epiworld

0.0-1

Generated by Doxygen 1.9.1

1 Example: 00-hello-world	1
2 Benchmarking	3
3 Contributor Code of Conduct	5
4 epiworld c++ template library	7
4.1 Main features	7
4.2 Algorithm	7
4.3 Hello world (C++)	8
4.4 Surveillance simulation	8
4.4.1 Preliminary results	9
4.4.2 Cases detected	10
5 MIT License	11
6 model1	13
7 Mixing probabilities in connected model	15
7.1 Case 1: No grouping	15
7.2 Case 2: Grouping	16
8 EPI Simulator	17
8.1 Disease dynamics	17
8.2 Network dynamics	17
8.3 Contagion dynamics	17
8.4 Time dynamics	17
8.5 Updating agent's status	18
8.5.1 Other parameters	18
9 Namespace Index	19
9.1 Namespace List	19
10 Hierarchical Index	21
10.1 Class Hierarchy	21
11 Class Index	23
11.1 Class List	23
12 File Index	27
12.1 File List	27
13 Namespace Documentation	29
13.1 epiworld::sampler Namespace Reference	29
13.1.1 Detailed Description	29
13.1.2 Function Documentation	29
13.1.2.1 make_sample_virus_neighbors()	29

13.1.2.2 make_update_susceptible()	30
13.1.2.3 sample_virus_single()	30
13.2 sampler Namespace Reference	32
13.2.1 Detailed Description	32
13.2.2 Function Documentation	32
13.2.2.1 make_sample_virus_neighbors()	32
13.2.2.2 make_update_susceptible()	33
13.2.2.3 sample_virus_single()	33
Class Documentation	37
14.1 AdjList Class Reference	37
14.1.1 Constructor & Destructor Documentation	37
14.1.1.1 AdjList()	37
14.1.2 Member Function Documentation	38
14.1.2.1 read_edgelist()	38
14.2 epiworld::AdjList Class Reference	38
14.2.1 Constructor & Destructor Documentation	39
14.2.1.1 AdjList()	39
14.2.2 Member Function Documentation	39
14.2.2.1 read_edgelist()	39
14.3 Agent $<$ TSeq $>$ Class Template Reference	40
14.3.1 Detailed Description	42
14.3.2 Member Function Documentation	42
14.3.2.1 operator()()	42
14.3.2.2 swap_neighbors()	43
14.3.3 Friends And Related Function Documentation	43
14.3.3.1 default_rm_entity	43
14.4 epiworld::Agent < TSeq > Class Template Reference	44
14.4.1 Detailed Description	46
14.4.2 Member Function Documentation	46
14.4.2.1 operator()()	46
14.4.2.2 swap_neighbors()	47
14.4.3 Friends And Related Function Documentation	47
14.4.3.1 default_rm_entity	47
14.5 AgentsSample < TSeq > Class Template Reference	47
14.5.1 Detailed Description	48
14.5.2 Constructor & Destructor Documentation	48
14.5.2.1 AgentsSample()	48
14.6 epiworld::AgentsSample < TSeq > Class Template Reference	49
14.6.1 Detailed Description	49
14.6.2 Constructor & Destructor Documentation	50
14.6.2.1 AgentsSample()	50

14.7 DataBase < TSeq > Class Template Reference	50
14.7.1 Detailed Description	52
14.7.2 Member Function Documentation	52
14.7.2.1 generation_time()	53
14.7.2.2 get_transmissions()	53
14.7.2.3 operator==() [1/3]	53
14.7.2.4 operator==() [2/3]	54
14.7.2.5 operator==() [3/3]	54
14.7.2.6 record_virus()	54
14.7.2.7 reproductive_number()	54
14.7.2.8 transition_probability()	55
14.8 epiworld::DataBase < TSeq > Class Template Reference	55
14.8.1 Detailed Description	57
14.8.2 Member Function Documentation	57
14.8.2.1 generation_time()	57
14.8.2.2 get_transmissions()	58
14.8.2.3 operator==()	58
14.8.2.4 record_virus()	58
14.8.2.5 reproductive_number()	59
14.8.2.6 transition_probability()	59
14.9 Entities < TSeq > Class Template Reference	59
14.9.1 Detailed Description	60
14.10 epiworld::Entities < TSeq > Class Template Reference	60
14.10.1 Detailed Description	61
14.11 Entities_const< TSeq > Class Template Reference	61
14.11.1 Detailed Description	61
14.12 epiworld::Entities_const< TSeq > Class Template Reference	62
14.12.1 Detailed Description	62
14.13 Entity < TSeq > Class Template Reference	63
14.13.1 Friends And Related Function Documentation	63
14.13.1.1 default_rm_entity	64
14.14 epiworld::Entity< TSeq > Class Template Reference	64
14.14.1 Friends And Related Function Documentation	65
14.14.1.1 default_rm_entity	65
14.15 epiworld::Event< TSeq > Struct Template Reference	65
14.15.1 Detailed Description	66
14.15.2 Constructor & Destructor Documentation	67
14.15.2.1 Event()	67
14.16 Event< TSeq > Struct Template Reference	67
14.16.1 Detailed Description	68
14.16.2 Constructor & Destructor Documentation	68
14.16.2.1 Event()	68

14.17 epiworld::GlobalEvent< TSeq > Class Template Reference	69
14.17.1 Detailed Description	70
14.17.2 Constructor & Destructor Documentation	70
14.17.2.1 GlobalEvent()	70
14.18 GlobalEvent < TSeq > Class Template Reference	70
14.18.1 Detailed Description	71
14.18.2 Constructor & Destructor Documentation	71
14.18.2.1 GlobalEvent()	71
14.19 epiworld::GroupSampler < TSeq > Class Template Reference	71
14.19.1 Detailed Description	72
14.20 GroupSampler < TSeq > Class Template Reference	72
14.20.1 Detailed Description	72
14.21 epiworld::LFMCMC< TData > Class Template Reference	72
14.21.1 Detailed Description	73
14.22 LFMCMC < TData > Class Template Reference	74
14.22.1 Detailed Description	75
14.23 epiworld::Model < TSeq > Class Template Reference	75
14.23.1 Detailed Description	83
14.23.2 Member Function Documentation	83
14.23.2.1 add_globalevent()	84
14.23.2.2 clone_ptr()	84
14.23.2.3 events_add()	84
14.23.2.4 events_run()	85
14.23.2.5 load_agents_entities_ties()	85
14.23.2.6 reset()	86
14.23.2.7 run_multiple()	86
14.23.2.8 set_agents_data()	86
14.23.2.9 set_name()	87
14.23.2.10 write_data()	87
14.23.3 Member Data Documentation	88
14.23.3.1 initial_states_fun	88
14.23.3.2 rbinomd	88
14.23.3.3 rexpd	88
14.23.3.4 rgammad	88
14.23.3.5 rlognormald	89
14.23.3.6 rnormd	89
14.23.3.7 runifd	89
14.23.3.8 time_elapsed	89
14.24 Model < TSeq > Class Template Reference	90
14.24.1 Detailed Description	98
14.24.2 Member Function Documentation	98
14.24.2.1 add_globalevent()	98

14.24.2.2 clone_ptr()	98
14.24.2.3 events_add()	99
14.24.2.4 events_run()	99
14.24.2.5 load_agents_entities_ties()	100
14.24.2.6 reset()	100
14.24.2.7 run_multiple()	100
14.24.2.8 set_agents_data()	101
14.24.2.9 set_name()	101
14.24.2.10 write_data()	101
14.24.3 Member Data Documentation	102
14.24.3.1 initial_states_fun	102
14.24.3.2 rbinomd	102
14.24.3.3 rexpd	102
14.24.3.4 rgammad	103
14.24.3.5 rlognormald	103
14.24.3.6 rnormd	103
14.24.3.7 runifd	103
14.24.3.8 time_elapsed	103
14.25 epiworld::epimodels::ModelDiffNet< TSeq > Class Template Reference	104
14.25.1 Detailed Description	105
14.26 ModelDiffNet< TSeq > Class Template Reference	105
14.26.1 Detailed Description	107
14.27 epiworld::epimodels::ModelSEIR< TSeq > Class Template Reference	107
14.27.1 Detailed Description	108
14.27.2 Member Function Documentation	109
14.27.2.1 initial_states()	109
14.27.3 Member Data Documentation	109
14.27.3.1 update_exposed_seir	109
14.27.3.2 update_infected_seir	110
14.28 ModelSEIR < TSeq > Class Template Reference	110
14.28.1 Detailed Description	111
14.28.2 Member Function Documentation	111
14.28.2.1 initial_states()	111
14.28.3 Member Data Documentation	112
14.28.3.1 update_exposed_seir	112
14.28.3.2 update_infected_seir	112
14.29 epiworld::epimodels::ModelSEIRCONN< TSeq > Class Template Reference	113
14.29.1 Constructor & Destructor Documentation	114
14.29.1.1 ModelSEIRCONN()	114
14.29.2 Member Function Documentation	114
14.29.2.1 clone_ptr()	115
14.29.2.2 initial_states()	115

14.29.2.3 reset()	115
14.30 ModelSEIRCONN < TSeq > Class Template Reference	116
14.30.1 Constructor & Destructor Documentation	117
14.30.1.1 ModelSEIRCONN()	117
14.30.2 Member Function Documentation	117
14.30.2.1 clone_ptr()	117
14.30.2.2 initial_states()	118
14.30.2.3 reset()	118
14.31 epiworld::epimodels::ModelSEIRD< TSeq > Class Template Reference	119
14.31.1 Detailed Description	120
14.31.2 Constructor & Destructor Documentation	120
14.31.2.1 ModelSEIRD() [1/2]	120
14.31.2.2 ModelSEIRD() [2/2]	121
14.31.3 Member Data Documentation	121
14.31.3.1 update_exposed_seir	121
14.32 ModelSEIRD< TSeq > Class Template Reference	122
14.32.1 Detailed Description	123
14.32.2 Constructor & Destructor Documentation	123
14.32.2.1 ModelSEIRD() [1/2]	123
14.32.2.2 ModelSEIRD() [2/2]	124
14.32.3 Member Data Documentation	124
14.32.3.1 update_exposed_seir	124
14.33 epiworld::epimodels::ModelSEIRDCONN< TSeq > Class Template Reference	125
14.33.1 Constructor & Destructor Documentation	126
14.33.1.1 ModelSEIRDCONN()	126
14.33.2 Member Function Documentation	127
14.33.2.1 clone_ptr()	127
14.33.2.2 initial_states()	127
14.33.2.3 reset()	127
14.34 ModelSEIRDCONN < TSeq > Class Template Reference	128
14.34.1 Constructor & Destructor Documentation	129
14.34.1.1 ModelSEIRDCONN()	129
14.34.2 Member Function Documentation	130
14.34.2.1 clone_ptr()	130
14.34.2.2 initial_states()	130
14.34.2.3 reset()	131
14.35 epiworld::epimodels::ModelSEIRMixing< TSeq > Class Template Reference	131
14.35.1 Constructor & Destructor Documentation	132
14.35.1.1 ModelSEIRMixing() [1/2]	133
14.35.1.2 ModelSEIRMixing() [2/2]	133
14.35.2 Member Function Documentation	134
14.35.2.1 clone_ptr()	134

14.35.2.2 initial_states()
14.35.2.3 reset()
14.36 ModelSEIRMixing < TSeq > Class Template Reference
14.36.1 Constructor & Destructor Documentation
14.36.1.1 ModelSEIRMixing() [1/2]
14.36.1.2 ModelSEIRMixing() [2/2]
14.36.2 Member Function Documentation
14.36.2.1 clone_ptr()
14.36.2.2 initial_states()
14.36.2.3 reset()
14.37 epiworld::epimodels::ModelSIR< TSeq > Class Template Reference
14.37.1 Detailed Description
14.37.2 Member Function Documentation
14.37.2.1 initial_states()
14.38 ModelSIR < TSeq > Class Template Reference
14.38.1 Detailed Description
14.38.2 Member Function Documentation
14.38.2.1 initial_states()
14.39 epiworld::epimodels::ModelSIRCONN< TSeq > Class Template Reference
14.39.1 Constructor & Destructor Documentation
14.39.1.1 ModelSIRCONN()
14.39.2 Member Function Documentation
14.39.2.1 clone_ptr()
14.39.2.2 get_n_infected()
14.39.2.3 initial_states()
14.39.2.4 reset()
14.40 ModelSIRCONN< TSeq > Class Template Reference
14.40.1 Constructor & Destructor Documentation
14.40.1.1 ModelSIRCONN()
14.40.2 Member Function Documentation
14.40.2.1 clone_ptr()
14.40.2.2 get_n_infected()
14.40.2.3 initial_states()
14.40.2.4 reset()
14.41 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference
14.41.1 Detailed Description
14.41.2 Constructor & Destructor Documentation
14.41.2.1 ModelSIRD()
14.41.3 Member Function Documentation
14.41.3.1 initial_states()
14.42 ModelSIRD< TSeq > Class Template Reference
14.42.1 Detailed Description

