Date when the source acquired the varia

14.7.2.6 record_virus()

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

Registering a new variant.

Parameters

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

14.7.2.7 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.

14.7.2.8 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

14.8 epiworld::DataBase< TSeq > Class Template Reference

Statistical data about the process.

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

Public Member Functions

- **DataBase** (Model< TSeq > &m)
- **DataBase** (const DataBase< TSeq > &db)
- void record_virus (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 **reset** ()
- 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_virus_info, std::string fn_virus_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, std::string fn_generation_time) const
- void **record_transmission** (int i, int j, int virus, int i_expo_date)
- size_t **get_n_viruses** () const
- size_t **get_n_tools** () const
- void **set_user_data** (std::vector< std::string > names)
- void **add_user_data** (std::vector< epiworld_double > x)
- void **add_user_data** (epiworld_fast_uint j, epiworld_double x)
- UserData< TSeq > & **get_user_data** ()
- std::vector< epiworld_double > transition_probability (bool print=true) const

  Calculates the transition probabilities.
- bool operator== (const DataBase< TSeq > &other) const
- bool **operator!=** (const DataBase< TSeq > &other) const

Get recorded information from the model

Parameters

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

Returns

In `get_today_total`*, the current counts of* `what`*.*

In `get_today_virus`*, the current counts of* `what` *for each virus.*

In `get_hist_total`*, the time series of* `what`

In `get_hist_virus`*, the time series of what for each virus.*

In `get_hist_total_date` *and* `get_hist_virus_date` *the corresponding date*

- 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 > *state=nullptr, std::vector< int > *counts=nullptr) const
- void **get_today_virus** (std::vector< std::string > &state, std::vector< int > &id, std::vector< int > &counts) const
- void **get_hist_total** (std::vector< int > *date, std::vector< std::string > *state, std::vector< int > *counts) const
- void **get_hist_virus** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &state, std::vector< int > &counts) const
- void **get_hist_tool** (std::vector< int > &date, std::vector< int > &id, std::vector< std::string > &state, std::vector< int > &counts) const
- void **get_hist_transition_matrix** (std::vector< std::string > &state_from, std::vector< std::string > &state_to, std::vector< int > &date, std::vector< int > &counts, bool skip_zeros) const

- void get_transmissions (std::vector< int > &date, std::vector< int > &source, std::vector< int > &target, std::vector< int > &virus, std::vector< int > &source_exposure_date) const

  Get the transmissions object.
- void **get_transmissions** (int *date, int *source, int *target, int *virus, int *source_exposure_date) const

- MapVec_type< int, int > reproductive_number () const

Computes the reproductive number of each case.

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

- void generation_time (std::vector< int > &agent_id, std::vector< int > &virus_id, std::vector< int > &time, std::vector< int > &gentime) const
- void **generation_time** (std::string fn) const

Friends

- class **Model**< **TSeq** >
- void **default_add_virus** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_add_tool** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_virus** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_tool** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_change_state** (Event< TSeq > &a, Model< TSeq > *m)

14.8.1 Detailed Description

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

Statistical data about the process.

Template Parameters

TSeq	

14.8.2 Member Function Documentation

14.8.2.1 generation_time()

```
template<typename TSeq >
void DataBase< TSeq >::generation_time (
            std::vector< int > & agent_id,
            std::vector< int > & virus_id,
            std::vector< int > & time,
            std::vector< int > & gentime ) const  [inline]
```

Calculates the generating time

Parameters

agent_id,virus_id,time,gentime	vectors where to save the values agent_id

14.8.2.2 get_transmissions()

```
template<typename TSeq >
void DataBase< TSeq >::get_transmissions (
            std::vector< int > & date,
            std::vector< int > & source,
            std::vector< int > & target,
            std::vector< int > & virus,
            std::vector< int > & source_exposure_date ) const  [inline]
```

Get the transmissions object.

Parameters

date	
source	
target	
virus	
source_exposure_date	

14.8.2.3 operator==()

```
template<typename TSeq >
bool DataBase< TSeq >::operator== (
            const DataBase< TSeq > & other ) const  [inline]
```

< Date of the transmission eve

< Id of the sour

< Id of the targ

< Id of the varia

< Date when the source acquired the varia

14.8.2.4 record_virus()

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

Registering a new variant.

Parameters

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

14.8.2.5 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.

14.8.2.6 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

14.9 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
- bool **operator==** (const Entities< TSeq > &other) const

Friends

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

14.9.1 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

14.10 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
- bool **operator==** (const Entities< TSeq > &other) const

Friends

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

14.10.1 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

14.11 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
- bool **operator==** (const Entities_const< TSeq > &other) const

Friends

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

14.11.1 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

14.12 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
- bool **operator==** (const Entities_const< TSeq > &other) const

Friends

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

14.12.1 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

14.13 Entity< TSeq > Class Template Reference

Public Member Functions

- **Entity** (std::string name)
- void **add_agent** (Agent< TSeq > &p, Model< TSeq > *model)
- void **add_agent** (Agent< TSeq > *p, Model< TSeq > *model)
- 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_state** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **reset** ()
- bool **operator==** (const Entity< TSeq > &other) const
- bool **operator!=** (const Entity< TSeq > &other) const

Friends

- class **Agent**< **TSeq** >
- class **AgentsSample**< **TSeq** >
- class **Model**< **TSeq** >
- void **default_add_entity** (Event< TSeq > &a, Model< TSeq > *m)
- void default_rm_entity (Event< TSeq > &a, Model< TSeq > *m)

14.13.1 Friends And Related Function Documentation

14.13.1.1 default_rm_entity

```
template<typename TSeq >
void default_rm_entity (
            Event< TSeq > & a,
            Model< TSeq > * m )  [friend]
```

< Last entity of the agent

< Last agent of the entity

< Last entity of the agent

< Last agent of the entity

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

14.14 epiworld::Entity< TSeq > Class Template Reference

Public Member Functions

- **Entity** (std::string name)
- void **add_agent** (Agent< TSeq > &p, Model< TSeq > *model)
- void **add_agent** (Agent< TSeq > *p, Model< TSeq > *model)
- 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_state** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **reset** ()
- bool **operator==** (const Entity< TSeq > &other) const
- bool **operator!=** (const Entity< TSeq > &other) const

Friends

- class **Agent**< **TSeq** >
- class **AgentsSample**< **TSeq** >
- class **Model**< **TSeq** >
- void **default_add_entity** (Event< TSeq > &a, Model< TSeq > *m)
- void default_rm_entity (Event< TSeq > &a, Model< TSeq > *m)

14.14.1 Friends And Related Function Documentation

14.14.1.1 default_rm_entity

```
template<typename TSeq >
void default_rm_entity (
            Event< TSeq > & a,
            Model< TSeq > * m )  [friend]
```

< Last entity of the agent

< Last agent of the entity

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

- epiworld.hpp

14.15 epiworld::Event< TSeq > Struct Template Reference

Event data for update an agent.

`#include <epiworld.hpp>`

Public Member Functions

- Event (Agent< TSeq > *agent_, VirusPtr< TSeq > virus_, ToolPtr< TSeq > tool_, Entity< TSeq > *entity_, epiworld_fast_int new_state_, epiworld_fast_int queue_, ActionFun< TSeq > call_, int idx_agent_, int idx_object_)

  Construct a new Event object.

Public Attributes

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

14.15.1 Detailed Description

template<typename TSeq>
struct epiworld::Event< TSeq >

Event data for update an agent.

Template Parameters

TSeq	

14.15.2 Constructor & Destructor Documentation

14.15.2.1 Event()

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

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

Construct a new Event 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_state_*	Next state
*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

14.16 Event< TSeq > Struct Template Reference

Event data for update an agent.

`#include <config.hpp>`

Public Member Functions

- Event (Agent< TSeq > *agent_, VirusPtr< TSeq > virus_, ToolPtr< TSeq > tool_, Entity< TSeq > *entity_, epiworld_fast_int new_state_, epiworld_fast_int queue_, ActionFun< TSeq > call_, int idx_agent_, int idx_object_)

  Construct a new Event object.

Public Attributes

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

14.16.1 Detailed Description

template<typename TSeq>
struct Event< TSeq >

Event data for update an agent.

Template Parameters

TSeq	

14.16.2 Constructor & Destructor Documentation

14.16.2.1 Event()

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

Construct a new Event object.

All the parameters are rather optional.

Parameters

*agent_*	Agent over who the action will happen
*virus_*	Virus to add
*tool_*	Tool to add

Parameters

virus_idx	Index of virus to be removed (if needed)
tool_idx	Index of tool to be removed (if needed)
*new_↩ state_*	Next state
*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

14.17 epiworld::GlobalEvent< TSeq > Class Template Reference

Template for a Global Event.

`#include <epiworld.hpp>`

Public Member Functions

- GlobalEvent (GlobalFun< TSeq > fun, std::string name, int day=-99)
  Construct a new Global Event object.
- void **operator()** (Model< TSeq > *m, int day)
- void **set_name** (std::string name)
- std::string **get_name** () const
- void **set_day** (int day)
- int **get_day** () const
- void **print** () const
- bool **operator==** (const GlobalEvent< TSeq > &other) const
- bool **operator!=** (const GlobalEvent< TSeq > &other) const

14.17.1 Detailed Description

template<typename TSeq>
class epiworld::GlobalEvent< TSeq >

Template for a Global Event.

Global events are functions that Model<TSeq> executes at the end of a day.

14.17.2 Constructor & Destructor Documentation

14.17.2.1 GlobalEvent()

```
template<typename TSeq >
GlobalEvent< TSeq >::GlobalEvent (
            GlobalFun< TSeq > fun,
            std::string name,
            int day = -99 )  [inline]
```

Construct a new Global Event object.

Parameters

fun	A function that takes a Model<TSeq> * as argument and returns void.
name	A descriptive name for the action.
day	The day when the action will be executed. If negative, it will be executed every day.

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

- epiworld.hpp

14.18 GlobalEvent< TSeq > Class Template Reference

Template for a Global Event.

`#include <globalevent-bones.hpp>`

Public Member Functions

- GlobalEvent (GlobalFun< TSeq > fun, std::string name, int day=-99)
  Construct a new Global Event object.
- void **operator()** (Model< TSeq > *m, int day)
- void **set_name** (std::string name)
- std::string **get_name** () const
- void **set_day** (int day)
- int **get_day** () const
- void **print** () const
- bool **operator==** (const GlobalEvent< TSeq > &other) const
- bool **operator!=** (const GlobalEvent< TSeq > &other) const

14.18.1 Detailed Description

template<typename TSeq>
class GlobalEvent< TSeq >

Template for a Global Event.

Global events are functions that Model<TSeq> executes at the end of a day.

14.18.2 Constructor & Destructor Documentation

14.18.2.1 GlobalEvent()

```
template<typename TSeq >
GlobalEvent< TSeq >::GlobalEvent (
            GlobalFun< TSeq > fun,
            std::string name,
            int day = -99 )  [inline]
```

Construct a new Global Event object.

Parameters

fun	A function that takes a Model<TSeq> * as argument and returns void.
name	A descriptive name for the action.
day	The day when the action will be executed. If negative, it will be executed every day.

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

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

14.19 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)
- std::vector< epiworld_double > **get_params_mean** ()
- std::vector< epiworld_double > **get_stats_mean** ()
- void **print** ()

Random number generation

Parameters

eng	

- void **set_rand_engine** (std::mt19937 &eng)
- std::mt19937 & **get_rand_endgine** ()
- void **seed** (epiworld_fast_uint 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)

14.19.1 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

14.20 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)
- std::vector< epiworld_double > **get_params_mean** ()
- std::vector< epiworld_double > **get_stats_mean** ()
- void **print** ()

Random number generation

Parameters

eng	

- void **set_rand_engine** (std::mt19937 &eng)
- std::mt19937 & **get_rand_endgine** ()
- void **seed** (epiworld_fast_uint 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)

14.20.1 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:

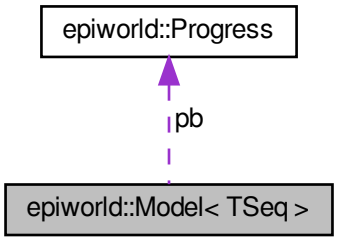
- include/epiworld/math/lfmcmc/lfmcmc-bones.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat-print.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat.hpp

14.21 epiworld::Model< TSeq > Class Template Reference

Core class of epiworld.

