epiworld

0.0-1

Generated by Doxygen 1.9.1

Chapter 1

Main Page

1.1 epiworld

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.

1.2 Hello world

Here is a simple SIRS model implemented with

```
#include "../include/epiworld/epiworld.hpp"
using namespace epiworld;
int main()
{
        // Creating a model
    Model<> model;
    // Adding the tool and virus
    Virus<> virus("covid 19");
    virus.set_post_immunity(1.0);
    model.add_virus_n(virus, 5);

    Tool<> tool("vaccine");
    model.add_tool(tool, .5);
    // Generating a random pop
    model.pop_from_random(100000);
    // Initializing setting days and seed
    model.init(100, 123);
```

```
    // Running the model
    model.run();
    model.print();

}
```

And you should get something like the following:

```
Running the model...
_____________________________________________________________________________
|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| done.
_____________________________________________________________________________
SIMULATION STUDY
Population size    : 100000
Days (duration)    : 100 (of 100)
Number of variants : 1
Last run elapsed t : 280.00ms
Rewiring           : off
Virus(es):
 - covid 19 (baseline prevalence: 5 seeds)
Tool(s):
 - vaccine (baseline prevalence: 50.00%)
Model parameters:
Distribution of the population at time 100:
 - Total healthy (S)   :   99995 -> 97390
 - Total recovered (S) :       0 -> 2554
 - Total infected (I)  :       5 -> 56
 - Total removed (R)   :       0 -> 0
(S): Susceptible, (I): Infected, (R): Recovered
_____________________________________________________________________________
```

Which took about 0.280 seconds.

1.2.1 Tools

1.2.2 Contagion

Susceptible individuals can acquire a virus from any of their infected connections. The probability that susceptible individual `i` gets the virus `v` from individual `j` depends on how three things:

1. The transmissibility of the virus, Pv in [0,1],
2. The contagion reduction factor of `i`, Cr in [0,1], and
3. The host's transmission reduction factor, Tr [0,1].

The last two are computed from `i` and `j`'s tools. Ultimately, the probability of `i` getting virus `v` from `j` equals:

```
P(Virus v) = Pv * (1 - Cr) *(1 - Tr)
```

Nonetheless, the default behavior of the simulation model is to assume that individuals can acquire only one disease at a time, if any. This way, the actual probability is:

```
P(Virus v| at most one virus) = Prcond(i,v,j)
```

The latter is calculated using Bayes' rule

```
Prcond(i,v,j) = P(at most one virus|Virus v) * P(Variant v) / P(at most one virus)
              = P(Only Virus v)/P(Virus v) * P(Virus v) / P(at most one virus)
              = P(Only Virus v)/P(at most one virus)
```

Where

```
P(Vnly Virus V)      = P(Virus V) * Prod(m!=V) (1 - P(Virus m))
P(at most one virus) = P(None) + Sum(k in viruses) P(Virus k) * Prod(m != k) (1 - P(Virus m))
P(None)              = Prod(k in Viruses) (1 - P(Virus k))
```

This way, viruses with higher transmissibility will be more likely to be acquired when competing with other variants.

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 AdjList Class Reference

Public Member Functions

- AdjList (const std::vector< unsigned int > &source, const std::vector< unsigned int > &target, bool directed, int min_id=-1, int max_id=-1)

  Construct a new Adj List object.
- void **read_edgelist** (std::string fn, int skip=0, bool directed=true, int min_id=-1, int max_id=-1)
- std::map< unsigned int, unsigned int > **operator()** (unsigned int i) const
- void **print** (unsigned int limit=20u) const
- unsigned int **get_id_max** () const
- unsigned int **get_id_min** () const
- size_t **vcount** () const
- size_t **ecount** () const
- std::map< unsigned int, std::map< unsigned int, unsigned int > > & **get_dat** ()
- bool **is_directed** () const

3.1.1 Constructor & Destructor Documentation

3.1.1.1 AdjList()

```
AdjList::AdjList (
            const std::vector< unsigned int > & source,
            const std::vector< unsigned int > & target,
            bool directed,
            int min_id = -1,
            int max_id = -1 )  [inline]
```

Construct a new Adj List object.

It will create an adjacency list object with `maxid - minid + 1` nodes. If min_id and max_id are not specified (both < 0), then the program will try to figure them out automatically by looking at the range of the observed ids.

Parameters

source	Unsigned int vector with the source
target	Unsigned int vector with the target
directed	Bool true if the network is directed
min_id	int min id.
max_id	int max id.

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

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

3.2 DataBase< TSeq > Class Template Reference

Statistical data about the process.