14.42.2 Constructor & Destructor Documentation
14.42.2.1 ModelSIRD()
14.42.3 Member Function Documentation
14.42.3.1 initial_states()
14.43 epiworld::epimodels::ModelSIRDCONN< TSeq > Class Template Reference
14.43.1 Constructor & Destructor Documentation
14.43.1.1 ModelSIRDCONN()
14.43.2 Member Function Documentation
14.43.2.1 clone_ptr()
14.43.2.2 reset()
14.44 ModelSIRDCONN< TSeq > Class Template Reference
14.44.1 Constructor & Destructor Documentation
14.44.1.1 ModelSIRDCONN()
14.44.2 Member Function Documentation
14.44.2.1 clone_ptr()
14.44.2.2 reset()
14.45 epiworld::epimodels::ModelSIRLogit < TSeq > Class Template Reference
14.45.1 Detailed Description
14.45.2 Constructor & Destructor Documentation
14.45.2.1 ModelSIRLogit()
14.45.3 Member Function Documentation
14.45.3.1 clone_ptr()
14.45.3.2 reset()
14.46 ModelSIRLogit < TSeq > Class Template Reference
14.46.1 Detailed Description
14.46.2 Constructor & Destructor Documentation
14.46.2.1 ModelSIRLogit()
14.46.3 Member Function Documentation
14.46.3.1 clone_ptr()
14.46.3.2 reset()
14.47 epiworld::epimodels::ModelSIRMixing< TSeq > Class Template Reference
14.47.1 Constructor & Destructor Documentation
14.47.1.1 ModelSIRMixing() [1/2]
14.47.1.2 ModelSIRMixing() [2/2]
14.47.2 Member Function Documentation
14.47.2.1 clone_ptr()
14.47.2.2 initial_states()
14.47.2.3 reset()
14.48 ModelSIRMixing < TSeq > Class Template Reference
14.48.1 Constructor & Destructor Documentation
14.48.1.1 ModelSIRMixing() [1/2]
14.48.1.2 ModelSIRMixing() [2/2]

14.48.2 Member Function Documentation
14.48.2.1 clone_ptr()
14.48.2.2 initial_states()
14.48.2.3 reset()
$14.49 \; epiworld:: epimodels:: Model SIS < TSeq > Class \; Template \; Reference \; . \; . \; . \; . \; . \; . \; . \; . \; . \; $
14.49.1 Detailed Description
14.50 ModelSIS < TSeq > Class Template Reference
14.50.1 Detailed Description
$14.51 \ epiworld:: epimodels:: Model SISD < TSeq > Class \ Template \ Reference \ . \ . \ . \ . \ . \ . \ . \ . \ . \ $
14.51.1 Detailed Description
14.52 ModelSISD< TSeq > Class Template Reference
14.52.1 Detailed Description
$14.53 \; epiworld:: epimodels:: Model SURV < TSeq > Class \; Template \; Reference \; \ldots \; \ldots \; \ldots \; 1790 \; description \; 1790 \; description$
14.54 ModelSURV < TSeq > Class Template Reference
$14.55 \ Network < Nettype, \ Nodetype, \ Edgetype > Class \ Template \ Reference \ \ldots \ldots \ldots \ldots \ldots 183 \ Network < Nettype, \ Nodetype, \ Nodetype, \ Nodetype > Class \ Network = Network$
$14.56 \; epiworld:: Person Tools < TSeq > Class \; Template \; Reference \; . \; . \; . \; . \; . \; . \; . \; . \; . \; $
14.57 PersonTools < TSeq > Class Template Reference
14.58 epiworld::Progress Class Reference
14.58.1 Detailed Description
14.59 Progress Class Reference
14.59.1 Detailed Description
14.60 epiworld::Queue < TSeq > Class Template Reference
14.60.1 Detailed Description
14.61 Queue < TSeq > Class Template Reference
14.61.1 Detailed Description
14.62 RandGraph Class Reference
14.63 epiworld::SAMPLETYPE Class Reference
14.64 SAMPLETYPE Class Reference
14.65 epiworld::Tool < TSeq > Class Template Reference
14.65.1 Detailed Description
14.66 Tool < TSeq > Class Template Reference
14.66.1 Detailed Description
14.67 epiworld::Tools < TSeq > Class Template Reference
14.67.1 Detailed Description
14.68 Tools < TSeq > Class Template Reference
14.68.1 Detailed Description
14.69 epiworld::Tools_const< TSeq > Class Template Reference
14.69.1 Detailed Description
14.70 Tools_const< TSeq > Class Template Reference
14.70.1 Detailed Description
14.71 epiworld::UserData < TSeq > Class Template Reference
14.71.1 Detailed Description

14.71.2 Constructor & Destructor Documentation) 5
14.71.2.1 UserData()	3 5
14.72 UserData < TSeq > Class Template Reference	}5
14.72.1 Detailed Description	96
14.72.2 Constructor & Destructor Documentation	96
14.72.2.1 UserData()) 6
14.73 epiworld::vecHasher< T > Struct Template Reference) 7
14.73.1 Detailed Description) 7
14.74 vecHasher < T > Struct Template Reference) 7
14.74.1 Detailed Description	}7
14.75 epiworld::Virus < TSeq > Class Template Reference	98
14.75.1 Detailed Description) 9
14.76 Virus < TSeq > Class Template Reference)0
14.76.1 Detailed Description)1
14.77 epiworld::Viruses< TSeq > Class Template Reference)2
14.77.1 Detailed Description)2
14.78 Viruses < TSeq > Class Template Reference)3
14.78.1 Detailed Description)3
14.79 epiworld::Viruses_const< TSeq > Class Template Reference)3
14.79.1 Detailed Description)4
14.80 Viruses_const< TSeq > Class Template Reference)4
14.80.1 Detailed Description)5
15 File Documentation 20)7
15.1 include/epiworld/agent-meat-state.hpp File Reference)7
15.1.1 Detailed Description)8
Index 20)9

Example: 00-hello-world

Output from the program:

```
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
```

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.

4 Benchmarking

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/

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 $\sim~\text{Gamma}$ (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...
##
##
##
## SIMULATION STUDY
##
                   : 20000
## Population size
## 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:
                           : 12.0000
## - Infect period
## - Latent period
## - 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
                                           0 -> 2
## - Total symptomatic isolated (I) :
##
   - Total asymptomatic (I)
  - 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 = " ")</pre>
surv1 <- read.csv("07-surveillance_user_data.txt", sep = " ")</pre>
# 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...
##
##
##
## SIMULATION STUDY
## Population size : 20000 . 200 (duration)
## Days (duration)
                    : 200 (of 200)
## Number of variants : 1
## Last run elapsed t : 530.00ms
## Rewiring
##
## Virus(es):
```

```
## - Covid19 (baseline prevalence: 100 seeds)
## Tool(s):
## - Vaccine (baseline prevalence: 25.00%)
##
## Model parameters:
## - Infect period
                           : 12.0000
                          : 3.0000
##
   - Latent period
## - Prob of symptoms
                            : 0.7000
  - Prob of transmission : 1.0000
                           : 0.0010
   - Prob. death
##
   - 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 = " ")</pre>
surv2 <- read.csv("07-surveillance_user_data.txt", sep = " ")</pre>
hist_comb <- rbind(
 cbind(sim = as.character(s_levels[1]), hist1),
 cbind(sim = as.character(s_levels[2]), hist2)
qqplot(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

MIT License

Copyright (c) 2021 George G. Vega Yon

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

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

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

12 MIT License

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 $\mathbb{E}\left(v,k\right)$ and pseudo vaccines (recovered) have efficacy rate $\mathbb{E}\left(r,k\right)$ $< \mathbb{E}\left(v,k\right)$. 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 ( $\mathbb{N}(i,t,v)$ ),
  - 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  $\mathbb{M}(i)$  and uniformly shared with other countries at rate  $\mathbb{S}(i)$ . c. Population flows between each country pair (i,j) at a rate  $\mathbb{F}(i,j)$ . Flows between countries do not change Population and are symmetric.

14 model1

- 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  ${\bf k}$  with probability:
    - ```  $Pr(h->s|i,t,k,u) \sim sum(g in \{u,v\}) (N(i,t-1,s,k|g) + sum(j != i) F(i,j) * N(j,t-1,s,k|g)) * C(k) / (N(i) + sum(j != i) N(j)) ```$
    - b. Vaccinated individuals can become sick of variant k with probability:  $\Pr(v->s|i,t,k,v) \sim \Pr(h->s|i,t,k) * (1 E(v,k))$ .
    - b. Recovered individuals can become sick of variant k with probability:  $\Pr(v->s|i,t,k,r) \sim \Pr(h->s|i,t,k) * (1 E(r,k))$ .
    - 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  $P(u->v) \sim V(A(i,t), B(i))$ .
    - e. The country vaccine supply changes.

## Mixing probabilities in connected model

George G. Vega Yon, Ph.D. 2024-04-25

### 7.1 Case 1: No grouping

We will look into the probability of drawing infected individuals to simplify the algorithm. There are \$1\$ infected individuals at any time in the simulation; thus, instead of drawing from \$Bern(c/N, N)\$, we will be drawing from \$ $\leftarrow$  Bern(c/N, I)\$. The next step is to check which infected individuals should be drawn. Let's compare the distributions using the hypergeometric as an example:

```
set.seed(132)
nsims <- 1e4
N <- 400
rate <- 5
p <- rate/N
I <- 10
sim_complex <- parallel::mclapply(1:nsims, \(i) {</pre>
 nsamples <- rbinom(N, N, p)
sum(rbinom(N, size = nsamples, prob = I/N) > 0)
}, mc.cores = 4L) |> unlist()
sim_simple <- parallel::mclapply(1:nsims, \(i) {</pre>
sum(rbinom(N, I, p) > 0)
}, mc.cores = 4L) |> unlist()
op <- par(mfrow = c(1,2))
MASS::truehist(sim_complex)
MASS::truehist(sim_simple)
par(op)
quantile(sim_complex)
 0% 25% 50% 75% 100%
 27 43 47 51
quantile(sim_simple)
 0% 25% 50% 75% 100%
 23 43 47 51 71
```

These two approaches are equivalent, but the second one is more efficient from the computational perspective.

plotter(sim\_complex, sim\_simple)

set.seed(123133)

### 7.2 Case 2: Grouping

This explores the case when we have mixing across groups. The question is if we can replicate the effect at the group level.

```
ngroups <- 3
mixing <- matrix(
 c(0.1, 0.2, 0.3, 0.2, 0.1, 0.2, 0.3, 0.2, 0.1),
 nrow = ngroups,
 ncol = ngroups
mixing <- mixing/rowSums(mixing)</pre>
mixing
 [,1]
 [,2]
[1,] 0.1666667 0.3333333 0.5000000
[2,] 0.4000000 0.2000000 0.4000000
[3,] 0.5000000 0.3333333 0.1666667
N <- 500
sizes <- c(100, 150, 250) rate <- 5
p <- rate/N
I <- c(10, 30, 20)
ids <- rep.int(1:ngroups, times = sizes)</pre>
nsims <- 1e4
sim_complex <- parallel::mclapply(1:nsims, \(i) {</pre>
 # Sampling group first
 # How many each individual will sample from the groups
 ans <- rbinom(
 n = N, size = sizes[g], prob = mixing[ids,][,g] * p
) |> sum()
 # Sampling with replacement rbinom(ans, size = 1, prob = I[g]/sizes[g]) |> sum()
 }) |> sum()
, mc.cores = 4L) > unlist()
Using the alternative method in which we directly weight the probabilities:
sim_simple <- parallel::mclapply(1:nsims, \(i) {
 # Sampling group first</pre>
 sapply(1:ngroups, \(g) {
 rbinom(
 n = N, size = I[g], prob = mixing[cbind(ids,g)] * p
) |> sum()
}) |> sum()
}, mc.cores = 4L) |> unlist()
op <- par(mfrow = c(1,2))</pre>
MASS::truehist(sim_complex)
MASS::truehist(sim_simple)
par(op)
quantile(sim_complex)
 0% 25% 50% 75% 100%
 94 101 131
 88
quantile(sim_simple)
 0% 25% 50% 75% 100%
 58 87 94 101 135
plotter(sim_complex, sim_simple)
```