`#include <epiworld.hpp>`

Collaboration diagram for epiworld::Model< TSeq >:



Public Member Functions

- DataBase< TSeq > & **get_db** ()
- epiworld_double & **operator()** (std::string pname)
- size_t **size** () const
- void load_agents_entities_ties (std::string fn, int skip)

  Associate agents-entities from a file.
- size_t get_n_viruses () const

  Number of viruses in the model.
- size_t get_n_tools () const

  Number of tools in the model.
- epiworld_fast_uint **get_ndays** () const
- epiworld_fast_uint **get_n_replicates** () const
- void **set_ndays** (epiworld_fast_uint ndays)
- bool **get_verbose** () const
- Model< TSeq > & **verbose_off** ()
- Model< TSeq > & **verbose_on** ()

- int today () const

  The current time of the model.
- void write_data (std::string fn_virus_info, std::string fn_virus_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, std::string fn_generation_time) const

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

  Reset the model.
- const Model< TSeq > & **print** (bool lite=false) 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_globalevent (std::function< void(Model< TSeq > *)> fun, std::string name="A global action", int date=-99)

  Set a global action.
- void **add_globalevent** (GlobalEvent< TSeq > action)
- GlobalEvent< TSeq > & get_globalevent (std::string name)

  Retrieve a global action by name.
- GlobalEvent< TSeq > & get_globalevent (size_t i)

  Retrieve a global action by index.
- void rm_globalevent (std::string name)

  Remove a global action by name.
- void rm_globalevent (size_t i)

  Remove a global action by index.
- void **run_globalevents** ()
- void **clear_state_set** ()
- const std::vector< VirusPtr< TSeq > > & **get_viruses** () const
- const std::vector< epiworld_double > & **get_prevalence_virus** () const
- const std::vector< bool > & **get_prevalence_virus_as_proportion** () const
- const std::vector< ToolPtr< TSeq > > & **get_tools** () const
- Virus< TSeq > & **get_virus** (size_t id)
- Tool< TSeq > & **get_tool** (size_t id)
- void set_agents_data (double *data_, size_t ncols_)

  Set the agents data object.
- double * **get_agents_data** ()
- size_t **get_agents_data_ncols** () const
- void set_name (std::string name)

  Set the name object.
- std::string **get_name** () const
- bool **operator==** (const Model< TSeq > &other) const
- bool **operator!=** (const Model< TSeq > &other) const
- void events_run ()

  Executes the stored action.

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** ()

Random number generation

Parameters

eng	*Random number generator*
s	*Seed*

- void **set_rand_engine** (std::mt19937 &eng)
- std::mt19937 & **get_rand_endgine** ()
- void **seed** (size_t 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)
- void **set_rand_binom** (int n, epiworld_double p)
- 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)
- int **rbinom** ()
- int **rbinom** (int n, epiworld_double p)

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, epiworld_fast_uint 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, epiworld_fast_uint preval)
- void **add_tool_fun** (Tool< TSeq > &t, ToolToAgentFun< TSeq > fun)
- void **add_entity** (Entity< TSeq > e)
- void **rm_virus** (size_t virus_pos)
- void **rm_tool** (size_t tool_pos)
- void **rm_entity** (size_t entity_pos)

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_edgelist** (const std::vector< int > &source, const std::vector< int > &target, int size, bool directed)
- void **agents_from_adjlist** (AdjList al)
- bool **is_directed** () const
- std::vector< Agent< TSeq > > & get_agents ()

  Returns a reference to the vector of agents.
- std::vector< epiworld_fast_uint > get_agents_states () const

  Returns a vector with the states of the agents.
- std::vector< Viruses_const< TSeq > > get_agents_viruses () const

  Returns a const vector with the viruses of the agents.
- std::vector< Viruses< TSeq > > get_agents_viruses ()

  Returns a vector with the viruses of the agents.
- std::vector< Entity< TSeq > > & **get_entities** ()
- Model< TSeq > & **agents_smallworld** (epiworld_fast_uint n=1000, epiworld_fast_uint k=5, bool d=false, epiworld_double p=.01)
- void **agents_empty_graph** (epiworld_fast_uint 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 **update_state** ()
- void **mutate_virus** ()
- void **next** ()
- virtual Model< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void run_multiple (epiworld_fast_uint ndays, epiworld_fast_uint nexperiments, int seed_=-1, std::function< void(size_t, Model< TSeq > *)> fun=make_save_run< TSeq >(), bool reset=true, bool verbose=true, int nthreads=1)

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< int > &source, std::vector< int > &target) const

Manage state (states) in the model

The functions `get_state` return the current values for the states included in the model.

Parameters

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

Returns

`add_state*` returns nothing.

`get_state_*` returns a vector of pairs with the states and their labels.

- void **add_state** (std::string lab, UpdateFun< TSeq > fun=nullptr)
- const std::vector< std::string > & **get_states** () const
- const std::vector< UpdateFun< TSeq > > & **get_state_fun** () const
- void **print_state_codes** () const

Initial states

These functions are called before the simulation starts.

Parameters

proportions_	*Vector of proportions for each state.*
queue_	*Vector of queue for each state.*

- virtual Model< TSeq > & **initial_states** (std::vector< double >, std::vector< int >)

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 `epiworld_fast_uint` 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 **add_param** (epiworld_double initial_val, std::string pname)
- void **read_params** (std::string fn)
- epiworld_double **get_param** (epiworld_fast_uint k)
- epiworld_double **get_param** (std::string pname)
- void **set_param** (std::string pname, epiworld_double val)
- epiworld_double **par** (std::string pname)

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** (epiworld_fast_uint 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.)
- Model< TSeq > & 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)

Protected Member Functions

- void **dist_tools** ()
- void **dist_virus** ()
- void **chrono_start** ()
- void **chrono_end** ()
- void events_add (Agent< TSeq > *agent_, VirusPtr< TSeq > virus_, ToolPtr< TSeq > tool_, Entity< TSeq > *entity_, epiworld_fast_int new_state_, epiworld_fast_int queue_, ActionFun< TSeq > call_, int idx_agent_, int idx_object_)

  Construct a new Event object.

Protected Attributes

- std::string name = ""

  Name of the model.
- DataBase< TSeq > **db** = DataBase<TSeq>(*this)
- std::vector< Agent< TSeq > > **population** = {}
- bool **using_backup** = true
- std::vector< Agent< TSeq > > **population_backup** = {}
- bool **directed** = false
- std::vector< VirusPtr< TSeq > > **viruses** = {}
- std::vector< epiworld_double > prevalence_virus = {}

  Initial prevalence_virus of each virus.
- std::vector< bool > **prevalence_virus_as_proportion** = {}
- std::vector< VirusToAgentFun< TSeq > > **viruses_dist_funs** = {}
- std::vector< ToolPtr< TSeq > > **tools** = {}
- std::vector< epiworld_double > **prevalence_tool** = {}
- std::vector< bool > **prevalence_tool_as_proportion** = {}
- std::vector< ToolToAgentFun< TSeq > > **tools_dist_funs** = {}
- std::vector< Entity< TSeq > > **entities** = {}
- std::vector< Entity< TSeq > > **entities_backup** = {}
- std::mt19937 **engine**
- std::uniform_real_distribution **runifd**
- std::normal_distribution **rnormd**
- std::gamma_distribution **rgammad**
- std::lognormal_distribution **rlognormald**
- std::exponential_distribution **rexpd**
- std::binomial_distribution **rbinomd**
- std::function< void(std::vector< Agent< TSeq >> *, Model< TSeq > *, epiworld_double)> **rewire_fun**
- epiworld_double **rewire_prop** = 0.0
- std::map< std::string, epiworld_double > **parameters**
- epiworld_fast_uint **ndays** = 0
- Progress **pb**
- std::vector< UpdateFun< TSeq > > state_fun = {}

  Functions to update states.
- std::vector< std::string > states_labels = {}

  Labels of the states.
- std::function< void(Model< TSeq > *)> initial_states_fun
- epiworld_fast_uint **nstates** = 0u
- bool **verbose** = true
- int **current_date** = 0
- std::chrono::time_point< std::chrono::steady_clock > **time_start**
- std::chrono::time_point< std::chrono::steady_clock > **time_end**

- std::chrono::duration< epiworld_double, std::micro > **time_elapsed**
- epiworld_fast_uint **n_replicates** = 0u
- std::vector< GlobalEvent< TSeq > > **globalevents**
- Queue< TSeq > **queue**
- bool **use_queuing** = true
- std::vector< Event< TSeq > > events = {}

  Variables used to keep track of the events to be made regarding viruses.
- epiworld_fast_uint **nactions** = 0u

Auxiliary variables for AgentsSample<TSeq> iterators

These variables+objects are used by the AgentsSample<TSeq> class for building efficient iterators over agents. The idea is to reduce the memory allocation, so only during the first call of AgentsSample<TSeq>::Agents↩ Sample(Model<TSeq>) these vectors are allocated.

- std::vector< Agent< TSeq > * > **sampled_population**
- size_t **sampled_population_n** = 0u
- std::vector< size_t > **population_left**
- size_t **population_left_n** = 0u

Agents features

Optionally, a model can include an external data source pointing to agents information. The data can then be access through the `Agent::operator()` method.

- double * **agents_data** = nullptr
- size_t **agents_data_ncols** = 0u

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.

- MixerFun< TSeq > **susceptibility_reduction_mixer** = susceptibility_reduction_mixer_default<TSeq>
- MixerFun< TSeq > **transmission_reduction_mixer** = transmission_reduction_mixer_default<TSeq>
- MixerFun< TSeq > **recovery_enhancer_mixer** = recovery_enhancer_mixer_default<TSeq>
- MixerFun< TSeq > **death_reduction_mixer** = death_reduction_mixer_default<TSeq>
- std::vector< epiworld_double > **array_double_tmp**
- std::vector< Virus< TSeq > * > **array_virus_tmp**
- virtual Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- **Model** ()
- **Model** (const Model< TSeq > &m)
- **Model** (Model< TSeq > &m)
- **Model** (Model< TSeq > &&m)
- Model< TSeq > & **operator=** (const Model< TSeq > &m)
- virtual ~**Model** ()
- void **clone_population** (std::vector< Agent< TSeq > > &other_population, std::vector< Entity< TSeq > > &other_entities, Model< TSeq > *other_model, bool &other_directed) const
- void **clone_population** (const Model< TSeq > &other_model)

14.21.1 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.)

14.21.2 Member Function Documentation

14.21.2.1 add_globalevent()

```
template<typename TSeq >
void Model< TSeq >::add_globalevent (
             std::function< void(Model< TSeq > *)> fun,
             std::string name = "A global action",
             int date = -99 )  [inline]
```

Set a global action.

Parameters

fun	A function to be called on the prescribed date
name	Name of the action.
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.

14.21.2.2 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * Model< TSeq >::clone_ptr  [inline], [protected], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented in ModelSIRLogit< TSeq >, ModelSIRDCONN< TSeq >, ModelSIRCONN< TSeq >, ModelSEIRDCONN< TSeq >, ModelSEIRCONN< TSeq >, epiworld::epimodels::ModelSIRLogit< TSeq >, epiworld::epimodels::ModelSEIRDCONN< TSeq >, epiworld::epimodels::ModelSIRDCONN< TSeq >, epiworld::epimodels::ModelSEIRCONN< TSeq >, and epiworld::epimodels::ModelSIRCONN< TSeq >.

14.21.2.3 events_add()

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

Construct a new Event object.

Parameters

*agent_*	Agent over which the action will be called
*virus_*	Virus pointer included in the action
*tool_*	Tool pointer included in the action
*entity_*	Entity pointer included in the action
*new_state_*	New state of the agent
*call_*	Function the action will call
*queue_*	Change in the queue
*idx_agent_*	Location of agent in object.
*idx_object_*	Location of object in agent.

14.21.2.4 events_run()

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

Executes the stored action.

Parameters

*model_*	Model over which it will be executed.

14.21.2.5 load_agents_entities_ties()

```
template<typename TSeq >
void Model< TSeq >::load_agents_entities_ties (
             std::string fn,
             int skip )  [inline]
```

Associate agents-entities from a file.

The structure of the file should be two columns separated by space. The first column indexing between 0 and nagents-1, and the second column between 0 and nentities - 1.

Parameters

fn	Path to the file.
skip	How many rows to skip.

14.21.2.6 reset()

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

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

Reimplemented in ModelSIRLogit< TSeq >, ModelSIRDCONN< TSeq >, ModelSIRCONN< TSeq >, ModelSEIRDCONN< TSeq >, ModelSEIRCONN< TSeq >, epiworld::epimodels::ModelSIRLogit< TSeq >, epiworld::epimodels::ModelSEIRDCONN< TSeq >, epiworld::epimodels::ModelSIRDCONN< TSeq >, epiworld::epimodels::ModelSEIRCONN< TSeq >, and epiworld::epimodels::ModelSIRCONN< TSeq >.

14.21.2.7 run_multiple()

```
template<typename TSeq >
void Model< TSeq >::run_multiple (
             epiworld_fast_uint ndays,
             epiworld_fast_uint nexperiments,
             int seed_ = -1,
             std::function< void(size_t, Model< TSeq > *)> fun = make_save_run<TSeq>(),
             bool reset = true,
             bool verbose = true,
             int nthreads = 1 )  [inline]
```

Parameters

ndays	Multiple runs of the simulation

14.21.2.8 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.

14.21.2.9 set_name()

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

Set the name object.