`#include <database-bones.hpp>`

Public Member Functions

- **DataBase** (int freq=1)
- void record_variant (Virus< TSeq > *v)

  Registering a new variant.
- void **set_seq_hasher** (std::function< std::vector< int >(TSeq)> fun)
- void **set_model** (Model< TSeq > &m)
- Model< TSeq > * **get_model** ()
- void **record** ()
- const std::vector< TSeq > & **get_sequence** () const
- const std::vector< int > & **get_nexposed** () const
- size_t **size** () const
- void **up_exposed** (Virus< TSeq > *v, epiworld_fast_uint new_status)
- void **down_exposed** (Virus< TSeq > *v, epiworld_fast_uint prev_status)
- void **state_change** (epiworld_fast_uint prev_status, epiworld_fast_uint new_status)
- void **record_transition** (epiworld_fast_uint from, epiworld_fast_uint to)
- void **write_data** (std::string fn_variant_info, std::string fn_variant_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition) const
- void **record_transmission** (int i, int j, int variant)
- size_t **get_nvariants** () const
- void **reset** ()
- void **set_user_data** (std::vector< std::string > names)
- void **add_user_data** (std::vector< epiworld_double > x)
- void **add_user_data** (unsigned int j, epiworld_double x)
- UserData< TSeq > & **get_user_data** ()

Get recorded information from the model

Parameters

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

Returns

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

In `get_today_variant`*, the current counts of* `what` *for each variant.*

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

In `get_hist_variant`*, the time series of what for each variant.*

In `get_hist_total_date` *and* `get_hist_variant_date` *the corresponding dates*

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

Friends

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

3.2.1 Detailed Description

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

Statistical data about the process.

Template Parameters

TSeq	

3.2.2 Member Function Documentation

3.2.2.1 record_variant()

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

Registering a new variant.

Parameters

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

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

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

3.3 LFMCMC< TData > Class Template Reference

Likelihood-Free Markov Chain Monte Carlo.

`#include <lfmcmc.hpp>`

Public Member Functions

- void **run** (VEC(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** (FUN< void(VEC(epiworld_double)&, LFMCMC< TData > *)> fun)
- void **set_simulation_fun** (FUN< TData(VEC(epiworld_double)&, LFMCMC< TData > *)> fun)
- void **set_summary_fun** (FUN< VEC(epiworld_double)(TData &, LFMCMC< TData > *)> fun)
- void **set_kernel_fun** (FUN< epiworld_double(VEC(epiworld_double)&, epiworld_double, LFMCMC< TData > *)> fun)
- const size_t **get_n_samples** ()
- const size_t **get_n_statistics** ()
- const size_t **get_n_parameters** ()
- const epiworld_double **get_epsilon** ()
- const **VEC** (epiworld_double) &get_params_now()
- const **VEC** (epiworld_double) &get_params_prev()
- const **VEC** (epiworld_double) &get_params_init()
- const **VEC** (epiworld_double) &get_statistics_obs()
- const **VEC** (epiworld_double) &get_statistics_hist()
- const **VEC** (bool) &get_statistics_accepted()
- const **VEC** (epiworld_double) &get_posterior_lf_prob()
- const **VEC** (epiworld_double) &get_acceptance_prob()
- const **VEC** (epiworld_double) &get_drawn_prob()
- **VEC** (TData) *get_sampled_data()

Random number generation

Parameters

eng	

- void **set_rand_engine** (std::mt19937 &eng)

- std::mt19937 * **get_rand_endgine** ()
- void **seed** (unsigned int s)
- void **set_rand_gamma** (epiworld_double alpha, epiworld_double beta)
- epiworld_double **runif** ()
- epiworld_double **rnorm** ()
- epiworld_double **rnorm** (epiworld_double mean, epiworld_double sd)
- epiworld_double **rgamma** ()
- epiworld_double **rgamma** (epiworld_double alpha, epiworld_double beta)

3.3.1 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 file:

- include/epiworld/math/lfmcmc.hpp

3.4 Location< TSeq > Class Template Reference

Public Member Functions

- **add_person** (Person< TSeq > &p)
- **add_person** (Person< TSeq > *p)
- size_t **count** () const
- void **reset** ()

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

- include/epiworld/location-bones.hpp

3.5 Model< TSeq > Class Template Reference

Core class of epiworld.