### **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  $\mathbb{L}\left(\frac{1}{N}\right)$  function of the local number of infections. This way, if

18 EPI Simulator

#### 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(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.

# Namespace Index

## 9.1 Namespace List

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

epiworld	::sampler					
	Functions for sampling viruses	 	 	 	 	 29
sampler						
	Functions for sampling viruses					32

20 Namespace Index

# **Hierarchical Index**

## 10.1 Class Hierarchy

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

AdjList
epiworld::AdjList
Agent< TSeq >
epiworld::Agent < TSeq >
AgentsSample < TSeq >
epiworld::AgentsSample < TSeq >
DataBase < TSeq >
epiworld::DataBase< TSeq >
Entities < TSeq >
epiworld::Entities < TSeq >
Entities_const< TSeq >
epiworld::Entities_const< TSeq >
Entity < TSeq >
epiworld::Entity< TSeq >
epiworld::Event < TSeq >
Event< TSeq >
epiworld::GlobalEvent< TSeq >
GlobalEvent < TSeq >
epiworld::GroupSampler< TSeq >
GroupSampler < TSeq >
epiworld::LFMCMC< TData >
LFMCMC< TData >
epiworld::Model < TSeq >
Model < TSeq >
epiworld::Model< EPI_DEFAULT_TSEQ >
ModelSEIRCONN < TSeq >
ModelSEIRDCONN< TSeq >
ModelSEIRMixing < TSeq >
ModelSIRCONN < TSeq >
ModelSIRDCONN< TSeq >
ModelSIRLogit < TSeq >
ModelSIRMixing < TSeq >
ModelSURV < TSeq >
epiworld::epimodels::ModelSEIRCONN< TSeq >
epiworld::epimodels::ModelSEIRDCONN< TSeq >

22 Hierarchical Index

epiworld::epimodels::ModelSEIRMixing< TSeq >	. 131
epiworld::epimodels::ModelSIRCONN< TSeq >	143
epiworld::epimodels::ModelSIRDCONN <tseq></tseq>	154
epiworld::epimodels::ModelSIRLogit< TSeq >	159
epiworld::epimodels::ModelSIRMixing< TSeq >	166
epiworld::epimodels::ModelSURV < TSeq >	179
epiworld::Model < int >	75
$ModelDiffNet {} \ \ldots \ $	
ModelSEIR< TSeq >	
ModelSEIRD< TSeq >	.122
ModelSIR < TSeq >	140
ModelSIRD< TSeq >	. 151
ModelSIS< TSeq >	175
ModelSISD< TSeq >	
epiworld::epimodels::ModelDiffNet< TSeq >	
${\sf epiworld::epimodels::} Model SEIR < TSeq > \dots $	
epiworld::epimodels::ModelSEIRD< TSeq >	
${\sf epiworld::epimodels::} {\sf ModelSIR} {\sf < TSeq} > \dots $	
${\it epiworld::epimodels::} Model SIRD < TSeq > \dots $	
${\sf epiworld::epimodels::} {\sf ModelSIS} {\sf < TSeq} > \dots $	
${\sf epiworld::epimodels::} {\sf ModelSISD}{<\sf TSeq} > \dots $	176
Network < Nettype, Nodetype, Edgetype >	183
epiworld::PersonTools < TSeq >	183
PersonTools < TSeq >	184
epiworld::Progress	184
Progress	184
epiworld::Queue < TSeq >	185
Queue < TSeq >	185
RandGraph	
epiworld::SAMPLETYPE	
SAMPLETYPE	
epiworld::Tool< TSeq >	
Tool< TSeq >	
epiworld::Tools< TSeq >	
Tools< TSeq >	
epiworld::Tools_const< TSeq >	
Tools_const< TSeq >	192
epiworld::UserData< TSeq >	193
UserData < TSeq >	195
epiworld::vecHasher< T >	197
vecHasher< T >	197
epiworld::Virus< TSeq >	198
Virus < TSeq >	200
epiworld::Viruses< TSeq >	202
Viruses < TSeq >	203
epiworld::Viruses_const< TSeq >	203
Viruses_const< TSeq >	204

# **Class Index**

### 11.1 Class List

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

AdjList	37
epiworld::AdjList	38
Agent < TSeq >	
Agent (agents)	40
epiworld::Agent < TSeq >	
Agent (agents)	44
AgentsSample < TSeq >	
Sample of agents	47
epiworld::AgentsSample < TSeq >	
Sample of agents	49
DataBase < TSeq >	
Statistical data about the process	50
epiworld::DataBase < TSeq >	
Statistical data about the process	55
Entities < TSeq >	
Set of Entities (useful for building iterators)	59
epiworld::Entities< TSeq >	
Set of Entities (useful for building iterators)	60
Entities_const< TSeq >	
Set of Entities (const) (useful for iterators)	61
epiworld::Entities_const< TSeq >	
Set of Entities (const) (useful for iterators)	62
Entity< TSeq >	63
epiworld::Entity< TSeq >	64
epiworld::Event< TSeq >	
Event data for update an agent	65
Event< TSeq >	
Event data for update an agent	67
epiworld::GlobalEvent< TSeq >	
Template for a Global Event	69
GlobalEvent< TSeq >	
Template for a Global Event	70
epiworld::GroupSampler< TSeq >	
Weighted sampling of groups	71
GroupSampler< TSeq >	
Weighted sampling of groups	72

24 Class Index

epiworld::LFMCMC< TData >	
Likelihood-Free Markov Chain Monte Carlo	72
LFMCMC< TData >	
Likelihood-Free Markov Chain Monte Carlo	74
epiworld::Model < TSeq >	
Core class of epiworld	75
Model < TSeq >	
Core class of epiworld	90
epiworld::epimodels::ModelDiffNet< TSeq >	
Template for a Network Diffusion Model	104
ModelDiffNet< TSeq >	
Template for a Network Diffusion Model	105
epiworld::epimodels::ModelSEIR< TSeq >	407
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model	107
ModelSEIR < TSeq >	440
Template for a Susceptible-Exposed-Infected-Removed (SEIR) model	
epiworld::epimodels::ModelSEIRCONN< TSeq >	113
ModelSEIRCONN < TSeq >	116
epiworld::epimodels::ModelSEIRD< TSeq >	440
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model	119
ModelSEIRD < TSeq >	400
Template for a Susceptible-Exposed-Infected-Removed-Deceased (SEIRD) model	122
epiworld::epimodels::ModelSEIRDCONN< TSeq >	125
ModelSEIRDCONN < TSeq >	128
epiworld::epimodels::ModelSEIRMixing< TSeq >	131
ModelSEIRMixing < TSeq >	135
epiworld::epimodels::ModelSIR< TSeq >	400
Template for a Susceptible-Infected-Removed (SIR) model	139
ModelSIR < TSeq >	4.40
Template for a Susceptible-Infected-Removed (SIR) model	
epiworld::epimodels::ModelSIRCONN< TSeq >	143
ModelSIRCONN< TSeq >	146
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model	149
ModelSIRD< TSeq >	149
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model	151
epiworld::epimodels::ModelSIRDCONN< TSeq >	154
ModelSIRDCONN < TSeq >	157
epiworld::epimodels::ModelSIRLogit< TSeq >	137
Template for a Susceptible-Infected-Removed (SIR) model	159
ModelSIRLogit < TSeq >	100
Template for a Susceptible-Infected-Removed (SIR) model	163
epiworld::epimodels::ModelSIRMixing< TSeq >	166
ModelSIRMixing < TSeq >	170
epiworld::epimodels::ModelSIS< TSeq >	170
Template for a Susceptible-Infected-Susceptible (SIS) model	173
ModelSIS< TSeq >	170
Template for a Susceptible-Infected-Susceptible (SIS) model	175
epiworld::epimodels::ModelSISD< TSeq >	175
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model	176
ModelSISD< TSeq >	
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model	178
epiworld::epimodels::ModelSURV< TSeq >	179
ModelSURV < TSeq >	181
Network< Nettype, Nodetype, Edgetype >	183
epiworld::PersonTools< TSeq >	183
PersonTools < TSeq >	

11.1 Class List

epiworld::Progress	
A simple progress bar	84
Progress	
1 1 0	84
epiworld::Queue< TSeq >	
Controls which agents are verified at each step	85
Queue < TSeq >	
Ŭ I	85
RandGraph	86
epiworld::SAMPLETYPE	87
	87
epiworld::Tool< TSeq >	
Tools for defending the agent against the virus	87
Tool< TSeq >	
	88
epiworld::Tools< TSeq >	
` '	90
Tools< TSeq >	
- · · · · · · · · · · · · · · · · · · ·	91
epiworld::Tools_const< TSeq >	
	92
Tools_const< TSeq >	
	92
epiworld::UserData< TSeq >	
·	93
UserData < TSeq >	
•	95
epiworld::vecHasher< T >	~~
	97
vecHasher < T >	97
Vector hasher	97
	98
Virus < TSeq >	90
·	00
epiworld::Viruses< TSeq >	JU
	02
Viruses < TSeq >	02
·	03
epiworld::Viruses_const< TSeq >	JJ
	03
Viruses_const< TSeq >	
	04

26 Class Index

# Chapter 12

# File Index

## 12.1 File List

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

epiworld.hpp
include/epiworld/adjlist-bones.hpp??
include/epiworld/adjlist-meat.hpp
include/epiworld/agent-bones.hpp
include/epiworld/agent-events-meat.hpp
include/epiworld/agent-meat-state.hpp
Sampling functions are getting big, so we keep them in a separate file
include/epiworld/agent-meat-virus-sampling.hpp??
include/epiworld/agent-meat.hpp??
include/epiworld/agentssample-bones.hpp
include/epiworld/config.hpp
include/epiworld/database-bones.hpp
include/epiworld/database-meat.hpp??
include/epiworld/entities-bones.hpp
include/epiworld/entity-bones.hpp
include/epiworld/entity-meat.hpp
include/epiworld/epiworld-macros.hpp
include/epiworld/epiworld.hpp
include/epiworld/globalevent-bones.hpp
include/epiworld/globalevent-meat.hpp
include/epiworld/ <b>groupsampler-bones.hpp</b>
include/epiworld/ <b>groupsampler-meat.hpp</b>
include/epiworld/misc.hpp
include/epiworld/model-bones.hpp??
include/epiworld/model-meat-print.hpp
include/epiworld/model-meat.hpp
include/epiworld/ <b>network-bones.hpp</b>
include/epiworld/ <b>progress.hpp</b>
include/epiworld/queue-bones.hpp??
include/epiworld/randgraph.hpp
include/epiworld/random_graph.hpp??
include/epiworld/seq_processing.hpp
include/epiworld/tool-bones.hpp
include/epiworld/tool-meat.hpp??
include/epiworld/tools-bones.hpp

28 File Index

include/epiworld/userdata-bones.hpp
include/epiworld/ <b>userdata-meat.hpp</b>
include/epiworld/virus-bones.hpp
include/epiworld/virus-meat.hpp
include/epiworld/viruses-bones.hpp
include/epiworld/math/lfmcmc.hpp??
include/epiworld/math/lfmcmc/lfmcmc-bones.hpp??
include/epiworld/math/lfmcmc/lfmcmc-meat-print.hpp
include/epiworld/math/lfmcmc/lfmcmc-meat.hpp??
include/epiworld/models/diffnet.hpp
include/epiworld/models/globalevents.hpp
include/epiworld/models/init-functions.hpp
include/epiworld/models/models.hpp??
include/epiworld/models/seir.hpp ??
include/epiworld/models/seirconnected.hpp??
include/epiworld/models/seird.hpp
include/epiworld/models/seirdconnected.hpp
include/epiworld/models/seirmixing.hpp??
include/epiworld/models/sir.hpp??
include/epiworld/models/sirconnected.hpp
include/epiworld/models/sird.hpp ??
include/epiworld/models/sirdconnected.hpp??
include/epiworld/models/sirlogit.hpp??
include/epiworld/models/sirmixing.hpp??
include/epiworld/models/sis.hpp
include/epiworld/models/sisd.hpp
include/epiworld/models/surveillance.hpp
tests/ <b>tests.hpp</b>

## **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()

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**

#### **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()

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**

avcluda	unsigned vector of states that need to be excluded from the sampling
CACIUUC	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()

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**

р	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()

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**

#### **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()

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**

TC-	
i Sea	

### **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()

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.)