Parameters

name	

14.21.2.10 write_data()

```
template<typename TSeq >
void Model< TSeq >::write_data (
            std::string fn_virus_info,
            std::string fn_virus_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,
            std::string fn_generation_time ) const  [inline]
```

Wrapper of `DataBase::write_data`

Parameters

fn_virus_info	Filename. Information about the virus.
fn_virus_hist	Filename. History of the virus.
fn_tool_info	Filename. Information about the tool.
fn_tool_hist	Filename. History of the tool.
fn_total_hist	Filename. Aggregated history (state)
fn_transmission	Filename. Transmission history.
fn_transition	Filename. Markov transition history.
fn_reproductive_number	Filename. Case by case reproductive number

14.21.3 Member Data Documentation

14.21.3.1 initial_states_fun

```
template<typename TSeq >
std::function<void(Model<TSeq>*)> epiworld::Model< TSeq >::initial_states_fun  [protected]
```

Initial value:

```
= [](Model<TSeq> * )
    -> void {}
```

Function to distribute states. Goes along with the function

14.21.3.2 rbinomd

```
template<typename TSeq >
std::binomial_distribution epiworld::Model< TSeq >::rbinomd  [protected]
```

Initial value:

```
=
        std::binomial_distribution<>()
```

14.21.3.3 rexpd

```
template<typename TSeq >
std::exponential_distribution epiworld::Model< TSeq >::rexpd  [protected]
```

Initial value:

```
=
        std::exponential_distribution<>()
```

14.21.3.4 rgammad

```
template<typename TSeq >
std::gamma_distribution epiworld::Model< TSeq >::rgammad  [protected]
```

Initial value:

```
=
        std::gamma_distribution<>()
```

14.21.3.5 rlognormald

```
template<typename TSeq >
std::lognormal_distribution epiworld::Model< TSeq >::rlognormald  [protected]
```

Initial value:

```
=
        std::lognormal_distribution<>()
```

14.21.3.6 rnormd

```
template<typename TSeq >
std::normal_distribution epiworld::Model< TSeq >::rnormd  [protected]
```

Initial value:

```
=
        std::normal_distribution<>(0.0)
```

14.21.3.7 runifd

```
template<typename TSeq >
std::uniform_real_distribution epiworld::Model< TSeq >::runifd  [protected]
```

Initial value:

```
=
        std::uniform_real_distribution<> (0.0, 1.0)
```

14.21.3.8 time_elapsed

```
template<typename TSeq >
std::chrono::duration<epiworld_double,std::micro> epiworld::Model< TSeq >::time_elapsed  [protected]
```

Initial value:

```
=
        std::chrono::duration<epiworld_double,std::micro>::zero()
```

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

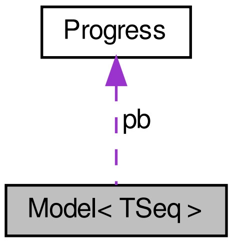
- epiworld.hpp

14.22 Model< TSeq > Class Template Reference

Core class of epiworld.

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

Collaboration diagram for Model< TSeq >:



Public Member Functions

- DataBase< TSeq > & **get_db** ()
- epiworld_double & **operator()** (std::string pname)
- size_t **size** () const
- void load_agents_entities_ties (std::string fn, int skip)

    Associate agents-entities from a file.
- size_t get_n_viruses () const

    Number of viruses in the model.
- size_t get_n_tools () const

    Number of tools in the model.
- epiworld_fast_uint **get_ndays** () const
- epiworld_fast_uint **get_n_replicates** () const
- void **set_ndays** (epiworld_fast_uint ndays)
- bool **get_verbose** () const
- Model< TSeq > & **verbose_off** ()
- Model< TSeq > & **verbose_on** ()
- int today () const

    The current time of the model.
- void write_data (std::string fn_virus_info, std::string fn_virus_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, std::string fn_generation_time) const

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

    Reset the model.
- const Model< TSeq > & **print** (bool lite=false) 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_globalevent (std::function< void(Model< TSeq > *)> fun, std::string name="A global action", int date=-99)

  Set a global action.
- void **add_globalevent** (GlobalEvent< TSeq > action)
- GlobalEvent< TSeq > & get_globalevent (std::string name)

  Retrieve a global action by name.
- GlobalEvent< TSeq > & get_globalevent (size_t i)

  Retrieve a global action by index.
- void rm_globalevent (std::string name)

  Remove a global action by name.
- void rm_globalevent (size_t i)

  Remove a global action by index.
- void **run_globalevents** ()
- void **clear_state_set** ()
- const std::vector< VirusPtr< TSeq > > & **get_viruses** () const
- const std::vector< epiworld_double > & **get_prevalence_virus** () const
- const std::vector< bool > & **get_prevalence_virus_as_proportion** () const
- const std::vector< ToolPtr< TSeq > > & **get_tools** () const
- Virus< TSeq > & **get_virus** (size_t id)
- Tool< TSeq > & **get_tool** (size_t id)
- void set_agents_data (double *data_, size_t ncols_)

  Set the agents data object.
- double * **get_agents_data** ()
- size_t **get_agents_data_ncols** () const
- void set_name (std::string name)

  Set the name object.
- std::string **get_name** () const
- bool **operator==** (const Model< TSeq > &other) const
- bool **operator!=** (const Model< TSeq > &other) const
- void events_run ()

  Executes the stored action.

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** ()

Random number generation

Parameters

eng	*Random number generator*
s	*Seed*

- void **set_rand_engine** (std::mt19937 &eng)
- std::mt19937 & **get_rand_endgine** ()
- void **seed** (size_t 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)
- void **set_rand_binom** (int n, epiworld_double p)
- 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)
- int **rbinom** ()
- int **rbinom** (int n, epiworld_double p)

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, epiworld_fast_uint 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, epiworld_fast_uint preval)
- void **add_tool_fun** (Tool< TSeq > &t, ToolToAgentFun< TSeq > fun)
- void **add_entity** (Entity< TSeq > e)
- void **rm_virus** (size_t virus_pos)
- void **rm_tool** (size_t tool_pos)
- void **rm_entity** (size_t entity_pos)

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_edgelist** (const std::vector< int > &source, const std::vector< int > &target, int size, bool directed)
- void **agents_from_adjlist** (AdjList al)
- bool **is_directed** () const
- std::vector< Agent< TSeq > > & get_agents ()

  Returns a reference to the vector of agents.
- std::vector< epiworld_fast_uint > get_agents_states () const

  Returns a vector with the states of the agents.

- std::vector< Viruses_const< TSeq > > get_agents_viruses () const
  Returns a const vector with the viruses of the agents.
- std::vector< Viruses< TSeq > > get_agents_viruses ()
  Returns a vector with the viruses of the agents.
- std::vector< Entity< TSeq > > & **get_entities** ()
- Model< TSeq > & **agents_smallworld** (epiworld_fast_uint n=1000, epiworld_fast_uint k=5, bool d=false, epiworld_double p=.01)
- void **agents_empty_graph** (epiworld_fast_uint 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 **update_state** ()
- void **mutate_virus** ()
- void **next** ()
- virtual Model< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)
  Runs the simulation (after initialization)
- void run_multiple (epiworld_fast_uint ndays, epiworld_fast_uint nexperiments, int seed_=-1, std::function< void(size_t, Model< TSeq > *)> fun=make_save_run< TSeq >(), bool reset=true, bool verbose=true, int nthreads=1)

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< int > &source, std::vector< int > &target) const

Manage state (states) in the model

The functions `get_state` return the current values for the states included in the model.

Parameters

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

Returns

`add_state*` returns nothing.

`get_state_*` returns a vector of pairs with the states and their labels.

- void **add_state** (std::string lab, UpdateFun< TSeq > fun=nullptr)
- const std::vector< std::string > & **get_states** () const
- const std::vector< UpdateFun< TSeq > > & **get_state_fun** () const
- void **print_state_codes** () const

Initial states

These functions are called before the simulation starts.

Parameters

proportions_	*Vector of proportions for each state.*
queue_	*Vector of queue for each state.*

- virtual Model< TSeq > & **initial_states** (std::vector< double >, std::vector< int >)

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 `epiworld_fast_uint` 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 **add_param** (epiworld_double initial_val, std::string pname)
- void **read_params** (std::string fn)
- epiworld_double **get_param** (epiworld_fast_uint k)
- epiworld_double **get_param** (std::string pname)
- void **set_param** (std::string pname, epiworld_double val)
- epiworld_double **par** (std::string pname)

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** (epiworld_fast_uint 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.)
- Model< TSeq > & 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)

Protected Member Functions

- void **dist_tools** ()
- void **dist_virus** ()

- void **chrono_start** ()
- void **chrono_end** ()
- void events_add (Agent< TSeq > *agent_, VirusPtr< TSeq > virus_, ToolPtr< TSeq > tool_, Entity< TSeq > *entity_, epiworld_fast_int new_state_, epiworld_fast_int queue_, ActionFun< TSeq > call_, int idx_↩ agent_, int idx_object_)

  Construct a new Event object.

Protected Attributes

- std::string name = ""

  Name of the model.
- DataBase< TSeq > **db** = DataBase<TSeq>(*this)
- std::vector< Agent< TSeq > > **population** = {}
- bool **using_backup** = true
- std::vector< Agent< TSeq > > **population_backup** = {}
- bool **directed** = false
- std::vector< VirusPtr< TSeq > > **viruses** = {}
- std::vector< epiworld_double > prevalence_virus = {}

  Initial prevalence_virus of each virus.
- std::vector< bool > **prevalence_virus_as_proportion** = {}
- std::vector< VirusToAgentFun< TSeq > > **viruses_dist_funs** = {}
- std::vector< ToolPtr< TSeq > > **tools** = {}
- std::vector< epiworld_double > **prevalence_tool** = {}
- std::vector< bool > **prevalence_tool_as_proportion** = {}
- std::vector< ToolToAgentFun< TSeq > > **tools_dist_funs** = {}
- std::vector< Entity< TSeq > > **entities** = {}
- std::vector< Entity< TSeq > > **entities_backup** = {}
- std::mt19937 **engine**
- std::uniform_real_distribution **runifd**
- std::normal_distribution **rnormd**
- std::gamma_distribution **rgammad**
- std::lognormal_distribution **rlognormald**
- std::exponential_distribution **rexpd**
- std::binomial_distribution **rbinomd**
- std::function< void(std::vector< Agent< TSeq >> *, Model< TSeq > *, epiworld_double)> **rewire_fun**
- epiworld_double **rewire_prop** = 0.0
- std::map< std::string, epiworld_double > **parameters**
- epiworld_fast_uint **ndays** = 0
- Progress **pb**
- std::vector< UpdateFun< TSeq > > state_fun = {}

  Functions to update states.
- std::vector< std::string > states_labels = {}

  Labels of the states.
- std::function< void(Model< TSeq > *)> initial_states_fun
- epiworld_fast_uint **nstates** = 0u
- bool **verbose** = true
- int **current_date** = 0
- std::chrono::time_point< std::chrono::steady_clock > **time_start**
- std::chrono::time_point< std::chrono::steady_clock > **time_end**
- std::chrono::duration< epiworld_double, std::micro > **time_elapsed**
- epiworld_fast_uint **n_replicates** = 0u
- std::vector< GlobalEvent< TSeq > > **globalevents**
- Queue< TSeq > **queue**

- bool **use_queuing** = true
- std::vector< Event< TSeq > > events = {}

  Variables used to keep track of the events to be made regarding viruses.
- epiworld_fast_uint **nactions** = 0u

Auxiliary variables for AgentsSample<TSeq> iterators

These variables+objects are used by the AgentsSample<TSeq> class for building efficient iterators over agents. The idea is to reduce the memory allocation, so only during the first call of AgentsSample<TSeq>::AgentsSample(Model<TSeq>) these vectors are allocated.

- std::vector< Agent< TSeq > * > **sampled_population**
- size_t **sampled_population_n** = 0u
- std::vector< size_t > **population_left**
- size_t **population_left_n** = 0u

Agents features

Optionally, a model can include an external data source pointing to agents information. The data can then be access through the `Agent::operator()` method.

- double * **agents_data** = nullptr
- size_t **agents_data_ncols** = 0u

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.

- MixerFun< TSeq > **susceptibility_reduction_mixer** = susceptibility_reduction_mixer_default<TSeq>
- MixerFun< TSeq > **transmission_reduction_mixer** = transmission_reduction_mixer_default<TSeq>
- MixerFun< TSeq > **recovery_enhancer_mixer** = recovery_enhancer_mixer_default<TSeq>
- MixerFun< TSeq > **death_reduction_mixer** = death_reduction_mixer_default<TSeq>
- std::vector< epiworld_double > **array_double_tmp**
- std::vector< Virus< TSeq > * > **array_virus_tmp**
- virtual Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- **Model** ()
- **Model** (const Model< TSeq > &m)
- **Model** (Model< TSeq > &m)
- **Model** (Model< TSeq > &&m)
- Model< TSeq > & **operator=** (const Model< TSeq > &m)
- virtual ~**Model** ()
- void **clone_population** (std::vector< Agent< TSeq > > &other_population, std::vector< Entity< TSeq > > &other_entities, Model< TSeq > *other_model, bool &other_directed) const
- void **clone_population** (const Model< TSeq > &other_model)

14.22.1 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.)

14.22.2 Member Function Documentation

14.22.2.1 add_globalevent()

```
template<typename TSeq >
void Model< TSeq >::add_globalevent (
             std::function< void(Model< TSeq > *)> fun,
             std::string name = "A global action",
             int date = -99 )  [inline]
```

Set a global action.

Parameters

fun	A function to be called on the prescribed date
name	Name of the action.
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.

14.22.2.2 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * Model< TSeq >::clone_ptr  [inline], [protected], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

14.22.2.3 events_add()

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

Construct a new Event object.

Parameters

*agent_*	Agent over which the action will be called
*virus_*	Virus pointer included in the action
*tool_*	Tool pointer included in the action
*entity_*	Entity pointer included in the action
*new_state_*	New state of the agent
*call_*	Function the action will call
*queue_*	Change in the queue
*idx_agent_*	Location of agent in object.
*idx_object_*	Location of object in agent.

14.22.2.4 events_run()

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

Executes the stored action.

Parameters

*model_*	Model over which it will be executed.