Tem	plate	Parai	meters

TSeq	
•	

#### **Parameters**

р	Pointer to person
m	Pointer to the model

## Returns

 $\label{thm:continuity} \mbox{Virus$<$TSeq$>*$ 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()

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()

Construct a new Adj List object.

lds 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.

lds 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  $\mbox{\bf operator==}$  (const  $\mbox{\bf Agent}<\mbox{\bf TSeq}>\mbox{\bf \&other})$  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 gueue=-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< TSeg > v, Model< TSeg > \*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()()

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**



#### Returns

double&

#### 14.3.2.2 swap\_neighbors()

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

- < 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 gueue=-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 gueue=-99)
- void rm\_tool (epiworld\_fast\_uint tool\_idx, Model < TSeq > \*model, epiworld\_fast\_int state\_new=-99, epiworld fast int gueue=-99)
- void rm\_tool (ToolPtr< TSeq > &tool, Model< TSeq > \*model, epiworld\_fast\_int state\_new=-99, epiworld fast int gueue=-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 gueue=-99)
- void rm\_entity (Entity < TSeq > &entity, Model < TSeq > \*model, epiworld\_fast\_int state\_new=-99, epiworld fast int gueue=-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

```
\label{template} \begin{tabular}{ll} template < typename TSeq > \\ class epiworld::Agent < TSeq > \\ \end{tabular}
```

Agent (agents)

**Template Parameters** 

TSea

Sequence type (should match TSeq across the model)

#### 14.4.2 Member Function Documentation

## 14.4.2.1 operator()()

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**



Returns

double&

#### 14.4.2.2 swap\_neighbors()

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

- < 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()

Sample from the agent's entities.

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

#### **Template Parameters**

#### **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**

TSea	
1004	

#### 14.6.2 Constructor & Destructor Documentation

#### 14.6.2.1 AgentsSample()

Sample from the agent's entities.

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

#### **Template Parameters**

|--|

## **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\_transmission, std::string fn\_transmiss
- 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=""></typename>	>
class DataBase < TSeq >	

Statistical data about the process.

**Template Parameters** 

TSeq	

## 14.7.2 Member Function Documentation

#### 14.7.2.1 generation\_time()

#### 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]

< Date when the source acquired the varia,

Generated by Doxygen

#### 14.7.2.4 operator==() [2/3]

#### 14.7.2.5 operator==() [3/3]

## 14.7.2.6 record\_virus()

< Date when the source acquired the varia

Registering a new variant.

#### **Parameters**

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()

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\_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()

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==()

- < 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()

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()

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** 



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

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

#### epiworld::Entities < TSeq > Class Template Reference 14.10

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 ti)
- 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** 



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**

TSea	
1009	

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** 



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
- void distribute ()
- std::vector< size\_t > & get\_agents ()

## **Public Attributes**

- epiworld\_double **prevalence** = 0.0
- bool prevalence as proportion = false
- EntityToAgentFun< TSeq > dist\_fun = nullptr

## **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

- < 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
- 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
- void distribute ()

## **Public Attributes**

- epiworld double **prevalence** = 0.0
- bool prevalence\_as\_proportion = false
- EntityToAgentFun< TSeq > dist\_fun = nullptr

### **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

- < 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

 $\begin{array}{l} {\tt template}{<} {\tt typename} \ {\tt TSeq}{>} \\ {\tt struct} \ {\tt epiworld::Event}{<} \ {\tt TSeq}{>} \end{array}$ 

Event data for update an agent.

# **Template Parameters**

### 14.15.2 Constructor & Destructor Documentation

## 14.15.2.1 Event()

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_←	Next state
state_	
queue_	Efect on the queue
call_	The action call (if needed)
idx_⊷	Location of agent in object.
agent_	
idx_⇔	Location of object in agent.
object_	

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** 



## 14.16.2 Constructor & Destructor Documentation

## 14.16.2.1 Event()

```
ActionFun< TSeq > call_,
int idx_agent_,
int idx_object_) [inline]
```

Construct a new **Event** object.

All the parameters are rather optional.

#### **Parameters**

Agent over who the action will happen
Virus to add
Tool to add
Index of virus to be removed (if needed)
Index of tool to be removed (if needed)
Next state
Efect on the queue
The action call (if needed)
Location of agent in 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**

- $\bullet \ \, \textbf{GlobalEvent} \ (\textbf{GlobalFun} < \textbf{TSeq} > \textbf{fun, std::string name, int day=-99}) \\$ 
  - 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

```
\label{template} \begin{tabular}{ll} template < typename TSeq > \\ class epiworld::GlobalEvent < TSeq > \\ \end{tabular}
```

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()

Construct a new Global Event object.

#### **Parameters**

fun	A function that takes a Model <tseq> * as argument and returns void.  A descriptive name for the action.</tseq>	
name		
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()

Construct a new Global Event object.

#### **Parameters**

fun	A function that takes a Model <tseq> * as argument and returns void.</tseq>	
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::GroupSampler < TSeq > Class Template Reference

Weighted sampling of groups.

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

# **Public Member Functions**

- GroupSampler (const std::vector < double > &contact\_matrix\_, const std::vector < size\_t > &group\_sizes ←
   \_, bool normalize=true)
- int sample\_1 (Model < TSeq > \*model, const int origin\_group)
- void sample\_n (Model < TSeq > \*model, std::vector < int > &sample, const int origin\_group, const int nsamples)

# 14.19.1 Detailed Description

```
template<typename TSeq>
class epiworld::GroupSampler< TSeq>
```

Weighted sampling of groups.

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

· epiworld.hpp

# 14.20 GroupSampler < TSeq > Class Template Reference

Weighted sampling of groups.

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

## **Public Member Functions**

- GroupSampler (const std::vector< double > &contact\_matrix\_, const std::vector< size\_t > &group\_sizes ←
   \_, bool normalize=true)
- int sample\_1 (Model < TSeq > \*model, const int origin\_group)
- void sample\_n (Model < TSeq > \*model, std::vector < int > &sample, const int origin\_group, const int nsamples)

## 14.20.1 Detailed Description

```
template<typename TSeq> class GroupSampler< TSeq >
```

Weighted sampling of groups.

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

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

# 14.21 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)

## 14.21.1 Detailed Description

 $\label{template} \mbox{template} < \mbox{typename TData} > \\ \mbox{class epiworld::LFMCMC} < \mbox{TData} > \\$ 

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

# 14.22 LFMCMC < TData > Class Template Reference

Likelihood-Free Markov Chain Monte Carlo.

#include <1fmcmc-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)

# 14.22.1 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:

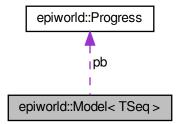
- include/epiworld/math/lfmcmc/lfmcmc-bones.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat-print.hpp
- include/epiworld/math/lfmcmc/lfmcmc-meat.hpp

# 14.23 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.

• void load\_agents\_entities\_ties (const std::vector< int > &agents\_ids, const std::vector< int > &entities\_ids)

Associate agents-entities from data.

• 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\_transmissio

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.

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 add\_entity\_n (Entity < TSeq > e, epiworld\_fast\_uint preval)
- void add\_entity\_fun (Entity < TSeq > e, EntityToAgentFun < TSeq > fun)
- void rm\_virus (size t virus pos)
- void rm\_tool (size\_t tool\_pos)
- void rm\_entity (size\_t entity\_id)

## 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.

- Agent < TSeq > & get\_agent (size\_t i)
- 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 ()
- Entity < TSeq > & get\_entity (size\_t entity\_id, int \*entity\_pos=nullptr)
- 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

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  $\{(i,j), (k,l)\} \rightarrow \{(i,l), (k,j)\}$ , the reciprocal is also true, i.e.,  $\{(j,i), (l,k)\} \rightarrow \{(j,k), (l,i)\}$ .

#### **Parameters**

proportion	Proportion of ties to be rewired.
p p	1

### 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**

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 dist\_entities ()
- 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 = {}
```

# Auxiliary variables for AgentsSample<TSeq> iterators

Variables used to keep track of the events to be made regarding viruses.

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

#### \_ . . \_ \_ \_

• epiworld fast uint nactions = 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
- std::vector< int > array\_int\_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)

## 14.23.1 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.)

## 14.23.2 Member Function Documentation

## 14.23.2.1 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.

## 14.23.2.2 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**

сору

## 14.23.2.3 events\_add()

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_←	New state of the agent
state_	
call_	Function the action will call
queue_	Change in the queue
idx_←	Location of agent in object.
agent_	
idx_←	Location of object in agent.
object_	

# 14.23.2.4 events\_run()

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

Executes the stored action.

### **Parameters**

model⊷	Model over which it will be executed.

# 14.23.2.5 load\_agents\_entities\_ties()

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.

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

#### 14.23.2.6 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

## 14.23.2.7 run\_multiple()

Parameters

ndays Multiple runs of the simulation

#### 14.23.2.8 set\_agents\_data()

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.	

# 14.23.2.9 set\_name()

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

Set the name object.

## **Parameters**

name

# 14.23.2.10 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

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

## 14.23.3 Member Data Documentation

# 14.23.3.1 initial\_states\_fun

Function to distribute states. Goes along with the function

## 14.23.3.2 rbinomd

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

Initial value:
=
```

std::binomial\_distribution<>()

## 14.23.3.3 rexpd

# 14.23.3.4 rgammad

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

# Initial value:

std::gamma\_distribution<>()

## 14.23.3.5 rlognormald

#### Initial value:

std::normal\_distribution<>(0.0)

template<typename TSeq >

## 14.23.3.7 runifd

std::normal\_distribution epiworld::Model< TSeq >::rnormd [protected]

## 14.23.3.8 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:

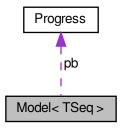
· epiworld.hpp

# 14.24 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.

- void load\_agents\_entities\_ties (const std::vector< int > &agents\_ids, const std::vector< int > &entities\_ids)
   Associate agents-entities from data.
- void load\_agents\_entities\_ties (const int \*agents\_id, const int \*entities\_id, size\_t n)
- 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\_creproductive\_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

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 add entity n (Entity < TSeq > e, epiworld fast uint preval)
- void add\_entity\_fun (Entity< TSeq > e, EntityToAgentFun< TSeq > fun)
- void rm\_virus (size t virus pos)
- void rm\_tool (size\_t tool\_pos)
- void rm\_entity (size\_t entity\_id)

## Accessing population of the model

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.

- Agent < TSeq > & get\_agent (size t i)
- 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 ()
- Entity < TSeq > & get\_entity (size\_t entity\_id, int \*entity\_pos=nullptr)
- 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  $\underline{\mathsf{Model}} < \mathsf{TSeq} > \& \mathsf{run}$  (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

fn std::string. File name.
fn ctd::ctring File name

#### **Parameters**

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 <code>epiworld\_fast\_uint</code> method directly fetches the parameters in the order these were added to the tool. Accessing parameters via the <code>std::string</code> 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.

initial_val	
-------------	--

#### **Parameters**

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**



## 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 < TSeg > fun)
- void set\_death\_reduction\_mixer (MixerFun < TSeq > fun)

### **Protected Member Functions**

```
· void dist_tools ()
```

- · void dist\_virus ()
- · void dist entities ()
- 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**

• int current date = 0

std::chrono::time\_point< std::chrono::steady\_clock > time\_start

```
• 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
```

- 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
- std::vector< int > array int 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)

# 14.24.1 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.)	

# 14.24.2 Member Function Documentation

# 14.24.2.1 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.

# 14.24.2.2 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**

сору

# 14.24.2.3 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 over which the action will be called
Virus pointer included in the action
Tool pointer included in the action
Entity pointer included in the action
New state of the agent
Function the action will call
Change in the queue
Location of agent in object.
Location of object in agent.

# 14.24.2.4 events\_run()

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

Executes the stored action.

#### **Parameters**

model⊷	Model over which it will be executed.

#### 14.24.2.5 load\_agents\_entities\_ties()

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.

# 14.24.2.6 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

# 14.24.2.7 run\_multiple()

#### **Parameters**

ridays   Multiple rulis of the simulation	ndays	Multiple runs of the simulation
-------------------------------------------	-------	---------------------------------

# 14.24.2.8 set\_agents\_data()

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
Ī	ncols⇔	Number of features included in the data.
	_	

# 14.24.2.9 set\_name()

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

Set the name object.

#### **Parameters**

name

# 14.24.2.10 write\_data()

```
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

# 14.24.3 Member Data Documentation

# 14.24.3.1 initial\_states\_fun

Function to distribute states. Goes along with the function

#### 14.24.3.2 rbinomd

# 14.24.3.3 rexpd

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

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

#### 14.24.3.4 rgammad

```
template<typename TSeq >
std::gamma_distribution Model< TSeq >::rgammad [protected]
Initial value:
 std::gamma_distribution<>()
14.24.3.5 rlognormald
template<typename TSeq >
std::lognormal_distribution Model< TSeq >::rlognormald [protected]
Initial value:
 std::lognormal_distribution<>()
14.24.3.6 rnormd
template<typename TSeq >
std::normal_distribution Model< TSeq >::rnormd [protected]
Initial value:
 std::normal_distribution<>(0.0)
14.24.3.7 runifd
template<typename TSeq >
std::uniform_real_distribution Model< TSeq >::runifd [protected]
Initial value:
 std::uniform_real_distribution<> (0.0, 1.0)
14.24.3.8 time_elapsed
{\tt template}{<}{\tt typename}~{\tt 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:

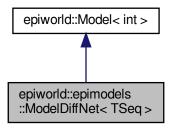
- include/epiworld/agent-bones.hpp
- · include/epiworld/model-bones.hpp
- include/epiworld/model-meat-print.hpp
- include/epiworld/model-meat.hpp

# 14.25 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**

# 14.25.1 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.</tseq>
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

# 14.26 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**

# 14.26.1 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.</tseq>
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy epiworld_double Initial susceptibility_reduction of the immune	
initial_recovery	epiworld_double Initial recovery rate of the immune system

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

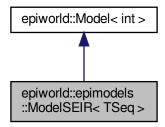
• include/epiworld/models/diffnet.hpp

# 14.27 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**

# 14.27.1 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.</tseq>
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.

# 14.27.2 Member Function Documentation

#### 14.27.2.1 initial\_states()

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 >.

# 14.27.3 Member Data Documentation

# 14.27.3.1 update\_exposed\_seir

#### 14.27.3.2 update\_infected\_seir

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

· epiworld.hpp

# 14.28 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**

# 14.28.1 Detailed Description

```
\label{template} \begin{tabular}{ll} template < typename TSeq = int > \\ class ModelSEIR < TSeq > \\ \end{tabular}
```

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

#### **Parameters**

model	A Model <tseq> object where to set up the SIR.</tseq>
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.

# 14.28.2 Member Function Documentation

# 14.28.2.1 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 >.

#### 14.28.3 Member Data Documentation

#### 14.28.3.1 update exposed seir

#### 14.28.3.2 update\_infected\_seir

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

• include/epiworld/models/seir.hpp

# 14.29 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.

• size\_t get\_n\_infected () const

# **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**

# 14.29.1 Constructor & Destructor Documentation

# 14.29.1.1 ModelSEIRCONN()

```
template<typename TSeq >
ModelSEIRCONN
ModelSEIRCONN
TSeq > immodel,
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.</tseq>
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

# 14.29.2 Member Function Documentation

#### 14.29.2.1 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**

сору

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

# 14.29.2.2 initial\_states()

Set the initial states of the model.

#### **Parameters**

proportions⊷	Double vector with a single element:
_	The proportion of non-infected individuals who have recovered.

 $\label{eq:control_equation} \mbox{Reimplemented from epiworld::} \mbox{Model} < \mbox{EPI\_DEFAULT\_TSEQ} >.$ 

#### 14.29.2.3 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

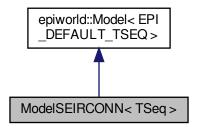
Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

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

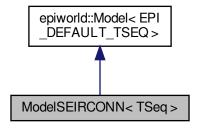
epiworld.hpp

# 14.30 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.

size\_t get\_n\_infected () const

# **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**

#### 14.30.1 Constructor & Destructor Documentation

# 14.30.1.1 ModelSEIRCONN()

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

# Parameters

model	A Model <tseq> object where to set up the SIR.</tseq>
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

# 14.30.2 Member Function Documentation

# 14.30.2.1 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**

сору

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

# 14.30.2.2 initial\_states()

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 >.

#### 14.30.2.3 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:

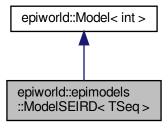
• include/epiworld/models/seirconnected.hpp

# 14.31 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**

# 14.31.1 Detailed Description

```
template<typename TSeq = int>
class epiworld::epimodels::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]

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.

#### **Parameters**

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> 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

# 14.32 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**

# 14.32.1 Detailed Description

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

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

# 14.32.2 Constructor & Destructor Documentation

#### 14.32.2.1 ModelSEIRD() [1/2]

Constructor for the SEIRD model.

# **Template Parameters**

is a light of the sequence used in the model.	TSeq	Type of the sequence used in the model.
-----------------------------------------------	------	-----------------------------------------

#### **Parameters**

model	Reference to the SEIRD model.
vname	Name of the model.

#### **Parameters**

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.

# 14.32.2.2 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.

# 14.32.3 Member Data Documentation

# 14.32.3.1 update\_exposed\_seir

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

• include/epiworld/models/seird.hpp

# 14.33 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.

• size\_t get\_n\_infected () const

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

#### **Parameters**

model	A Model <tseq> object where to set up the SIR.</tseq>
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**

сору

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

# 14.33.2.2 initial\_states()

Set up the initial states of the model.

# **Parameters**

Double vector with the following values:
 O: 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:

· epiworld.hpp

# 14.34 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.

• size t get n infected () const

# **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**

#### 14.34.1 Constructor & Destructor Documentation

#### 14.34.1.1 ModelSEIRDCONN()

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

#### **Parameters**

model	A Model <tseq> object where to set up the SIR.</tseq>
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

# 14.34.2 Member Function Documentation

#### 14.34.2.1 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**



Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

# 14.34.2.2 initial\_states()

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.

 $\label{lem:lemented_lemented_lemented} \textbf{Reimplemented from epiworld::} \\ \textbf{Model} < \textbf{EPI\_DEFAULT\_TSEQ} >.$ 

#### 14.34.2.3 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:

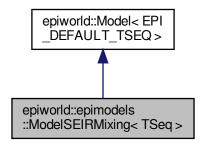
• include/epiworld/models/seirdconnected.hpp

# 14.35 epiworld::epimodels::ModelSEIRMixing< TSeq > Class Template Reference

 $Inheritance\ diagram\ for\ epiworld::epimodels::ModelSEIRMixing < TSeq >:$ 



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



# **Public Member Functions**

ModelSEIRMixing (ModelSEIRMixing < 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, std::vector < double > contact\_matrix)

Constructs a ModelSEIRMixing object.

ModelSEIRMixing (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, std::vector < double > contact\_matrix)

Constructs a ModelSEIRMixing object.

ModelSEIRMixing < 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.

ModelSEIRMixing < TSeq > & initial\_states (std::vector < double > proportions\_, std::vector < int > queue ← \_={})

Set the initial states of the model.

- size\_t get\_n\_infected (size\_t group) const
- void set\_contact\_matrix (std::vector< double > cmat)

# **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**

# 14.35.1 Constructor & Destructor Documentation

## 14.35.1.1 ModelSEIRMixing() [1/2]

Constructs a ModelSEIRMixing object.

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

### **Parameters**

model	A reference to an existing ModelSEIRMixing object.
vname	The name of the ModelSEIRMixing object.
n	The number of entities in the model.
prevalence	The initial prevalence of the disease in the model.
contact_rate	The contact rate between entities in the model.
transmission_rate	The transmission rate of the disease in the model.
avg_incubation_days	The average incubation period of the disease in the model.
recovery_rate	The recovery rate of the disease in the model.
contact_matrix	The contact matrix between entities in the model.
model	A Model <tseq> object where to set up the SIR.</tseq>
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

## 14.35.1.2 ModelSEIRMixing() [2/2]

Constructs a ModelSEIRMixing object.

### **Parameters**

vname	The name of the ModelSEIRMixing object.
n	The number of entities in the model.
prevalence	The initial prevalence of the disease in the model.
contact_rate	The contact rate between entities in the model.
transmission_rate	The transmission rate of the disease in the model.
avg_incubation_days	The average incubation period of the disease in the model.
recovery_rate	The recovery rate of the disease in the model.
contact_matrix	The contact matrix between entities in the model.

# 14.35.2 Member Function Documentation

# 14.35.2.1 clone\_ptr()

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

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

### **Parameters**



Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

# 14.35.2.2 initial\_states()

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 >.

### 14.35.2.3 reset()

```
template<typename TSeq >
void ModelSEIRMixing< 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

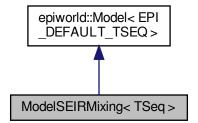
Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

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

· epiworld.hpp

# 14.36 ModelSEIRMixing < TSeq > Class Template Reference

Inheritance diagram for ModelSEIRMixing< TSeq >:



Collaboration diagram for ModelSEIRMixing < TSeq >:



### **Public Member Functions**

ModelSEIRMixing (ModelSEIRMixing < 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, std::vector < double > contact\_matrix)

Constructs a ModelSEIRMixing object.

• ModelSEIRMixing (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, std::vector< double > contact\_matrix)

Constructs a ModelSEIRMixing object.

ModelSEIRMixing< 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.

ModelSEIRMixing < TSeq > & initial\_states (std::vector < double > proportions\_, std::vector < int > queue ← \_={})

Set the initial states of the model.

- · size\_t get\_n\_infected (size\_t group) const
- void set\_contact\_matrix (std::vector< double > cmat)

### **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**

# 14.36.1 Constructor & Destructor Documentation

# 14.36.1.1 ModelSEIRMixing() [1/2]

Constructs a ModelSEIRMixing object.

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

### **Parameters**

model	A reference to an existing ModelSEIRMixing object.
vname	The name of the ModelSEIRMixing object.
п	The number of entities in the model.
prevalence	The initial prevalence of the disease in the model.
contact_rate	The contact rate between entities in the model.
transmission_rate	The transmission rate of the disease in the model.
avg_incubation_days	The average incubation period of the disease in the model.
recovery_rate	The recovery rate of the disease in the model.
contact_matrix	The contact matrix between entities in the model.
model	A Model <tseq> object where to set up the SIR.</tseq>
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

# 14.36.1.2 ModelSEIRMixing() [2/2]

# Constructs a ModelSEIRMixing object.

### **Parameters**

vname	The name of the ModelSEIRMixing object.
п	The number of entities in the model.
prevalence	The initial prevalence of the disease in the model.
contact_rate	The contact rate between entities in the model.
transmission_rate	The transmission rate of the disease in the model.
avg_incubation_days	The average incubation period of the disease in the model.
recovery_rate	The recovery rate of the disease in the model.
contact_matrix	The contact matrix between entities in the model.

## 14.36.2 Member Function Documentation

## 14.36.2.1 clone\_ptr()

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

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

### **Parameters**

```
сору
```

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

## 14.36.2.2 initial\_states()

Set the initial states of the model.

### **Parameters**

proportions↔	Double vector with a single element:
_	The proportion of non-infected individuals who have recovered.

 $\label{eq:control_equation} \mbox{Reimplemented from epiworld::} \mbox{Model} < \mbox{EPI\_DEFAULT\_TSEQ} >.$ 

### 14.36.2.3 reset()

```
template<typename TSeq >
void ModelSEIRMixing< 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/seirmixing.hpp

# 14.37 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**

# 14.37.1 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.</tseq>
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

### 14.37.2 Member Function Documentation

# 14.37.2.1 initial\_states()

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

# 14.38 ModelSIR < TSeq > Class Template Reference

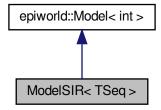
 $Template \ for \ a \ Susceptible-Infected-Removed \ (SIR) \ model.$ 

```
#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**

# 14.38.1 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.</tseq>
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

# 14.38.2 Member Function Documentation

# 14.38.2.1 initial\_states()

Set the initial states of the model.

### **Parameters**

proportions↔	Double vector with a single element:
_	The proportion of non-infected individuals who have recovered.

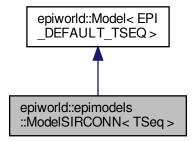
Reimplemented from epiworld::Model < int >.

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

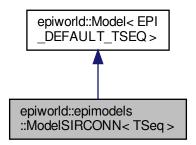
• include/epiworld/models/sir.hpp

# 14.39 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.

• size\_t get\_n\_infected () const

Get the infected individuals.

## **Static Public Attributes**

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

# **Additional Inherited Members**

## 14.39.1 Constructor & Destructor Documentation

## 14.39.1.1 ModelSIRCONN()

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

# **Parameters**

model	A Model <tseq> object where to set up the SIR.</tseq>
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

# 14.39.2 Member Function Documentation

### 14.39.2.1 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**

```
сору
```

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

## 14.39.2.2 get\_n\_infected()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
size_t epiworld::epimodels::ModelSIRCONN< TSeq >::get_n_infected () const [inline]
```

Get the infected individuals.