14.22.2.5 load_agents_entities_ties()

```
template<typename TSeq >
void Model< TSeq >::load_agents_entities_ties (
             std::string fn,
             int skip )  [inline]
```

Associate agents-entities from a file.

The structure of the file should be two columns separated by space. The first column indexing between 0 and nagents-1, and the second column between 0 and nentities - 1.

Parameters

fn	Path to the file.
skip	How many rows to skip.

14.22.2.6 reset()

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

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

14.22.2.7 run_multiple()

```
template<typename TSeq >
void Model< TSeq >::run_multiple (
             epiworld_fast_uint ndays,
             epiworld_fast_uint nexperiments,
             int seed_ = -1,
             std::function< void(size_t, Model< TSeq > *)> fun = make_save_run<TSeq>(),
             bool reset = true,
             bool verbose = true,
             int nthreads = 1 )  [inline]
```

Parameters

ndays	Multiple runs of the simulation

14.22.2.8 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.

14.22.2.9 set_name()

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

Set the name object.

Parameters

name	

14.22.2.10 write_data()

```
template<typename TSeq >
void Model< TSeq >::write_data (
            std::string fn_virus_info,
            std::string fn_virus_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,
            std::string fn_generation_time ) const  [inline]
```

Wrapper of `DataBase::write_data`

Parameters

fn_virus_info	Filename. Information about the virus.
fn_virus_hist	Filename. History of the virus.
fn_tool_info	Filename. Information about the tool.
fn_tool_hist	Filename. History of the tool.
fn_total_hist	Filename. Aggregated history (state)
fn_transmission	Filename. Transmission history.
fn_transition	Filename. Markov transition history.
fn_reproductive_number	Filename. Case by case reproductive number

14.22.3 Member Data Documentation

14.22.3.1 initial_states_fun

```
template<typename TSeq >
std::function<void(Model<TSeq>*)> Model< TSeq >::initial_states_fun  [protected]
```

Initial value:

```
= [](Model<TSeq> * )
    -> void {}
```

Function to distribute states. Goes along with the function

14.22.3.2 rbinomd

```
template<typename TSeq >
std::binomial_distribution Model< TSeq >::rbinomd  [protected]
```

Initial value:

```
=
        std::binomial_distribution<>()
```

14.22.3.3 rexpd

```
template<typename TSeq >
std::exponential_distribution Model< TSeq >::rexpd  [protected]
```

Initial value:

```
=
        std::exponential_distribution<>()
```

14.22.3.4 rgammad

```
template<typename TSeq >
std::gamma_distribution Model< TSeq >::rgammad  [protected]
```

Initial value:

```
=
        std::gamma_distribution<>()
```

14.22.3.5 rlognormald

```
template<typename TSeq >
std::lognormal_distribution Model< TSeq >::rlognormald  [protected]
```

Initial value:

```
=
        std::lognormal_distribution<>()
```

14.22.3.6 rnormd

```
template<typename TSeq >
std::normal_distribution Model< TSeq >::rnormd  [protected]
```

Initial value:

```
=
        std::normal_distribution<>(0.0)
```

14.22.3.7 runifd

```
template<typename TSeq >
std::uniform_real_distribution Model< TSeq >::runifd  [protected]
```

Initial value:

```
=
        std::uniform_real_distribution<> (0.0, 1.0)
```

14.22.3.8 time_elapsed

```
template<typename TSeq >
std::chrono::duration<epiworld_double,std::micro> Model< TSeq >::time_elapsed  [protected]
```

Initial value:

```
=
        std::chrono::duration<epiworld_double,std::micro>::zero()
```

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

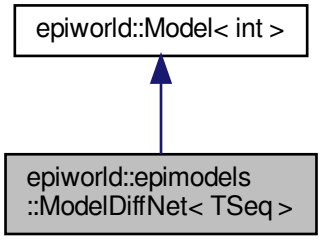
- include/epiworld/agent-bones.hpp
- include/epiworld/model-bones.hpp
- include/epiworld/model-meat-print.hpp
- include/epiworld/model-meat.hpp

14.23 epiworld::epimodels::ModelDiffNet< TSeq > Class Template Reference

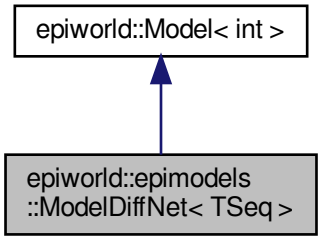
Template for a Network Diffusion Model.

`#include <epiworld.hpp>`

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



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



Public Member Functions

- **ModelDiffNet** (ModelDiffNet< TSeq > &model, std::string innovation_name, epiworld_double prevalence, epiworld_double prob_adopt, bool normalize_exposure=true, double *agents_data=nullptr, size_t data_↩ ncols=0u, std::vector< size_t > data_cols={}, std::vector< double > params={})
- **ModelDiffNet** (std::string innovation_name, epiworld_double prevalence, epiworld_double prob_adopt, bool normalize_exposure=true, double *agents_data=nullptr, size_t data_ncols=0u, std::vector< size_t > data↩ _cols={}, std::vector< double > params={})

Public Attributes

- bool **normalize_exposure** = true
- std::vector< size_t > **data_cols**
- std::vector< double > **params**

Static Public Attributes

- static const int **NONADOPTER** = 0
- static const int **ADOPTER** = 1

Additional Inherited Members

14.23.1 Detailed Description

template<typename TSeq = int>
class epiworld::epimodels::ModelDiffNet< TSeq >

Template for a Network Diffusion 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:

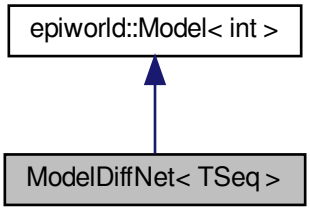
- epiworld.hpp

14.24 ModelDiffNet< TSeq > Class Template Reference

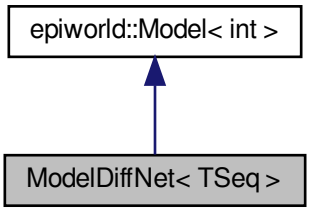
Template for a Network Diffusion Model.

`#include <diffnet.hpp>`

Inheritance diagram for ModelDiffNet< TSeq >:



Collaboration diagram for ModelDiffNet< TSeq >:



Public Member Functions

- **ModelDiffNet** (ModelDiffNet< TSeq > &model, std::string innovation_name, epiworld_double prevalence, epiworld_double prob_adopt, bool normalize_exposure=true, double *agents_data=nullptr, size_t data_↩ ncols=0u, std::vector< size_t > data_cols={}, std::vector< double > params={})
- **ModelDiffNet** (std::string innovation_name, epiworld_double prevalence, epiworld_double prob_adopt, bool normalize_exposure=true, double *agents_data=nullptr, size_t data_ncols=0u, std::vector< size_t > data↩ _cols={}, std::vector< double > params={})

Public Attributes

- bool **normalize_exposure** = true
- std::vector< size_t > **data_cols**
- std::vector< double > **params**

Static Public Attributes

- static const int **NONADOPTER** = 0
- static const int **ADOPTER** = 1

Additional Inherited Members

14.24.1 Detailed Description

template<typename TSeq = int>
class ModelDiffNet< TSeq >

Template for a Network Diffusion 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:

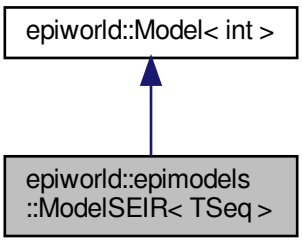
- include/epiworld/models/diffnet.hpp

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

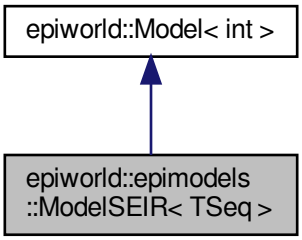
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

`#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 transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
- **ModelSEIR** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld↩ _double avg_incubation_days, epiworld_double recovery_rate)
- ModelSEIR< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set up the initial states of the model.

Public Attributes

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

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3

Additional Inherited Members

14.25.1 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
prevalence	epiworld_double Initial prevalence the immune system
transmission_rate	epiworld_double Transmission rate of the virus
avg_incubation_days	epiworld_double Average incubation days of the virus.
recovery_rate	epiworld_double Recovery rate of the virus.

14.25.2 Member Function Documentation

14.25.2.1 initial_states()

```
template<typename TSeq >
ModelSEIR< TSeq > & ModelSEIR< TSeq >::initial_states (
             std::vector< double > proportions_,
             std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set up the initial states of the model.

Parameters

*proportions_*	Double vector with the following values: • 0: Proportion of non-infected agents who are removed. • 1: Proportion of exposed agents to be set as infected.

Reimplemented from epiworld::Model< int >.

14.25.3 Member Data Documentation

14.25.3.1 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 {

        auto v = p->get_virus();

        if (m->runif() < 1.0/(v->get_incubation(m)))
            p->change_state(m, ModelSEIR<TSeq>::INFECTED);
        return;
    }
```

14.25.3.2 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->par("Recovery rate")))
            p->rm_virus(m);
        return;
    }
```

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

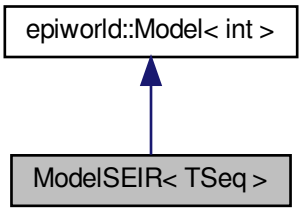
- epiworld.hpp

14.26 ModelSEIR< TSeq > Class Template Reference

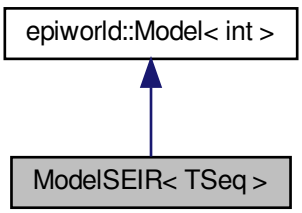
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.

`#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 transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
- **ModelSEIR** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld↩ _double avg_incubation_days, epiworld_double recovery_rate)
- ModelSEIR< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set up the initial states of the model.

Public Attributes

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

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3

Additional Inherited Members

14.26.1 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
prevalence	epiworld_double Initial prevalence the immune system
transmission_rate	epiworld_double Transmission rate of the virus
avg_incubation_days	epiworld_double Average incubation days of the virus.
recovery_rate	epiworld_double Recovery rate of the virus.

14.26.2 Member Function Documentation

14.26.2.1 initial_states()

```
template<typename TSeq >
ModelSEIR< TSeq > & ModelSEIR< TSeq >::initial_states (
```

```
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set up the initial states of the model.

Parameters

*proportions_*	Double vector with the following values: • 0: Proportion of non-infected agents who are removed. • 1: Proportion of exposed agents to be set as infected.

Reimplemented from epiworld::Model< int >.

14.26.3 Member Data Documentation

14.26.3.1 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 {

        auto v = p->get_virus();

        if (m->runif() < 1.0/(v->get_incubation(m)))
            p->change_state(m, ModelSEIR<TSeq>::INFECTED);
        return;
    }
```

14.26.3.2 update_infected_seir

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

Initial value:

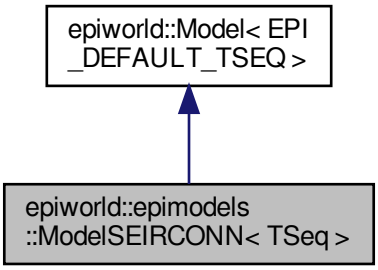
```
= [](
        epiworld::Agent<TSeq> * p,
        epiworld::Model<TSeq> * m
    ) -> void {

        if (m->runif() < (m->par("Recovery rate")))
            p->rm_virus(m);
        return;
    }
```

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

- include/epiworld/models/seir.hpp

14.27 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, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)

  Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRCONN** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
- ModelSEIRCONN< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void reset ()

  Reset the model.

- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- ModelSEIRCONN< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue↩ _={})

  Set the initial states of the model.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **RECOVERED** = 3

Additional Inherited Members

14.27.1 Constructor & Destructor Documentation

14.27.1.1 ModelSEIRCONN()

```
template<typename TSeq >
ModelSEIRCONN< TSeq >::ModelSEIRCONN (
            ModelSEIRCONN< TSeq > & model,
            std::string vname,
            epiworld_fast_uint n,
            epiworld_double prevalence,
            epiworld_double contact_rate,
            epiworld_double transmission_rate,
            epiworld_double avg_incubation_days,
            epiworld_double recovery_rate )  [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)
contact_rate	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery

14.27.2 Member Function Documentation

14.27.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRCONN< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.27.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRCONN< TSeq > & ModelSEIRCONN< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set the initial states of the model.

Parameters

*proportions_*	Double vector with a single element: • The proportion of non-infected individuals who have recovered.