### Returns

```
std::vector< epiworld::Agent<TSeq> * >
```

### 14.39.2.3 initial states()

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 >.

# 14.39.2.4 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

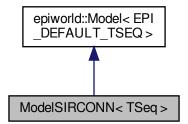
Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

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

· epiworld.hpp

# 14.40 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.

• size t get n infected () const

Get the infected individuals.

### **Static Public Attributes**

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

### **Additional Inherited Members**

# 14.40.1 Constructor & Destructor Documentation

### 14.40.1.1 ModelSIRCONN()

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

### **Parameters**

model	A Model <tseq> object where to set up the SIR.</tseq>
vname	std::string Name of the virus
prevalence	Initial prevalence (proportion)
Generated by Paraygen	Average number of contacts (interactions) per step.
transmission_rate	Probability of transmission
recovery_rate	Probability of recovery

## 14.40.2 Member Function Documentation

## 14.40.2.1 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**

сору

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

## 14.40.2.2 get\_n\_infected()

```
template<typename TSeq = EPI_DEFAULT_TSEQ>
size_t ModelSIRCONN< TSeq >::get_n_infected () const [inline]
```

Get the infected individuals.

### Returns

std::vector< epiworld::Agent<TSeq> \* >

# 14.40.2.3 initial\_states()

Set the initial states of the model.

# **Parameters**

proportions↔	Double vector with a single element:
_	The proportion of non-infected individuals who have recovered.

 $\label{lem:lemented_lemented} \textbf{Reimplemented from epiworld::} \\ \textbf{Model} < \textbf{EPI\_DEFAULT\_TSEQ} >. \\$ 

### 14.40.2.4 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:

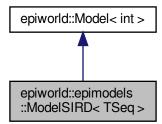
· include/epiworld/models/sirconnected.hpp

# 14.41 epiworld::epimodels::ModelSIRD< TSeq > Class Template Reference

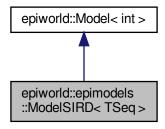
 $Template \ for \ a \ Susceptible-Infected-Removed-Deceased \ (SIRD) \ model.$ 

```
#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**

# 14.41.1 Detailed Description

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

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

### 14.41.2 Constructor & Destructor Documentation

# 14.41.2.1 ModelSIRD()

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.

### 14.41.3 Member Function Documentation

## 14.41.3.1 initial\_states()

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:

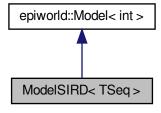
· epiworld.hpp

# 14.42 ModelSIRD< TSeq> Class Template Reference

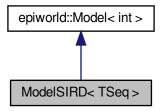
Template for a Susceptible-Infected-Removed-Deceased (SIRD) model.

```
#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**

# 14.42.1 Detailed Description

 $\label{template} \begin{tabular}{ll} template < typename TSeq = int > \\ class ModelSIRD < TSeq > \\ \end{tabular}$ 

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

# 14.42.2 Constructor & Destructor Documentation

# 14.42.2.1 ModelSIRD()

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.

# 14.42.3 Member Function Documentation

### 14.42.3.1 initial\_states()

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.

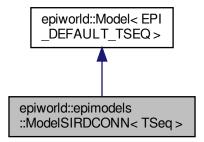
Reimplemented from epiworld::Model < int >.

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

• include/epiworld/models/sird.hpp

# 14.43 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**

# 14.43.1 Constructor & Destructor Documentation

# 14.43.1.1 ModelSIRDCONN()

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

### **Parameters**

model	A Model <tseq> object where to set up the SIR.</tseq>
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

# 14.43.2 Member Function Documentation

# 14.43.2.1 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** 

сору

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

## 14.43.2.2 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:

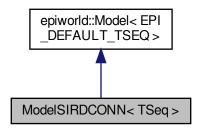
· epiworld.hpp

# 14.44 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**

### 14.44.1 Constructor & Destructor Documentation

## 14.44.1.1 ModelSIRDCONN()

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

## **Parameters**

model	A Model <tseq> object where to set up the SIR.</tseq>
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

## 14.44.2 Member Function Documentation

# 14.44.2.1 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**

сору

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

### 14.44.2.2 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:

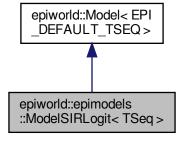
• include/epiworld/models/sirdconnected.hpp

# 14.45 epiworld::epimodels::ModelSIRLogit < TSeq > Class Template Reference

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

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

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



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**

# 14.45.1 Detailed Description

 $\label{template} \mbox{typename TSeq = EPI\_DEFAULT\_TSEQ} > \mbox{class epiworld::epimodels::ModelSIRLogit} < \mbox{TSeq} >$ 

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

In this model, infection and recoveru probabilities are computed using a logit model. Particularly, the probability of infection is computed as:

$$\frac{1}{1 + \exp\left(-\left(\beta_0 E_i + \sum_{i=1}^n \beta_i x_i\right)\right)}$$

where  $\beta_0$  is the exposure coefficient and  $E_i$  is the exposure number,  $\beta_i$  are the coefficients for the features  $x_i$  of the agents, and n is the number of features. The probability of recovery is computed as:

$$\frac{1}{1 + \exp\left(-\left(\sum_{i=1}^{n} \beta_i x_i\right)\right)}$$

where  $\beta_i$  are the coefficients for the features  $x_i$  of the agents, and n is the number of features.

### **Parameters**

	TSeq	Type of the sequence (e.g. std::vector, std::deque)	
--	------	-----------------------------------------------------	--

# 14.45.2 Constructor & Destructor Documentation

# 14.45.2.1 ModelSIRLogit()

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.</unsigned>
coef_recover_cols	Vector <unsigned int="">. Ids of recover vars.</unsigned>

### **Parameters**

model	A Model <tseq> object where to set up the SIR.</tseq>
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

## 14.45.3 Member Function Documentation

### 14.45.3.1 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**

сору

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

# 14.45.3.2 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

# 14.46 ModelSIRLogit < TSeq > Class Template Reference

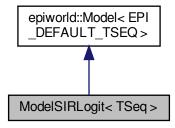
Template for a Susceptible-Infected-Removed (SIR) model.

#include <sirlogit.hpp>

Inheritance diagram for ModelSIRLogit< TSeg >:



Collaboration diagram for ModelSIRLogit < TSeq >:



### **Public Member Functions**

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**

# 14.46.1 Detailed Description

```
template<typename TSeq = EPI_DEFAULT_TSEQ> class ModelSIRLogit< TSeq >
```

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

In this model, infection and recoveru probabilities are computed using a logit model. Particularly, the probability of infection is computed as:

$$\frac{1}{1 + \exp\left(-\left(\beta_0 E_i + \sum_{i=1}^n \beta_i x_i\right)\right)}$$

where  $\beta_0$  is the exposure coefficient and  $E_i$  is the exposure number,  $\beta_i$  are the coefficients for the features  $x_i$  of the agents, and n is the number of features. The probability of recovery is computed as:

$$\frac{1}{1 + \exp\left(-\left(\sum_{i=1}^{n} \beta_i x_i\right)\right)}$$

where  $\beta_i$  are the coefficients for the features  $x_i$  of the agents, and n is the number of features.

### **Parameters**

```
TSeq Type of the sequence (e.g. std::vector, std::deque)
```

### 14.46.2 Constructor & Destructor Documentation

# 14.46.2.1 ModelSIRLogit()

```
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.</unsigned>
coef_recover_cols	Vector <unsigned int="">. Ids of recover vars.</unsigned>
model	A Model <tseq> object where to set up the SIR.</tseq>
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

# 14.46.3 Member Function Documentation

### 14.46.3.1 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**

сору

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

### 14.46.3.2 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

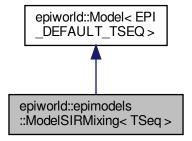
Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

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

• include/epiworld/models/sirlogit.hpp

# 14.47 epiworld::epimodels::ModelSIRMixing< TSeq > Class Template Reference

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



 $\label{localized} \mbox{Collaboration diagram for epiworld::epimodels::} \mbox{ModelSIRMixing} < \mbox{TSeq} > :$ 



### **Public Member Functions**

ModelSIRMixing (ModelSIRMixing
 TSeq > &model, std::string vname, epiworld\_fast\_uint n, epiworld
 \_double prevalence, epiworld\_double contact\_rate, epiworld\_double transmission\_rate, epiworld\_double recovery\_rate, std::vector< double > contact\_matrix)

Constructs a ModelSIRMixing object.

ModelSIRMixing (std::string vname, epiworld\_fast\_uint n, epiworld\_double prevalence, epiworld\_double contact\_rate, epiworld\_double transmission\_rate, epiworld\_double recovery\_rate, std::vector< double > contact\_matrix)

Constructs a ModelSIRMixing object.

ModelSIRMixing < 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.

ModelSIRMixing< TSeq > & initial\_states (std::vector< double > proportions\_, std::vector< int > queue ← \_={})

Set the initial states of the model.

- · size t get n infected (size t group) const
- void set\_contact\_matrix (std::vector< double > cmat)

### Static Public Attributes

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

### **Additional Inherited Members**

### 14.47.1 Constructor & Destructor Documentation

## 14.47.1.1 ModelSIRMixing() [1/2]

Constructs a ModelSIRMixing object.

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

### **Parameters**

model	A reference to an existing ModelSIRMixing object.
vname	The name of the ModelSIRMixing object.
n	The number of entities in the model.
prevalence	The initial prevalence of the disease in the model.
contact_rate	The contact rate between entities in the model.
transmission_rate	The transmission rate of the disease in the model.
recovery_rate	The recovery rate of the disease in the model.
contact_matrix	The contact matrix between entities in the model.
model	A Model <tseq> object where to set up the SIR.</tseq>
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

# 14.47.1.2 ModelSIRMixing() [2/2]

# Constructs a ModelSIRMixing object.

### **Parameters**

vname	The name of the ModelSIRMixing object.
n	The number of entities in the model.
prevalence	The initial prevalence of the disease in the model.
contact_rate	The contact rate between entities in the model.
transmission_rate	The transmission rate of the disease in the model.
recovery_rate	The recovery rate of the disease in the model.
contact_matrix	The contact matrix between entities in the model.

# 14.47.2 Member Function Documentation

#### 14.47.2.1 clone\_ptr()

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

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

#### **Parameters**

```
сору
```

Reimplemented from epiworld::Model < EPI\_DEFAULT\_TSEQ >.

#### 14.47.2.2 initial\_states()

Set the initial states of the model.

#### **Parameters**

proportions⊷	Double vector with a single element:
_	The proportion of non-infected individuals who have recovered.

 $\label{eq:control_equation} \mbox{Reimplemented from epiworld::} \mbox{Model} < \mbox{EPI\_DEFAULT\_TSEQ} >.$ 

#### 14.47.2.3 reset()

```
template<typename TSeq >
void ModelSIRMixing< 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

## 14.48 ModelSIRMixing < TSeq > Class Template Reference

Inheritance diagram for ModelSIRMixing < TSeq >:



Collaboration diagram for ModelSIRMixing< TSeq >:



#### **Public Member Functions**

ModelSIRMixing (ModelSIRMixing
 TSeq > &model, std::string vname, epiworld\_fast\_uint n, epiworld
 \_double prevalence, epiworld\_double contact\_rate, epiworld\_double transmission\_rate, epiworld\_double recovery\_rate, std::vector< double > contact\_matrix)

Constructs a ModelSIRMixing object.

ModelSIRMixing (std::string vname, epiworld\_fast\_uint n, epiworld\_double prevalence, epiworld\_double contact\_rate, epiworld\_double transmission\_rate, epiworld\_double recovery\_rate, std::vector< double > contact\_matrix)

Constructs a ModelSIRMixing object.

ModelSIRMixing< 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.

ModelSIRMixing< TSeq > & initial\_states (std::vector< double > proportions\_, std::vector< int > queue ← \_={})

Set the initial states of the model.

- size\_t get\_n\_infected (size\_t group) const
- void set\_contact\_matrix (std::vector< double > cmat)

#### **Static Public Attributes**

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

#### **Additional Inherited Members**

#### 14.48.1 Constructor & Destructor Documentation

#### 14.48.1.1 ModelSIRMixing() [1/2]

#### Constructs a ModelSIRMixing object.

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

#### **Parameters**

model	A reference to an existing ModelSIRMixing object.	
vname	The name of the ModelSIRMixing object.	
п	The number of entities in the model.	
prevalence	The initial prevalence of the disease in the model.	
contact_rate	The contact rate between entities in the model.	
transmission_rate	The transmission rate of the disease in the model.	
recovery_rate	The recovery rate of the disease in the model.	
contact_matrix	The contact matrix between entities in the model.	
model	A Model <tseq> object where to set up the SIR.</tseq>	
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 Generated by Doxygen	Probability of recovery	

#### 14.48.1.2 ModelSIRMixing() [2/2]

Constructs a ModelSIRMixing object.

#### **Parameters**

vname	The name of the ModelSIRMixing object.
n	The number of entities in the model.
prevalence	The initial prevalence of the disease in the model.
contact_rate	The contact rate between entities in the model.
transmission_rate	The transmission rate of the disease in the model.
recovery_rate	The recovery rate of the disease in the model.
contact_matrix	The contact matrix between entities in the model.

#### 14.48.2 Member Function Documentation

#### 14.48.2.1 clone\_ptr()

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

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

#### **Parameters**



 $\label{eq:local_problem} \mbox{Reimplemented from epiworld::} \mbox{Model} < \mbox{EPI\_DEFAULT\_TSEQ} >.$ 

#### 14.48.2.2 initial\_states()

```
template<typename TSeq >
ModelSIRMixing< TSeq > & ModelSIRMixing< 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 >.

#### 14.48.2.3 reset()

```
template<typename TSeq >
void ModelSIRMixing< 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:

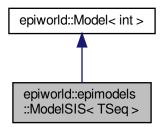
• include/epiworld/models/sirmixing.hpp

# 14.49 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**

## 14.49.1 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:

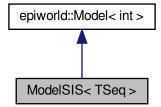
· epiworld.hpp

## 14.50 ModelSIS< TSeq > Class Template Reference

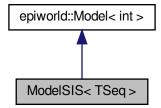
Template for a Susceptible-Infected-Susceptible (SIS) model.

#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**

#### 14.50.1 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

# 14.51 epiworld::epimodels::ModelSISD< TSeq > Class Template Reference

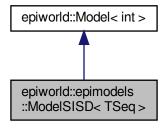
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

```
#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**

#### 14.51.1 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:

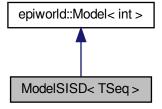
· epiworld.hpp

## 14.52 ModelSISD< TSeq > Class Template Reference

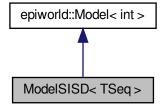
Template for a Susceptible-Infected-Susceptible-Deceased (SISD) model.

#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**

## 14.52.1 Detailed Description

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

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

#### **Parameters**

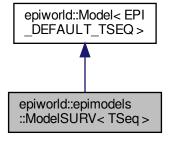
vname	std::string Name of the virus
initial_prevalence	epiworld_double Initial prevalence
initial_efficacy	epiworld_double Initial susceptibility_reduction of the immune system
initial_recovery	epiworld_double Initial recovery_rate rate of the immune system
inital_death	epiworld_double Initial death_rate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

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

• include/epiworld/models/sisd.hpp

# 14.53 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.	
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\_tob=0.001, epiworld\_double prob\_transmission=1.0, epiworld\_double prob\_death=0.001, epiworld\_tob=0.001, epiworld\_tob=0.001
- **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)

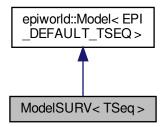
#### **Additional Inherited Members**

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

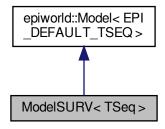
· epiworld.hpp

## 14.54 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.	
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

# 14.55 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

## 14.56 epiworld::PersonTools < TSeq > Class Template Reference

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

epiworld.hpp

## 14.57 PersonTools < TSeq > Class Template Reference

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

· include/epiworld/config.hpp

## 14.58 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 ()

#### 14.58.1 Detailed Description

A simple progress bar.

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

· epiworld.hpp

## 14.59 Progress Class Reference

A simple progress bar.

```
#include cpress.hpp>
```

#### **Public Member Functions**

- Progress (int n\_, int width\_)
- void start ()
- · void next ()
- void **end** ()

#### 14.59.1 Detailed Description

A simple progress bar.

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

• include/epiworld/progress.hpp

## 14.60 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 >

#### 14.60.1 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** 



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

· epiworld.hpp

## 14.61 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 >

## 14.61.1 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

## 14.62 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

## 14.63 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

#### 14.64 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

## 14.65 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**

٧	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)

#### 14.65.1 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

## 14.66 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**

٧	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)

#### 14.66.1 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

## 14.67 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 >

#### 14.67.1 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

## 14.68 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 >

#### 14.68.1 Detailed Description

template<typename TSeq> class Tools< TSeq>

Set of tools (useful for building iterators)

**Template Parameters** 



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

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

## 14.69 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 >

#### 14.69.1 Detailed Description

template<typename TSeq>
class epiworld::Tools\_const< TSeq >

Set of Tools (const) (useful for iterators)

**Template Parameters** 



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

· epiworld.hpp

## ${\bf 14.70 \quad Tools\_const} < {\bf TSeq} > {\bf Class\ Template\ Reference}$

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 >

#### 14.70.1 Detailed Description

template<typename TSeq> class Tools\_const< TSeq >

Set of Tools (const) (useful for iterators)

**Template Parameters** 



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

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

## 14.71 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**

Х	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**

	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  ${\bf DataBase}{<}{\,{\sf TSeq}}{\,>}$

#### 14.71.1 Detailed Description

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

Personalized data by the user.

**Template Parameters** 

TSeq	

#### 14.71.2 Constructor & Destructor Documentation

#### 14.71.2.1 UserData()

Construct a new User Data object.

#### **Parameters**

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

· epiworld.hpp

## 14.72 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<</li>
   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**

Х	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 >

## 14.72.1 Detailed Description

```
template<typename TSeq> class UserData< TSeq>
```

Personalized data by the user.

**Template Parameters** 

```
TSeq
```

#### 14.72.2 Constructor & Destructor Documentation

## 14.72.2.1 UserData()

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

## 14.73 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

#### 14.73.1 Detailed Description

 $\label{template} $$\operatorname{template} < \operatorname{typename} T > $$\operatorname{truct} \operatorname{epiworld}::\operatorname{vecHasher} < T > $$$ $$\operatorname{Vector} \ hasher.$ 

**Template Parameters** 

T

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

epiworld.hpp

## 14.74 vecHasher < T > Struct Template Reference

Vector hasher.

#include <misc.hpp>

#### **Public Member Functions**

std::size\_t operator() (std::vector< T > const &dat) const noexcept

#### 14.74.1 Detailed Description

template<typename T> struct vecHasher< T>

Vector hasher.

#### **Template Parameters**



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

· include/epiworld/misc.hpp

## 14.75 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)

#### 14.75.1 Detailed Description

 $\label{template} \begin{tabular}{ll} template < typename TSeq > \\ class epiworld:: Virus < TSeq > \\ \end{tabular}$ 

#### 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

## 14.76 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**

٧	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)

#### 14.76.1 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

## 14.77 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 >

#### 14.77.1 Detailed Description

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

Set of viruses (useful for building iterators)

**Template Parameters** 



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

· epiworld.hpp

## 14.78 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**

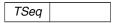
- class Virus < TSeq >
- class Agent < TSeq >

#### 14.78.1 Detailed Description

template<typename TSeq> class Viruses< TSeq>

Set of viruses (useful for building iterators)

**Template Parameters** 



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

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

## 14.79 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 >

#### 14.79.1 Detailed Description

template<typename TSeq>
class epiworld::Viruses\_const< TSeq>

Set of Viruses (const) (useful for iterators)

**Template Parameters** 



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

· epiworld.hpp

## 14.80 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 >

## 14.80.1 Detailed Description

$$\label{template} \begin{split} & \mathsf{template}\!<\!\mathsf{typename}\;\mathsf{TSeq}\!> \\ & \mathsf{class}\;\mathsf{Viruses\_const}\!<\mathsf{TSeq}> \end{split}$$

Set of Viruses (const) (useful for iterators)

**Template Parameters** 



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:



208 File Documentation

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

Copyright (c) 2022

## Index