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.27.2.3 reset()

```
template<typename TSeq >
void ModelSEIRCONN< TSeq >::reset  [inline], [virtual]
```

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

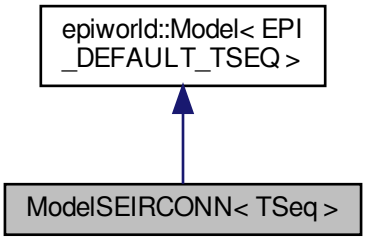
Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

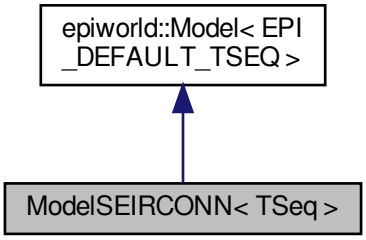
- epiworld.hpp

14.28 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, epiworld_fast_uint n, epiworld↩ _double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)

  Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRCONN** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate)
- ModelSEIRCONN< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void reset ()

  Reset the model.
- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- ModelSEIRCONN< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue↩ _={})

  Set the initial states of the model.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **RECOVERED** = 3

Additional Inherited Members

14.28.1 Constructor & Destructor Documentation

14.28.1.1 ModelSEIRCONN()

```
template<typename TSeq >
ModelSEIRCONN< TSeq >::ModelSEIRCONN (
            ModelSEIRCONN< TSeq > & model,
            std::string vname,
            epiworld_fast_uint n,
            epiworld_double prevalence,
            epiworld_double contact_rate,
            epiworld_double transmission_rate,
            epiworld_double avg_incubation_days,
            epiworld_double recovery_rate )  [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)
contact_rate	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery

14.28.2 Member Function Documentation

14.28.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRCONN< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.28.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRCONN< TSeq > & ModelSEIRCONN< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set the initial states of the model.

Parameters

*proportions_*	Double vector with a single element: • The proportion of non-infected individuals who have recovered.

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.28.2.3 reset()

```
template<typename TSeq >
void ModelSEIRCONN< TSeq >::reset  [inline], [virtual]
```

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

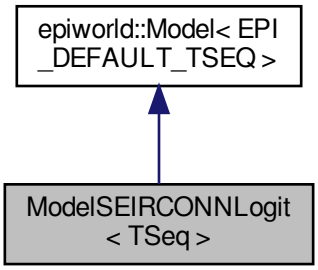
Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

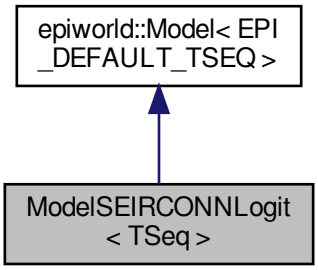
- include/epiworld/models/seirconnected.hpp

14.29 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, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, double *covars, std::vector< double > logit_params)

  Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRCONNLogit** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate 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

Additional Inherited Members

14.29.1 Constructor & Destructor Documentation

14.29.1.1 ModelSEIRCONNLogit()

```
template<typename TSeq >
ModelSEIRCONNLogit< TSeq >::ModelSEIRCONNLogit (
             ModelSEIRCONNLogit< TSeq > & model,
             std::string vname,
             epiworld_fast_uint n,
             epiworld_double prevalence,
             epiworld_double contact_rate,
             epiworld_double transmission_rate,
             epiworld_double avg_incubation_days,
             epiworld_double recovery_rate,
             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)
contact_rate	Reproductive number (beta)
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery

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

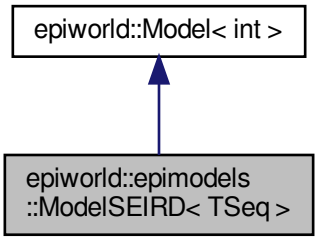
- include/epiworld/models/seirconnected_logit.hpp

14.30 epiworld::epimodels::ModelSEIRD< TSeq > Class Template Reference

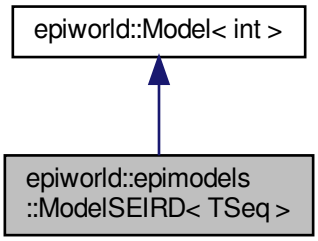
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

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

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



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



Public Member Functions

- ModelSEIRD (ModelSEIRD< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_↩ double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld↩ _double death_rate)

  Constructor for the SEIRD model.
- ModelSEIRD (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld↩ _double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)

  Constructor for the SEIRD model.
- ModelSEIRD< TSeq > & **initial_states** (std::vector< double > proportions_, std::vector< int > queue_={})

Public Attributes

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

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3
- static const int **DECEASED** = 4

Additional Inherited Members

14.30.1 Detailed Description

template<typename TSeq = int>
class epiworld::epimodels::ModelSEIRD< TSeq >

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

14.30.2 Constructor & Destructor Documentation

14.30.2.1 ModelSEIRD() [1/2]

```
template<typename TSeq >
ModelSEIRD< TSeq >::ModelSEIRD (
            ModelSEIRD< TSeq > & model,
            std::string vname,
            epiworld_double prevalence,
            epiworld_double transmission_rate,
            epiworld_double avg_incubation_days,
            epiworld_double recovery_rate,
            epiworld_double death_rate )  [inline]
```

Constructor for the SEIRD model.

Template Parameters

TSeq	Type of the sequence used in the model.

Parameters

model	Reference to the SEIRD model.
vname	Name of the model.
prevalence	Prevalence of the disease.
transmission_rate	Transmission rate of the disease.
avg_incubation_days	Average incubation period of the disease.
recovery_rate	Recovery rate of the disease.
death_rate	Death rate of the disease.

14.30.2.2 ModelSEIRD() [2/2]

```
template<typename TSeq >
ModelSEIRD< TSeq >::ModelSEIRD (
             std::string vname,
             epiworld_double prevalence,
             epiworld_double transmission_rate,
             epiworld_double avg_incubation_days,
             epiworld_double recovery_rate,
             epiworld_double death_rate )  [inline]
```

Constructor for the SEIRD model.

Parameters

vname	Name of the model.
prevalence	Initial prevalence of the disease.
transmission_rate	Transmission rate of the disease.
avg_incubation_days	Average incubation period of the disease.
recovery_rate	Recovery rate of the disease.
death_rate	Death rate of the disease.

14.30.3 Member Data Documentation

14.30.3.1 update_exposed_seir

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

Initial value:

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

    auto v = p->get_virus();

    if (m->runif() < 1.0/(v->get_incubation(m)))
      p->change_state(m, ModelSEIRD<TSeq>::INFECTED);
    return;
  }
```

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

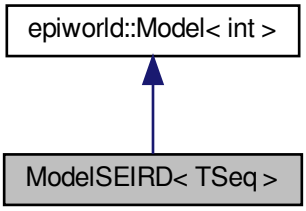
- epiworld.hpp

14.31 ModelSEIRD< TSeq > Class Template Reference

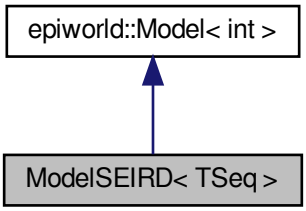
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model.

`#include <seird.hpp>`

Inheritance diagram for ModelSEIRD< TSeq >:



Collaboration diagram for ModelSEIRD< TSeq >:



Public Member Functions

- ModelSEIRD (ModelSEIRD< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_↩ double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld↩ _double death_rate)

  Constructor for the SEIRD model.
- ModelSEIRD (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld↩ _double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)

  Constructor for the SEIRD model.
- ModelSEIRD< TSeq > & **initial_states** (std::vector< double > proportions_, std::vector< int > queue_={})

Public Attributes

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

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3
- static const int **DECEASED** = 4

Additional Inherited Members

14.31.1 Detailed Description

template<typename TSeq = int>
class ModelSEIRD< TSeq >

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

14.31.2 Constructor & Destructor Documentation

14.31.2.1 ModelSEIRD() [1/2]

```
template<typename TSeq >
ModelSEIRD< TSeq >::ModelSEIRD (
            ModelSEIRD< TSeq > & model,
            std::string vname,
            epiworld_double prevalence,
            epiworld_double transmission_rate,
            epiworld_double avg_incubation_days,
            epiworld_double recovery_rate,
            epiworld_double death_rate )  [inline]
```

Constructor for the SEIRD model.

Template Parameters

TSeq	Type of the sequence used in the model.

Parameters

model	Reference to the SEIRD model.
vname	Name of the model.
prevalence	Prevalence of the disease.
transmission_rate	Transmission rate of the disease.
avg_incubation_days	Average incubation period of the disease.
recovery_rate	Recovery rate of the disease.
death_rate	Death rate of the disease.

14.31.2.2 ModelSEIRD() [2/2]

```
template<typename TSeq >
ModelSEIRD< TSeq >::ModelSEIRD (
             std::string vname,
             epiworld_double prevalence,
             epiworld_double transmission_rate,
             epiworld_double avg_incubation_days,
             epiworld_double recovery_rate,
             epiworld_double death_rate )  [inline]
```

Constructor for the SEIRD model.

Parameters

vname	Name of the model.
prevalence	Initial prevalence of the disease.
transmission_rate	Transmission rate of the disease.
avg_incubation_days	Average incubation period of the disease.
recovery_rate	Recovery rate of the disease.
death_rate	Death rate of the disease.

14.31.3 Member Data Documentation

14.31.3.1 update_exposed_seir

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

Initial value:

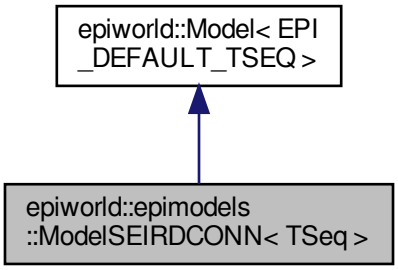
```
= [](
    epiworld::Agent<TSeq> * p,
    epiworld::Model<TSeq> * m
  ) -> void {

    auto v = p->get_virus();

    if (m->runif() < 1.0/(v->get_incubation(m)))
      p->change_state(m, ModelSEIRD<TSeq>::INFECTED);
    return;
  }
```

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

- include/epiworld/models/seird.hpp

14.32 epiworld::epimodels::ModelSEIRDCONN< TSeq > Class Template Reference

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



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

Public Member Functions

- ModelSEIRDCONN (ModelSEIRDCONN< TSeq > &model, std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_↩ double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)

  Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRDCONN** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
- ModelSEIRDCONN< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void reset ()

  Reset the model.

- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- ModelSEIRDCONN< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set up the initial states of the model.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3
- static const int **DECEASED** = 4

Additional Inherited Members

14.32.1 Constructor & Destructor Documentation

14.32.1.1 ModelSEIRDCONN()

```
template<typename TSeq >
ModelSEIRDCONN< TSeq >::ModelSEIRDCONN (
            ModelSEIRDCONN< TSeq > & model,
            std::string vname,
            epiworld_fast_uint n,
            epiworld_double prevalence,
            epiworld_double contact_rate,
            epiworld_double transmission_rate,
            epiworld_double avg_incubation_days,
            epiworld_double recovery_rate,
            epiworld_double death_rate )  [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)
contact_rate	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery
death_rate	Probability of death

14.32.2 Member Function Documentation

14.32.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRDCONN< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.32.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRDCONN< TSeq > & ModelSEIRDCONN< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set up the initial states of the model.

Parameters

*proportions_*	Double vector with the following values: • 0: Proportion of non-infected agents who are removed. • 1: Proportion of exposed agents to be set as infected.

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.32.2.3 reset()

```
template<typename TSeq >
void ModelSEIRDCONN< TSeq >::reset  [inline], [virtual]
```

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

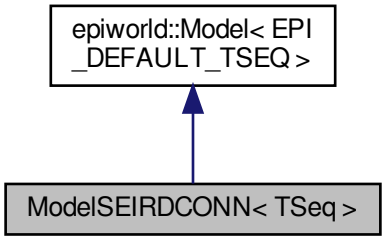
Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

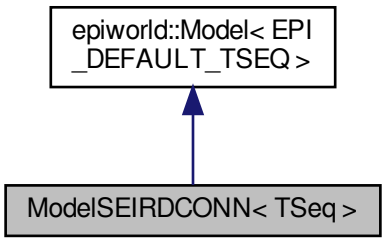
- epiworld.hpp

14.33 ModelSEIRDCONN< TSeq > Class Template Reference

Inheritance diagram for ModelSEIRDCONN< TSeq >:



Collaboration diagram for ModelSEIRDCONN< TSeq >:



Public Member Functions

- ModelSEIRDCONN (ModelSEIRDCONN< TSeq > &model, std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_↩ double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)

  Template for a Susceptible-Exposed-Infected-Removed (SEIR) model.
- **ModelSEIRDCONN** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double avg_incubation_days, epiworld_double recovery_rate, epiworld_double death_rate)
- ModelSEIRDCONN< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void reset ()

  Reset the model.
- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- ModelSEIRDCONN< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set up the initial states of the model.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **EXPOSED** = 1
- static const int **INFECTED** = 2
- static const int **REMOVED** = 3
- static const int **DECEASED** = 4

Additional Inherited Members

14.33.1 Constructor & Destructor Documentation

14.33.1.1 ModelSEIRDCONN()

```
template<typename TSeq >
ModelSEIRDCONN< TSeq >::ModelSEIRDCONN (
            ModelSEIRDCONN< TSeq > & model,
            std::string vname,
            epiworld_fast_uint n,
            epiworld_double prevalence,
            epiworld_double contact_rate,
            epiworld_double transmission_rate,
            epiworld_double avg_incubation_days,
            epiworld_double recovery_rate,
            epiworld_double death_rate )  [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)
contact_rate	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery
death_rate	Probability of death

14.33.2 Member Function Documentation

14.33.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSEIRDCONN< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.33.2.2 initial_states()

```
template<typename TSeq >
ModelSEIRDCONN< TSeq > & ModelSEIRDCONN< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set up the initial states of the model.

Parameters

*proportions_*	Double vector with the following values: • 0: Proportion of non-infected agents who are removed. • 1: Proportion of exposed agents to be set as infected.

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.33.2.3 reset()

```
template<typename TSeq >
void ModelSEIRDCONN< TSeq >::reset  [inline], [virtual]
```

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

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

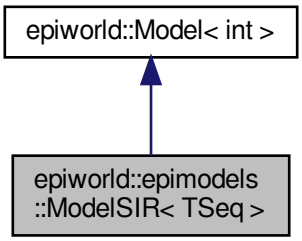
- include/epiworld/models/seirdconnected.hpp

14.34 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference

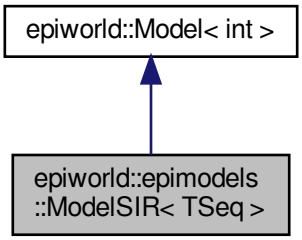
Template for a Susceptible-Infected-Removed (SIR) model.

```
#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 transmission_rate, epiworld_double recovery_rate)
- **ModelSIR** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_↩ double recovery_rate)
- ModelSIR< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set the initial states of the model.

Additional Inherited Members

14.34.1 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 rate of the immune system

14.34.2 Member Function Documentation

14.34.2.1 initial_states()

```
template<typename TSeq >
ModelSIR< TSeq > & ModelSIR< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set the initial states of the model.

Parameters

*proportions_*	Double vector with a single element: • The proportion of non-infected individuals who have recovered.

Reimplemented from epiworld::Model< int >.

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

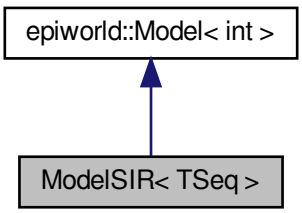
- epiworld.hpp

14.35 ModelSIR< TSeq > Class Template Reference

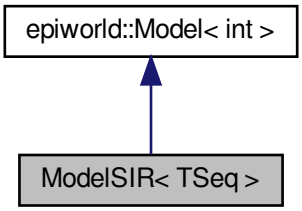
Template for a Susceptible-Infected-Removed (SIR) model.

`#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 transmission_rate, epiworld_double recovery_rate)
- **ModelSIR** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_↩ double recovery_rate)
- ModelSIR< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set the initial states of the model.

Additional Inherited Members

14.35.1 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 rate of the immune system

14.35.2 Member Function Documentation

14.35.2.1 initial_states()

```
template<typename TSeq >
ModelSIR< TSeq > & ModelSIR< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set the initial states of the model.

Parameters

*proportions_*	Double vector with a single element: • The proportion of non-infected individuals who have recovered.

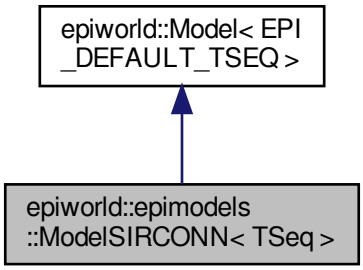
Reimplemented from epiworld::Model< int >.

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

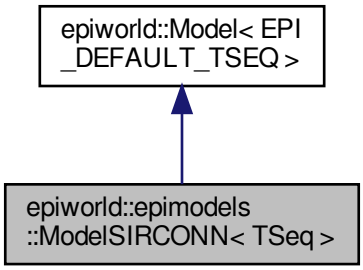
- include/epiworld/models/sir.hpp

14.36 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, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate)

  Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRCONN** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate)
- ModelSIRCONN< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void reset ()

  Reset the model.
- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- ModelSIRCONN< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set the initial states of the model.

Additional Inherited Members

14.36.1 Constructor & Destructor Documentation

14.36.1.1 ModelSIRCONN()

```
template<typename TSeq >
ModelSIRCONN< TSeq >::ModelSIRCONN (
            ModelSIRCONN< TSeq > & model,
            std::string vname,
            epiworld_fast_uint n,
            epiworld_double prevalence,
            epiworld_double contact_rate,
            epiworld_double transmission_rate,
            epiworld_double recovery_rate )  [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)
contact_rate	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery

14.36.2 Member Function Documentation

14.36.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRCONN< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.36.2.2 initial_states()

```
template<typename TSeq >
ModelSIRCONN< TSeq > & ModelSIRCONN< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set the initial states of the model.

Parameters

*proportions_*	Double vector with a single element: • The proportion of non-infected individuals who have recovered.

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.36.2.3 reset()

```
template<typename TSeq >
void ModelSIRCONN< TSeq >::reset  [inline], [virtual]
```

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

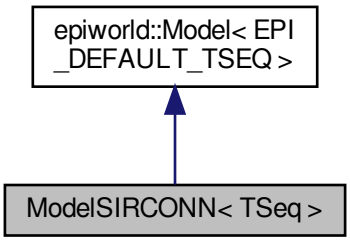
Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

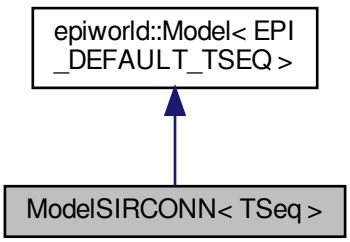
- epiworld.hpp

14.37 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, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate)

  Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRCONN** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate)
- ModelSIRCONN< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void reset ()

  Reset the model.
- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- ModelSIRCONN< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set the initial states of the model.

Additional Inherited Members

14.37.1 Constructor & Destructor Documentation

14.37.1.1 ModelSIRCONN()

```
template<typename TSeq >
ModelSIRCONN< TSeq >::ModelSIRCONN (
            ModelSIRCONN< TSeq > & model,
            std::string vname,
            epiworld_fast_uint n,
            epiworld_double prevalence,
            epiworld_double contact_rate,
            epiworld_double transmission_rate,
            epiworld_double recovery_rate )  [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)
contact_rate	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery

14.37.2 Member Function Documentation

14.37.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRCONN< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.37.2.2 initial_states()

```
template<typename TSeq >
ModelSIRCONN< TSeq > & ModelSIRCONN< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set the initial states of the model.

Parameters

*proportions_*	Double vector with a single element: • The proportion of non-infected individuals who have recovered.

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.37.2.3 reset()

```
template<typename TSeq >
void ModelSIRCONN< TSeq >::reset  [inline], [virtual]
```

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

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

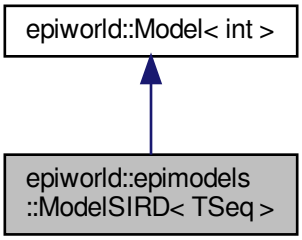
- include/epiworld/models/sirconnected.hpp

14.38 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference

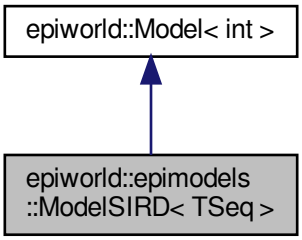
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

`#include <epiworld.hpp>`

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



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



Public Member Functions

- ModelSIRD< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})

  Set the initial states of the model.

- ModelSIRD (ModelSIRD< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

  Constructs a new SIRD model with the given parameters.

- **ModelSIRD** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

Additional Inherited Members

14.38.1 Detailed Description

template<typename TSeq = int>
class epiworld::epimodels::ModelSIRD< TSeq >

Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

14.38.2 Constructor & Destructor Documentation

14.38.2.1 ModelSIRD()

```
template<typename TSeq >
ModelSIRD< TSeq >::ModelSIRD (
            ModelSIRD< TSeq > & model,
            std::string vname,
            epiworld_double prevalence,
            epiworld_double transmission_rate,
            epiworld_double recovery_rate,
            epiworld_double death_rate ) [inline]
```

Constructs a new SIRD model with the given parameters.

Parameters

model	The SIRD model to copy from.
vname	The name of the vertex associated with this model.
prevalence	The initial prevalence of the disease in the population.
transmission_rate	The rate at which the disease spreads from infected to susceptible individuals.
recovery_rate	The rate at which infected individuals recover and become immune.
death_rate	The rate at which infected individuals die.

14.38.3 Member Function Documentation

14.38.3.1 initial_states()

```
template<typename TSeq >
ModelSIRD< TSeq > & ModelSIRD< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set the initial states of the model.

Parameters

*proportions_*	Double vector with two elements: • The proportion of non-infected individuals who have recovered. • The proportion of non-infected individuals who have died.

Reimplemented from epiworld::Model< int >.

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

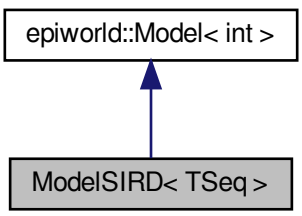
- epiworld.hpp

14.39 ModelSIRD< TSeq > Class Template Reference

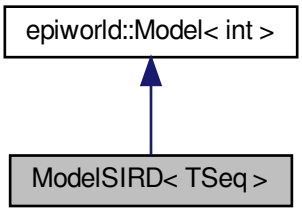
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

`#include <sird.hpp>`

Inheritance diagram for ModelSIRD< TSeq >:



Collaboration diagram for ModelSIRD< TSeq >:



Public Member Functions

- ModelSIRD< TSeq > & initial_states (std::vector< double > proportions_, std::vector< int > queue_={})
  Set the initial states of the model.

- ModelSIRD (ModelSIRD< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
  Constructs a new SIRD model with the given parameters.
- **ModelSIRD** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

Additional Inherited Members

14.39.1 Detailed Description

template<typename TSeq = int>
class ModelSIRD< TSeq >

Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

14.39.2 Constructor & Destructor Documentation

14.39.2.1 ModelSIRD()

```
template<typename TSeq >
ModelSIRD< TSeq >::ModelSIRD (
            ModelSIRD< TSeq > & model,
            std::string vname,
            epiworld_double prevalence,
            epiworld_double transmission_rate,
            epiworld_double recovery_rate,
            epiworld_double death_rate )  [inline]
```

Constructs a new SIRD model with the given parameters.

Parameters

model	The SIRD model to copy from.
vname	The name of the vertex associated with this model.
prevalence	The initial prevalence of the disease in the population.
transmission_rate	The rate at which the disease spreads from infected to susceptible individuals.
recovery_rate	The rate at which infected individuals recover and become immune.
death_rate	The rate at which infected individuals die.

14.39.3 Member Function Documentation

14.39.3.1 initial_states()

```
template<typename TSeq >
ModelSIRD< TSeq > & ModelSIRD< TSeq >::initial_states (
            std::vector< double > proportions_,
            std::vector< int > queue_ = {} )  [inline], [virtual]
```

Set the initial states of the model.

Parameters

*proportions_*	Double vector with two elements: • The proportion of non-infected individuals who have recovered. • The proportion of non-infected individuals who have died.

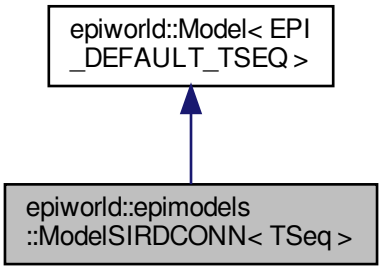
Reimplemented from epiworld::Model< int >.

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

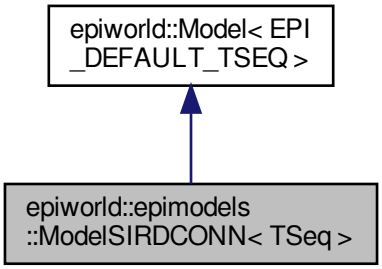
- include/epiworld/models/sird.hpp

14.40 epiworld::epimodels::ModelSIRDCONN< TSeq > Class Template Reference

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



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



Public Member Functions

- ModelSIRDCONN (ModelSIRDCONN< TSeq > &model, std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

  Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRDCONN** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
- ModelSIRDCONN< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void reset ()

  Reset the model.
- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1
- static const int **RECOVERED** = 2
- static const int **DECEASED** = 3

Additional Inherited Members

14.40.1 Constructor & Destructor Documentation

14.40.1.1 ModelSIRDCONN()

```
template<typename TSeq >
ModelSIRDCONN< TSeq >::ModelSIRDCONN (
            ModelSIRDCONN< TSeq > & model,
            std::string vname,
            epiworld_fast_uint n,
            epiworld_double prevalence,
            epiworld_double contact_rate,
            epiworld_double transmission_rate,
            epiworld_double recovery_rate,
            epiworld_double death_rate )  [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)
contact_rate	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery
death_rate	Probability of death

14.40.2 Member Function Documentation

14.40.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRDCONN< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.40.2.2 reset()

```
template<typename TSeq >
void ModelSIRDCONN< TSeq >::reset  [inline], [virtual]
```

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

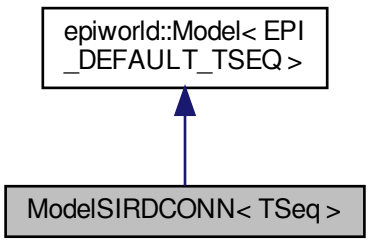
Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

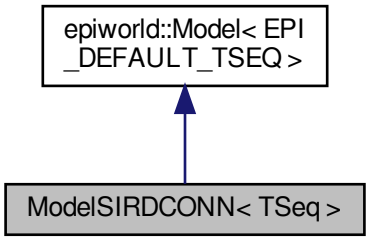
- epiworld.hpp

14.41 ModelSIRDCONN< TSeq > Class Template Reference

Inheritance diagram for ModelSIRDCONN< TSeq >:



Collaboration diagram for ModelSIRDCONN< TSeq >:



Public Member Functions

- ModelSIRDCONN (ModelSIRDCONN< TSeq > &model, std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)

  Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRDCONN** (std::string vname, epiworld_fast_uint n, epiworld_double prevalence, epiworld_double contact_rate, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
- ModelSIRDCONN< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- void reset ()

  Reset the model.
- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1
- static const int **RECOVERED** = 2
- static const int **DECEASED** = 3

Additional Inherited Members

14.41.1 Constructor & Destructor Documentation

14.41.1.1 ModelSIRDCONN()