```
add globalevent
 Agent < TSeq >, 43
 epiworld::Model < TSeq >, 83
 Entity < TSeq >, 63
 Model < TSeq >, 98
 epiworld::Agent < TSeq >, 47
 epiworld::Entity< TSeq >, 65
AdjList, 37
 AdjList, 37
 Entities < TSeq >, 59
 epiworld::AdjList, 39
 Entities_const< TSeq >, 61
 read edgelist, 38
 Entity < TSeq >, 63
Agent < TSeq >, 40
 default_rm_entity, 63
 default_rm_entity, 43
 epiworld::AdjList, 38
 operator(), 42
 AdjList, 39
 swap neighbors, 43
 read edgelist, 39
AgentsSample
 epiworld::Agent < TSeq >, 44
 AgentsSample < TSeq >, 48
 default rm entity, 47
 epiworld::AgentsSample < TSeq >, 50
 operator(), 46
AgentsSample < TSeq >, 47
 swap neighbors, 46
 AgentsSample, 48
 epiworld::AgentsSample < TSeq >, 49
 AgentsSample, 50
clone ptr
 epiworld::epimodels::ModelSEIRCONN< TSeq >,
 epiworld::DataBase< TSeq >, 55
 generation_time, 57
 epiworld::epimodels::ModelSEIRDCONN< TSeq
 get_transmissions, 58
 >, 127
 operator==, 58
 epiworld::epimodels::ModelSEIRMixing< TSeq >,
 record virus, 58
 reproductive number, 59
 epiworld::epimodels::ModelSIRCONN< TSeq >,
 transition probability, 59
 epiworld::Entities < TSeq >, 60
 epiworld::epimodels::ModelSIRDCONN< TSeq >,
 epiworld::Entities_const< TSeq >, 62
 epiworld::Entity< TSeq >, 64
 epiworld::epimodels::ModelSIRLogit<
 TSea
 default rm entity, 65
 162
 epiworld::epimodels::ModelDiffNet< TSeq >, 104
 epiworld::epimodels::ModelSIRMixing< TSeq >,
 epiworld::epimodels::ModelSEIR< TSeq >, 107
 initial states, 109
 168
 epiworld::Model < TSeq >, 84
 update exposed seir, 109
 Model < TSeq >, 98
 update infected seir, 109
 ModelSEIRCONN < TSeq >, 117
 epiworld::epimodels::ModelSEIRCONN< TSeq >, 113
 ModelSEIRDCONN < TSeq >, 130
 clone ptr, 114
 ModelSEIRMixing< TSeq >, 137
 initial states, 115
 ModelSIRCONN< TSeq >, 148
 ModelSEIRCONN, 114
 ModelSIRDCONN < TSeq >, 158
 reset, 115
 ModelSIRLogit < TSeq >, 165
 epiworld::epimodels::ModelSEIRD< TSeq >, 119
 ModelSIRMixing< TSeq >, 172
 ModelSEIRD, 120, 121
 update_exposed_seir, 121
DataBase < TSeq >, 50
 epiworld::epimodels::ModelSEIRDCONN< TSeq >,
 generation time, 52
 125
 get_transmissions, 53
 clone ptr, 127
 operator==, 53, 54
 initial states, 127
 record virus, 54
 ModelSEIRDCONN, 126
 reproductive number, 54
 reset, 127
 transition probability, 55
 epiworld::epimodels::ModelSEIRMixing< TSeq >, 131
default_rm_entity
 clone_ptr, 134
```

initial_states, 134 ModelSEIRMixing, 132, 133 reset, 134 epiworld::epimodels::ModelSIR < TSeq >, 139	epiworld::sampler, 29 make_sample_virus_neighbors, 29 make_update_susceptible, 30 sample_virus_single, 30
initial_states, 140	epiworld::SAMPLETYPE, 187
epiworld::epimodels::ModelSIRCONN< TSeq >, 143	epiworld::Tool< TSeq >, 187
clone_ptr, 144	epiworld::Tools< TSeq >, 190
get_n_infected, 145	epiworld::Tools_const< TSeq >, 192
initial_states, 145	epiworld::UserData< TSeq >, 193
ModelSIRCONN, 144	UserData, 195
reset, 145	epiworld::vecHasher $<$ T $>$ , 197
epiworld::epimodels::ModelSIRD< TSeq >, 149	epiworld::Virus< TSeq >, 198
initial_states, 151	epiworld::Viruses< TSeq >, 202
ModelSIRD, 150	epiworld::Viruses_const< TSeq >, 203
epiworld::epimodels::ModelSIRDCONN< TSeq >, 154	Event
clone_ptr, 156	epiworld::Event< TSeq >, 67
ModelSIRDCONN, 155	Event< TSeq >, 68
reset, 156	Event< TSeq >, 67
epiworld::epimodels::ModelSIRLogit< TSeq >, 159	Event, 68
clone_ptr, 162	events_add
ModelSIRLogit, 161	epiworld::Model< TSeq >, 84
reset, 162	Model < TSeq >, 99
epiworld::epimodels::ModelSIRMixing< TSeq >, 166	events run
clone_ptr, 168	epiworld::Model< TSeq >, 85
initial_states, 169	Model < TSeq >, 99
ModelSIRMixing, 167, 168	
reset, 169	generation_time
epiworld::epimodels::ModelSIS< TSeq >, 173	DataBase < TSeq >, 52
epiworld::epimodels::ModelSISD< TSeq >, 176	epiworld::DataBase< TSeq >, 57
epiworld::epimodels::ModelSURV< TSeq >, 179	get_n_infected
epiworld::Event< TSeq >, 65	epiworld::epimodels::ModelSIRCONN< TSeq >,
Event, 67	145
epiworld::GlobalEvent< TSeq >, 69	ModelSIRCONN< TSeq >, 148
GlobalEvent, 70	get_transmissions
epiworld::GroupSampler< TSeq >, 71	DataBase< TSeq >, 53
epiworld::LFMCMC< TData >, 72	epiworld::DataBase< TSeq >, 58
•	GlobalEvent
epiworld::Model < TSeq >, 75 add_globalevent, 83	epiworld::GlobalEvent< TSeq >, 70
— <del>-</del>	GlobalEvent< TSeq >, 71
clone_ptr, 84	GlobalEvent< TSeq >, 70
events_add, 84	GlobalEvent, 71
events_run, 85	GroupSampler < TSeq >, 72
initial_states_fun, 88	Groupoampier Toeq >, 72
load_agents_entities_ties, 85	include/epiworld/agent-meat-state.hpp, 207
rbinomd, 88	initial states
reset, 85	epiworld::epimodels::ModelSEIR< TSeq >, 109
rexpd, 88	epiworld::epimodels::ModelSEIRCONN< TSeq >,
rgammad, 88	115
rlognormald, 88	epiworld::epimodels::ModelSEIRDCONN< TSeq
rnormd, 89	>, 127
run_multiple, 86	epiworld::epimodels::ModelSEIRMixing< TSeq >,
runifd, 89	
set_agents_data, 86	134 epiworld::epimodels::ModelSIR< TSeq >, 140
set_name, 87	·
time_elapsed, 89	epiworld::epimodels::ModelSIRCONN< TSeq >,
write_data, 87	145
epiworld::PersonTools< TSeq >, 183	epiworld::epimodels::ModelSIRD< TSeq >, 151
epiworld::Progress, 184	epiworld::epimodels::ModelSIRMixing< TSeq >,
epiworld::Queue < TSeq >, 185	169 M - 105 ID + TO - + + + + + + + + + + + + + + + + + +
	ModelSEIR< TSeq >, 111

ModelSEIRCONN< TSeq >, 118 ModelSEIRDCONN< TSeq >, 130	ModelSEIRD < TSeq >, 122 ModelSEIRD, 123, 124
ModelSEIRMixing< TSeq >, 138	update_exposed_seir, 124
ModelSIR< TSeq >, 142	ModelSEIRDCONN
ModelSIRCONN< TSeq >, 148	epiworld::epimodels::ModelSEIRDCONN< TSeq
ModelSIRD< TSeq >, 153	>, 126
ModelSIRMixing< TSeq >, 172	ModelSEIRDCONN< TSeq >, 129
initial_states_fun	ModelSEIRDCONN< TSeq >, 128
epiworld::Model< TSeq >, 88	clone_ptr, 130
Model < TSeq >, 102	initial_states, 130
	ModelSEIRDCONN, 129
LFMCMC < TData >, 74	reset, 131
load_agents_entities_ties	ModelSEIRMixing
epiworld::Model < TSeq >, 85	${\it epiworld::epimodels::} Model SEIRMixing < TSeq >,$
Model < TSeq >, 99	132, 133
maka aamala virua najahbara	ModelSEIRMixing< TSeq >, 136, 137
make_sample_virus_neighbors	ModelSEIRMixing< TSeq >, 135
epiworld::sampler, 29	clone_ptr, 137
sampler, 32	initial_states, 138
make_update_susceptible	ModelSEIRMixing, 136, 137
epiworld::sampler, 30	reset, 138
sampler, 33	ModelSIR< TSeq >, 140
Model < TSeq >, 90	initial_states, 142
add_globalevent, 98	ModelSIRCONN
clone_ptr, 98	epiworld::epimodels::ModelSIRCONN $<$ TSeq $>$ ,
events_add, 99	144
events_run, 99	ModelSIRCONN< TSeq >, 147
initial_states_fun, 102	ModelSIRCONN< TSeq >, 146
load_agents_entities_ties, 99	clone_ptr, 148
rbinomd, 102	get_n_infected, 148
reset, 100	initial_states, 148
rexpd, 102	ModelSIRCONN, 147
rgammad, 102 rlognormald, 103	reset, 149
rnormd, 103	ModelSIRD
run_multiple, 100	epiworld::epimodels::ModelSIRD< TSeq >, 150
runifd, 103	ModelSIRD< TSeq >, 153
set_agents_data, 101	ModelSIRD< TSeq >, 151
set name, 101	initial_states, 153
time_elapsed, 103	ModelSIRD, 153
write_data, 101	ModelSIRDCONN
ModelDiffNet< TSeq >, 105	epiworld::epimodels::ModelSIRDCONN< TSeq >,
ModelSEIR< TSeq >, 110	155
initial_states, 111	ModelSIRDCONN< TSeq >, 158
update_exposed_seir, 112	ModelSIRDCONN < TSeq >, 157
update_infected_seir, 112	clone_ptr, 158
ModelSEIRCONN	ModelSIRDCONN, 158
epiworld::epimodels::ModelSEIRCONN< TSeq >,	reset, 159
114	ModelSIRLogit
ModelSEIRCONN< TSeq >, 117	epiworld::epimodels::ModelSIRLogit< TSeq >,
ModelSEIRCONN< TSeq >, 116	161
clone_ptr, 117	ModelSIRLogit < TSeq >, 164
initial_states, 118	ModelSIRLogit < TSeq >, 163
ModelSEIRCONN, 117	clone_ptr, 165
reset, 118	ModelSIRLogit, 164
ModelSEIRD	reset, 165
epiworld::epimodels::ModelSEIRD< TSeq >, 120,	ModelSIRMixing
121	epiworld::epimodels::ModelSIRMixing< TSeq >,
ModelSEIRD< TSeq >, 123, 124	167, 168

ModelSIRMixing< TSeq >, 171, 172 ModelSIRMixing< TSeq >, 170	ModelSIRLogit < TSeq >, 165 ModelSIRMixing < TSeq >, 173
clone_ptr, 172	rexpd
initial_states, 172	epiworld::Model < TSeq >, 88
ModelSIRMixing, 171, 172	Model < TSeq >, 102
reset, 173	rgammad
ModelSIS < TSeq >, 175	epiworld::Model< TSeq >, 88
ModelSISD< TSeq >, 178	Model < TSeq >, 102
ModelSURV < TSeq >, 181	rlognormald
Naturalis ( Nathura - Nadatura - Eductura > 100	epiworld::Model < TSeq >, 88
Network< Nettype, Nodetype, Edgetype >, 183	Model < TSeq >, 103
operator/\	rnormd
operator()	epiworld::Model < TSeq >, 89
Agent < TSeq >, 42	Model < TSeq >, 103
epiworld::Agent < TSeq >, 46	run_multiple
operator==	epiworld::Model < TSeq >, 86
DataBase < TSeq >, 53, 54	Model < TSeq >, 100
epiworld::DataBase< TSeq >, 58	runifd
PorconTools / TSog > 194	epiworld::Model < TSeq >, 89
PersonTools < TSeq >, 184	Model < TSeq >, 103
Progress, 184	
Queue < TSeq >, 185	sample_virus_single
Quoue < 100q >, 100	epiworld::sampler, 30
RandGraph, 186	sampler, 33
rbinomd	sampler, 32
epiworld::Model< TSeq >, 88	make_sample_virus_neighbors, 32
Model< TSeq >, 102	make_update_susceptible, 33
read_edgelist	sample_virus_single, 33
AdjList, 38	SAMPLETYPE, 187
epiworld::AdjList, 39	set_agents_data
record_virus	epiworld::Model < TSeq >, 86
DataBase < TSeq >, 54	Model < TSeq >, 101
epiworld::DataBase< TSeq >, 58	set_name
reproductive_number	epiworld::Model < TSeq >, 87
DataBase< TSeq >, 54	Model < TSeq >, 101
epiworld::DataBase< TSeq >, 59	swap_neighbors
reset	Agent < TSeq >, 43
epiworld::epimodels::ModelSEIRCONN< TSeq >,	epiworld::Agent < TSeq >, 46
115	
epiworld::epimodels::ModelSEIRDCONN< TSeq	time_elapsed
>, 127	epiworld::Model < TSeq >, 89
epiworld::epimodels::ModelSEIRMixing< TSeq >,	Model < TSeq >, 103
134	Tool < TSeq >, 188
epiworld::epimodels::ModelSIRCONN< TSeq >,	Tools< TSeq >, 191
145	Tools_const< TSeq >, 192
epiworld::epimodels::ModelSIRDCONN< TSeq >,	transition_probability
156	DataBase< TSeq >, 55
	epiworld::DataBase< TSeq >, 59
epiworld::epimodels::ModelSIRLogit< TSeq >,	
	update_exposed_seir
epiworld::epimodels::ModelSIRMixing< TSeq >, 169	epiworld::epimodels::ModelSEIR< TSeq >, 109
	epiworld::epimodels::ModelSEIRD< TSeq >, 12
epiworld::Model < TSeq >, 85	ModelSEIR< TSeq >, 112
Model < TSeq >, 100	ModelSEIRD< TSeq >, 124
ModelSEIRCONN < TSeq >, 118	update_infected_seir
ModelSEIRDCONN< TSeq >, 131	epiworld::epimodels::ModelSEIR< TSeq >, 109
ModelSEIRMixing < TSeq >, 138	ModelSEIR< TSeq >, 112
ModelSIRCONN < TSeq >, 149	UserData
ModelSIRDCONN< TSeq >, 159	epiworld::UserData< TSeq >, 195

```
UserData< TSeq >, 196
UserData< TSeq >, 195
UserData, 196

vecHasher< T >, 197
Virus< TSeq >, 200
Viruses< TSeq >, 203
Viruses_const< TSeq >, 204

write_data
 epiworld::Model< TSeq >, 87
Model< TSeq >, 101
```