```
template<typename TSeq >
ModelSIRDCONN< TSeq >::ModelSIRDCONN (
            ModelSIRDCONN< TSeq > & model,
            std::string vname,
            epiworld_fast_uint n,
            epiworld_double prevalence,
            epiworld_double contact_rate,
            epiworld_double transmission_rate,
            epiworld_double recovery_rate,
            epiworld_double death_rate )  [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)
contact_rate	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery
death_rate	Probability of death

14.41.2 Member Function Documentation

14.41.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRDCONN< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.41.2.2 reset()

```
template<typename TSeq >
void ModelSIRDCONN< TSeq >::reset  [inline], [virtual]
```

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

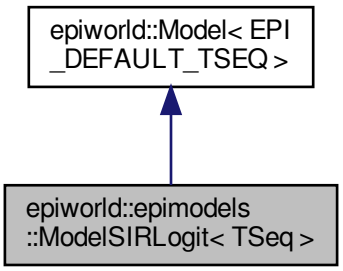
Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

- include/epiworld/models/sirdconnected.hpp

14.42 epiworld::epimodels::ModelSIRLogit< TSeq > Class Template Reference

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



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

Public Member Functions

- ModelSIRLogit (ModelSIRLogit< TSeq > &model, std::string vname, double *data, size_t ncols, std::vector< double > coefs_infect, std::vector< double > coefs_recover, std::vector< size_t > coef_infect_cols, std::vector< size_t > coef_recover_cols, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double prevalence)

  Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRLogit** (std::string vname, double *data, size_t ncols, std::vector< double > coefs_infect, std::vector< double > coefs_recover, std::vector< size_t > coef_infect_cols, std::vector< size_t > coef_recover_cols, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double prevalence)
- ModelSIRLogit< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- void reset ()

  Reset the model.

Public Attributes

- std::vector< double > **coefs_infect**
- std::vector< double > **coefs_recover**
- std::vector< size_t > **coef_infect_cols**
- std::vector< size_t > **coef_recover_cols**

Additional Inherited Members

14.42.1 Constructor & Destructor Documentation

14.42.1.1 ModelSIRLogit()

```
template<typename TSeq >
ModelSIRLogit< TSeq >::ModelSIRLogit (
            ModelSIRLogit< TSeq > & model,
            std::string vname,
            double * data,
            size_t ncols,
            std::vector< double > coefs_infect,
            std::vector< double > coefs_recover,
            std::vector< size_t > coef_infect_cols,
            std::vector< size_t > coef_recover_cols,
            epiworld_double transmission_rate,
            epiworld_double recovery_rate,
            epiworld_double prevalence )  [inline]
```

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

Parameters

vname	Name of the virus.
coefs_infect	Double ptr. Infection coefficients.
coefs_recover	Double ptr. Recovery coefficients.
ncoef_infect	Unsigned int. Number of infection coefficients.
ncoef_recover	Unsigned int. Number of recovery coefficients.
coef_infect_cols	Vector<unsigned int>. Ids of infection vars.
coef_recover_cols	Vector<unsigned int>. Ids of recover vars.
model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
prevalence	Initial prevalence (proportion)
contact_rate	Average number of contacts (interactions) per step.
prob_transmission	Probability of transmission
prob_recovery	Probability of recovery

14.42.2 Member Function Documentation

14.42.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRLogit< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.42.2.2 reset()

```
template<typename TSeq >
void ModelSIRLogit< TSeq >::reset  [inline], [virtual]
```

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

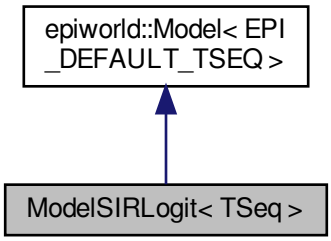
Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

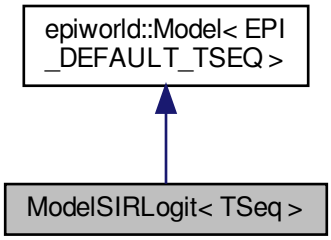
- epiworld.hpp

14.43 ModelSIRLogit< TSeq > Class Template Reference

Inheritance diagram for ModelSIRLogit< TSeq >:



Collaboration diagram for ModelSIRLogit< TSeq >:



Public Member Functions

- ModelSIRLogit (ModelSIRLogit< TSeq > &model, std::string vname, double *data, size_t ncols, std::vector< double > coefs_infect, std::vector< double > coefs_recover, std::vector< size_t > coef_infect_cols, std::vector< size_t > coef_recover_cols, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double prevalence)

  Template for a Susceptible-Infected-Removed (SIR) model.
- **ModelSIRLogit** (std::string vname, double *data, size_t ncols, std::vector< double > coefs_infect, std::vector< double > coefs_recover, std::vector< size_t > coef_infect_cols, std::vector< size_t > coef_recover_cols, epiworld_double transmission_rate, epiworld_double recovery_rate, epiworld_double prevalence)
- ModelSIRLogit< TSeq > & run (epiworld_fast_uint ndays, int seed=-1)

  Runs the simulation (after initialization)
- Model< TSeq > * clone_ptr ()

  Advanced usage: Makes a copy of data and returns it as undeleted pointer.
- void reset ()

  Reset the model.

Public Attributes

- std::vector< double > **coefs_infect**
- std::vector< double > **coefs_recover**
- std::vector< size_t > **coef_infect_cols**
- std::vector< size_t > **coef_recover_cols**

Additional Inherited Members

14.43.1 Constructor & Destructor Documentation

14.43.1.1 ModelSIRLogit()

```
template<typename TSeq >
ModelSIRLogit< TSeq >::ModelSIRLogit (
            ModelSIRLogit< TSeq > & model,
            std::string vname,
            double * data,
            size_t ncols,
            std::vector< double > coefs_infect,
            std::vector< double > coefs_recover,
            std::vector< size_t > coef_infect_cols,
            std::vector< size_t > coef_recover_cols,
            epiworld_double transmission_rate,
            epiworld_double recovery_rate,
            epiworld_double prevalence )  [inline]
```

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

Parameters

vname	Name of the virus.
coefs_infect	Double ptr. Infection coefficients.
coefs_recover	Double ptr. Recovery coefficients.
ncoef_infect	Unsigned int. Number of infection coefficients.
ncoef_recover	Unsigned int. Number of recovery coefficients.
coef_infect_cols	Vector<unsigned int>. Ids of infection vars.
coef_recover_cols	Vector<unsigned int>. Ids of recover vars.
model	A Model<TSeq> object where to set up the SIR.
vname	std::string Name of the virus
prevalence	Initial prevalence (proportion)
contact_rate	Average number of contacts (interactions) per step.
prob_transmission	Probability of transmission
prob_recovery	Probability of recovery

14.43.2 Member Function Documentation

14.43.2.1 clone_ptr()

```
template<typename TSeq >
Model< TSeq > * ModelSIRLogit< TSeq >::clone_ptr  [inline], [virtual]
```

Advanced usage: Makes a copy of data and returns it as undeleted pointer.

Parameters

copy	

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

14.43.2.2 reset()

```
template<typename TSeq >
void ModelSIRLogit< TSeq >::reset  [inline], [virtual]
```

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

Reimplemented from epiworld::Model< EPI_DEFAULT_TSEQ >.

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

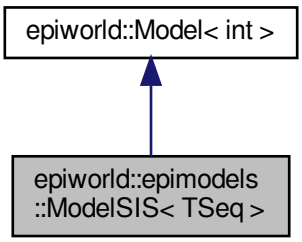
- include/epiworld/models/sirlogit.hpp

14.44 epiworld::epimodels::ModelSIS< TSeq > Class Template Reference

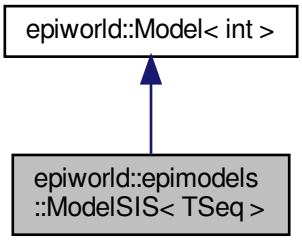
Template for a Susceptible-Infected-Susceptible (SIS) model.

`#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 transmission_rate, epiworld_double recovery_rate)
- **ModelSIS** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_↩ double recovery_rate)

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1

Additional Inherited Members

14.44.1 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 rate of the immune system

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

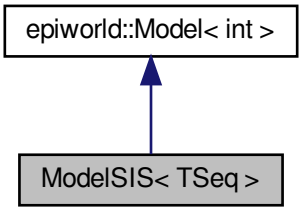
- epiworld.hpp

14.45 ModelSIS< TSeq > Class Template Reference

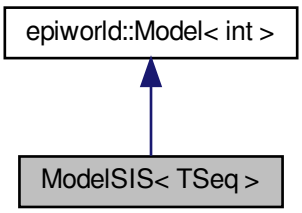
Template for a Susceptible-Infected-Susceptible (SIS) model.

```
#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 transmission_rate, epiworld_double recovery_rate)
- **ModelSIS** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld_↩ double recovery_rate)

Static Public Attributes

- static const int **SUSCEPTIBLE** = 0
- static const int **INFECTED** = 1

Additional Inherited Members

14.45.1 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 rate of the immune system

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

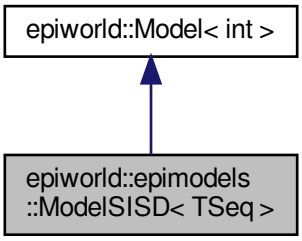
- include/epiworld/models/sis.hpp

14.46 epiworld::epimodels::ModelSISD< TSeq > Class Template Reference

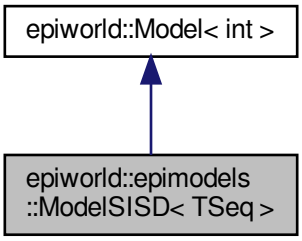
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

`#include <epiworld.hpp>`

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



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



Public Member Functions

- **ModelSISD** (ModelSISD< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_↩ double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
- **ModelSISD** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld↩ _double recovery_rate, epiworld_double death_rate)

Additional Inherited Members

14.46.1 Detailed Description

template<typename TSeq = int>
class epiworld::epimodels::ModelSISD< TSeq >

Template for a Susceptible-Infected-Susceptible-Deceased (SISD) 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 rate of the immune system
inital_death	epiworld_double Initial death_rate of the immune system

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

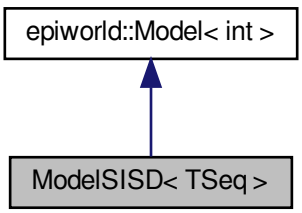
- epiworld.hpp

14.47 ModelSISD< TSeq > Class Template Reference

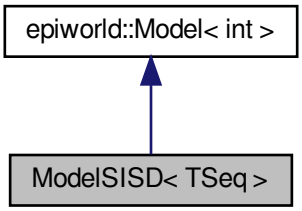
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

`#include <sisd.hpp>`

Inheritance diagram for ModelSISD< TSeq >:



Collaboration diagram for ModelSISD< TSeq >:



Public Member Functions

- **ModelSISD** (ModelSISD< TSeq > &model, std::string vname, epiworld_double prevalence, epiworld_↩ double transmission_rate, epiworld_double recovery_rate, epiworld_double death_rate)
- **ModelSISD** (std::string vname, epiworld_double prevalence, epiworld_double transmission_rate, epiworld↩ _double recovery_rate, epiworld_double death_rate)

Additional Inherited Members

14.47.1 Detailed Description

template<typename TSeq = int>
class ModelSISD< TSeq >

Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

Parameters

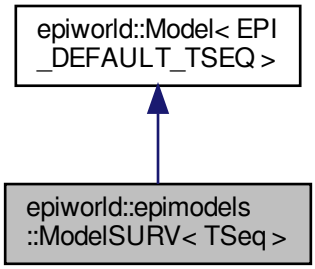
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery_rate rate of the immune system
inital_death	epiworld_double Initial death_rate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

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

- include/epiworld/models/sisd.hpp

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

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



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

Public Member Functions

Construct a new ModelSURV object

The ModelSURV class simulates a survaillence model where agents can be isolated, even if asyptomatic.

Parameters

vname	*String. Name of the virus*
prevalence	*Integer. Number of initial cases of the virus.*
efficacy_vax	*Double. Efficacy of the vaccine (1 - P(acquire the disease)).*
latent_period	*Double. Shape parameter of a `Gamma(latent_period, 1)` distribution. This coincides with the expected number of latent days.*

Parameters

infect_period	*Double. Shape parameter of a `Gamma(infected_period, 1)` distribution. This coincides with the expected number of infectious days.*
prob_symptoms	*Double. Probability of generating symptoms.*
prop_vaccinated	*Double. Probability of vaccination. Coincides with the initial prevalence of vaccinated individuals.*
prop_vax_redux_transm	*Double. Factor by which the vaccine reduces transmissibility.*
prop_vax_redux_infect	*Double. Factor by which the vaccine reduces the chances of becoming infected.*
surveillance_prob	*Double. Probability of testing an agent.*
prob_transmission	*Double. Raw transmission probability.*
prob_death	*Double. Raw probability of death for symptomatic individuals.*
prob_noreinfect	*Double. Probability of no re-infection.*

This model features the following states:

- *Susceptible*
- *Latent*
- *Symptomatic*
- *Symptomatic isolated*
- *Asymptomatic*
- *Asymptomatic isolated*
- *Recovered*
- *Removed*

Returns

An object of class `epiworld_surv`

- **ModelSURV** ()
- **ModelSURV** (ModelSURV< TSeq > &model, std::string vname, epiworld_fast_uint 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)
- **ModelSURV** (std::string vname, epiworld_fast_uint 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)

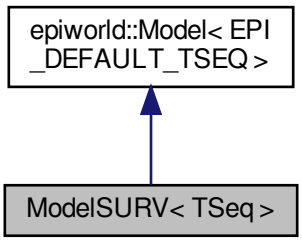
Additional Inherited Members

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

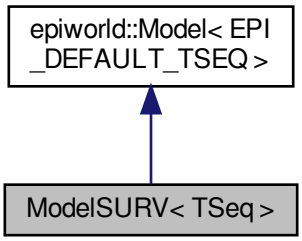
- epiworld.hpp

14.49 ModelSURV< TSeq > Class Template Reference

Inheritance diagram for ModelSURV< TSeq >:



Collaboration diagram for ModelSURV< TSeq >:



Public Member Functions

Construct a new ModelSURV object

The ModelSURV class simulates a survaillence model where agents can be isolated, even if asyptomatic.

Parameters

vname	*String. Name of the virus*
prevalence	*Integer. Number of initial cases of the virus.*
efficacy_vax	*Double. Efficacy of the vaccine (1 - P(acquire the disease)).*
latent_period	*Double. Shape parameter of a `Gamma(latent_period, 1)` distribution. This coincides with the expected number of latent days.*
infect_period	*Double. Shape parameter of a `Gamma(infected_period, 1)` distribution. This coincides with the expected number of infectious days.*
prob_symptoms	*Double. Probability of generating symptoms.*

Parameters

prop_vaccinated	*Double. Probability of vaccination. Coincides with the initial prevalence of vaccinated individuals.*
prop_vax_redux_transm	*Double. Factor by which the vaccine reduces transmissibility.*
prop_vax_redux_infect	*Double. Factor by which the vaccine reduces the chances of becoming infected.*
surveillance_prob	*Double. Probability of testing an agent.*
prob_transmission	*Double. Raw transmission probability.*
prob_death	*Double. Raw probability of death for symptomatic individuals.*
prob_noreinfect	*Double. Probability of no re-infection.*

This model features the following states:

- *Susceptible*
- *Latent*
- *Symptomatic*
- *Symptomatic isolated*
- *Asymptomatic*
- *Asymptomatic isolated*
- *Recovered*
- *Removed*

Returns

An object of class `epiworld_surv`

- **ModelSURV** ()
- **ModelSURV** (ModelSURV< TSeq > &model, std::string vname, epiworld_fast_uint 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)
- **ModelSURV** (std::string vname, epiworld_fast_uint 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

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

- include/epiworld/models/surveillance.hpp

14.50 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

14.51 epiworld::PersonTools< TSeq > Class Template Reference

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

- epiworld.hpp

14.52 PersonTools< TSeq > Class Template Reference

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

- include/epiworld/config.hpp

14.53 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** ()

14.53.1 Detailed Description

A simple progress bar.

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

- epiworld.hpp

14.54 Progress Class Reference

A simple progress bar.

`#include <progress.hpp>`

Public Member Functions

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

14.54.1 Detailed Description

A simple progress bar.

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

- include/epiworld/progress.hpp

14.55 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[ ]** (epiworld_fast_uint i)
- void **reset** ()
- bool **operator==** (const Queue< TSeq > &other) const
- bool **operator!=** (const Queue< TSeq > &other) const

Static Public Attributes

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

Friends

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

14.55.1 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

14.56 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[ ]** (epiworld_fast_uint i)
- void **reset** ()
- bool **operator==** (const Queue< TSeq > &other) const
- bool **operator!=** (const Queue< TSeq > &other) const

Static Public Attributes

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

Friends

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

14.56.1 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

14.57 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

14.58 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

14.59 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

14.60 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** (std::shared_ptr< TSeq > d)
- std::shared_ptr< TSeq > **get_sequence** ()
- 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_state** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *post)
- bool **operator==** (const Tool< TSeq > &other) const
- bool **operator!=** (const Tool< TSeq > &other) const
- void **print** () const

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, Model< TSeq > *model)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v, Model< TSeq > *model)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v, Model< TSeq > *model)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v, Model< TSeq > *model)
- 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** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_tool** (Event< TSeq > &a, Model< TSeq > *m)

14.60.1 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

14.61 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** (std::shared_ptr< TSeq > d)
- std::shared_ptr< TSeq > **get_sequence** ()
- 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_state** (epiworld_fast_int init, epiworld_fast_int post)
- void **set_queue** (epiworld_fast_int init, epiworld_fast_int post)
- void **get_state** (epiworld_fast_int *init, epiworld_fast_int *post)
- void **get_queue** (epiworld_fast_int *init, epiworld_fast_int *post)
- bool **operator==** (const Tool< TSeq > &other) const
- bool **operator!=** (const Tool< TSeq > &other) const
- void **print** () const
- bool **operator==** (const Tool< std::vector< int >> &other) const
- bool **operator==** (const Tool< std::vector< int >> &other) const

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, Model< TSeq > *model)
- epiworld_double **get_transmission_reduction** (VirusPtr< TSeq > v, Model< TSeq > *model)
- epiworld_double **get_recovery_enhancer** (VirusPtr< TSeq > v, Model< TSeq > *model)
- epiworld_double **get_death_reduction** (VirusPtr< TSeq > v, Model< TSeq > *model)
- 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** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_tool** (Event< TSeq > &a, Model< TSeq > *m)

14.61.1 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

14.62 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
- void **print** () const noexcept

Friends

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

14.62.1 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

14.63 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
- void **print** () const noexcept

Friends

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

14.63.1 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

14.64 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
- void **print** () const noexcept

Friends

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

14.64.1 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

14.65 Tools_const< TSeq > Class Template Reference

Set of Tools (const) (useful for iterators)

`#include <tools-bones.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
- void **print** () const noexcept

Friends

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

14.65.1 Detailed Description

template<typename TSeq>
class Tools_const< TSeq >

Set of Tools (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

14.66 epiworld::UserData< TSeq > Class Template Reference

Personalized data by the user.

`#include <epiworld.hpp>`

Public Member Functions

- **UserData** (Model< TSeq > &m)
- **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)
- epiworld_fast_uint **nrow** () const
- epiworld_fast_uint **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** (epiworld_fast_uint j, epiworld_double x)

Access data

Parameters

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

Returns

epiworld_double&

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

Friends

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

14.66.1 Detailed Description

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

Personalized data by the user.

Template Parameters

TSeq	

14.66.2 Constructor & Destructor Documentation

14.66.2.1 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

14.67 UserData< TSeq > Class Template Reference

Personalized data by the user.

`#include <userdata-bones.hpp>`

Public Member Functions

- **UserData** (Model< TSeq > &m)
- **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)
- epiworld_fast_uint **nrow** () const
- epiworld_fast_uint **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** (epiworld_fast_uint j, epiworld_double x)

Access data

Parameters

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

Returns

epiworld_double&

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

Friends

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

14.67.1 Detailed Description

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

Personalized data by the user.

Template Parameters

TSeq	

14.67.2 Constructor & Destructor Documentation

14.67.2.1 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

14.68 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

14.68.1 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

14.69 vecHasher< T > Struct Template Reference

Vector hasher.

`#include <misc.hpp>`

Public Member Functions

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

14.69.1 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

14.70 epiworld::Virus< TSeq > Class Template Reference

Virus.

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

Public Member Functions

- **Virus** (std::string name="unknown virus")
- void **mutate** (Model< TSeq > *model)
- void **set_mutation** (MutFun< TSeq > fun)
- std::shared_ptr< TSeq > **get_sequence** ()
- void **set_sequence** (TSeq sequence)
- Agent< TSeq > * **get_agent** ()
- void **set_agent** (Agent< TSeq > *p)
- 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** ()
- bool **operator==** (const Virus< TSeq > &other) const
- bool **operator!=** (const Virus< TSeq > &other) const
- void **print** () const

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** (Model< TSeq > *model)
- epiworld_double **get_prob_recovery** (Model< TSeq > *model)
- epiworld_double **get_prob_death** (Model< TSeq > *model)
- epiworld_double **get_incubation** (Model< TSeq > *model)
- void **post_recovery** (Model< TSeq > *model)

- 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_incubation_fun** (VirusFun< TSeq > fun)
- void **set_prob_infecting** (const epiworld_double *prob)
- void **set_prob_recovery** (const epiworld_double *prob)
- void **set_prob_death** (const epiworld_double *prob)
- void **set_incubation** (const epiworld_double *prob)
- void **set_prob_infecting** (epiworld_double prob)
- void **set_prob_recovery** (epiworld_double prob)
- void **set_prob_death** (epiworld_double prob)
- void **set_incubation** (epiworld_double prob)

Get and set the state and queue

After applied, viruses can change the state 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 state 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_state** (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_state** (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** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_virus** (Event< TSeq > &a, Model< TSeq > *m)

14.70.1 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

14.71 Virus< TSeq > Class Template Reference

Virus.

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

Public Member Functions

- **Virus** (std::string name="unknown virus")
- void **mutate** (Model< TSeq > *model)
- void **set_mutation** (MutFun< TSeq > fun)
- std::shared_ptr< TSeq > **get_sequence** ()
- void **set_sequence** (TSeq sequence)
- Agent< TSeq > * **get_agent** ()
- void **set_agent** (Agent< TSeq > *p)
- 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** ()
- bool **operator==** (const Virus< TSeq > &other) const
- bool **operator!=** (const Virus< TSeq > &other) const
- void **print** () const
- bool **operator==** (const Virus< std::vector< int >> &other) const
- bool **operator==** (const Virus< std::vector< int >> &other) const

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** (Model< TSeq > *model)
- epiworld_double **get_prob_recovery** (Model< TSeq > *model)
- epiworld_double **get_prob_death** (Model< TSeq > *model)
- epiworld_double **get_incubation** (Model< TSeq > *model)
- void **post_recovery** (Model< TSeq > *model)

- 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_incubation_fun** (VirusFun< TSeq > fun)
- void **set_prob_infecting** (const epiworld_double *prob)
- void **set_prob_recovery** (const epiworld_double *prob)
- void **set_prob_death** (const epiworld_double *prob)
- void **set_incubation** (const epiworld_double *prob)
- void **set_prob_infecting** (epiworld_double prob)
- void **set_prob_recovery** (epiworld_double prob)
- void **set_prob_death** (epiworld_double prob)
- void **set_incubation** (epiworld_double prob)

Get and set the state and queue

After applied, viruses can change the state 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 state 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_state** (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_state** (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** (Event< TSeq > &a, Model< TSeq > *m)
- void **default_rm_virus** (Event< TSeq > &a, Model< TSeq > *m)

14.71.1 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

14.72 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< TSeq > >::iterator **begin** ()
- std::vector< VirusPtr< TSeq > >::iterator **end** ()
- VirusPtr< TSeq > & **operator()** (size_t i)
- VirusPtr< TSeq > & **operator[ ]** (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

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

14.72.1 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

14.73 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< TSeq > >::iterator **begin** ()
- std::vector< VirusPtr< TSeq > >::iterator **end** ()
- VirusPtr< TSeq > & **operator()** (size_t i)
- VirusPtr< TSeq > & **operator[ ]** (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

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

14.73.1 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

14.74 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< TSeq > >::const_iterator **begin** () const
- std::vector< VirusPtr< TSeq > >::const_iterator **end** () const
- const VirusPtr< TSeq > & **operator()** (size_t i)
- const VirusPtr< TSeq > & **operator[ ]** (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

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

14.74.1 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

14.75 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< TSeq > >::const_iterator **begin** () const
- std::vector< VirusPtr< TSeq > >::const_iterator **end** () const
- const VirusPtr< TSeq > & **operator()** (size_t i)
- const VirusPtr< TSeq > & **operator[ ]** (size_t i)
- size_t **size** () const noexcept
- void **print** () const noexcept

Friends

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

14.75.1 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 15

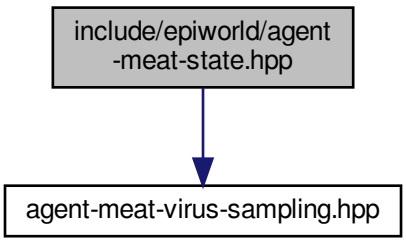
File Documentation

15.1 include/epiworld/agent-meat-state.hpp File Reference

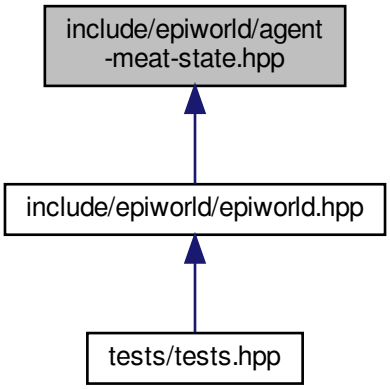
Sampling functions are getting big, so we keep them in a separate file.

```
#include "agent-meat-virus-sampling.hpp"
```

Include dependency graph for agent-meat-state.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)

15.1.1 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

Index