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

Public Member Functions

- **Model** (const Model< TSeq > &m)
- **Model** (Model< TSeq > &&m)
- Model< TSeq > & **operator=** (const Model< TSeq > &m)
- void **clone_population** (std::vector< Person< TSeq > > &p, std::map< int, int > &p_ids, bool &d, Model< TSeq > *m=nullptr) const
- void **clone_population** (const Model< TSeq > &m)
- DataBase< TSeq > & **get_db** ()
- epiworld_double & **operator()** (std::string pname)
- size_t **size** () const
- void **add_virus** (Virus< TSeq > v, epiworld_double preval)
- void **add_virus_n** (Virus< TSeq > v, unsigned int preval)
- void **add_tool** (Tool< TSeq > t, epiworld_double preval)
- void **add_tool_n** (Tool< TSeq > t, unsigned int preval)
- void **record_variant** (Virus< TSeq > *v)
- int **get_nvariants** () const
- unsigned int **get_ndays** () const
- unsigned int **get_n_replicates** () const
- void **set_ndays** (unsigned int ndays)
- bool **get_verbose** () const
- void **verbose_off** ()
- void **verbose_on** ()
- int today () const

  The current time of the model.
- void **set_update_susceptible** (UpdateFun< TSeq > fun)
- void **set_update_exposed** (UpdateFun< TSeq > fun)
- void **set_update_removed** (UpdateFun< TSeq > fun)
- void write_data (std::string fn_variant_info, std::string fn_variant_hist, std::string fn_total_hist, std::string fn_transmission, std::string fn_transition) const

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

  Reset the model.
- void **print** () const
- Model< TSeq > && **clone** () const
- void reset_status_codes (std::vector< epiworld_fast_uint > codes, std::vector< std::string > names, bool verbose=true)

  Reset all the status codes of the model.
- void **get_elapsed** (std::string unit="auto", epiworld_double *last_elapsed=nullptr, epiworld_double *total_elapsed=nullptr, std::string *unit_abbr=nullptr, bool print=true) const
- void add_global_action (std::function< void(Model< TSeq > *)> fun, int date)

  Set a global action.
- void **run_global_actions** ()
- void **clear_status_set** ()
- void **toggle_visited** ()

Set the backup object

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

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

Random number generation

Parameters

eng	

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

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.*
min_id	*int Minimum id number (if negative, the program will try to guess from the data.)*
max_id	*int Maximum id number (if negative, the program will try to guess from the data.)*
al	*AdjList to read into the model.*

- void **pop_from_adjlist** (std::string fn, int skip=0, bool directed=false, int min_id=-1, int max_id=-1)
- void **pop_from_adjlist** (AdjList al)
- bool **is_directed** () const
- std::vector< Person< TSeq > > * **get_population** ()
- void **pop_from_random** (unsigned int n=1000, unsigned int k=5, bool d=false, epiworld_double p=.01)

Functions to run the model

Parameters

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

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

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

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< Person< TSeq >> *, Model< TSeq > *, epiworld_double)> fun)
- void **set_rewire_prop** (epiworld_double prop)
- epiworld_double **get_rewire_prop** () const
- void **rewire** ()

Export the network data in edgelist form

Parameters

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

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

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

Manage status (states) in the model

Adding values of `s` that are already present in the model will result in an error.

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

Parameters

s	`unsigned int` *Code of the status*
lab	`std::string` *Name of the status.*

Returns

`add_status*` *returns nothing.*

`get_status_*` *returns a vector of pairs with the statuses and their labels.*

- void **add_status_susceptible** (epiworld_fast_uint s, std::string lab)
- void **add_status_exposed** (epiworld_fast_uint s, std::string lab)
- void **add_status_removed** (epiworld_fast_uint s, std::string lab)
- void **add_status_susceptible** (std::string lab)
- void **add_status_exposed** (std::string lab)
- void **add_status_removed** (std::string lab)
- const std::vector< epiworld_fast_uint > & **get_status_susceptible** () const
- const std::vector< epiworld_fast_uint > & **get_status_exposed** () const
- const std::vector< epiworld_fast_uint > & **get_status_removed** () const
- const std::vector< std::string > & **get_status_susceptible_labels** () const
- const std::vector< std::string > & **get_status_exposed_labels** () const
- const std::vector< std::string > & **get_status_removed_labels** () const
- void **print_status_codes** () const
- epiworld_fast_uint **get_default_susceptible** () const
- epiworld_fast_uint **get_default_exposed** () const

- epiworld_fast_uint **get_default_removed** () const

Set the user data object

Parameters

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

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

Queuing system

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

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

Public Attributes

- std::vector< epiworld_double > **array_double_tmp**
- std::vector< Virus< TSeq > * > **array_virus_tmp**

Friends

- class **Person**< **TSeq** >
- class **DataBase**< **TSeq** >
- class **Queue**< **TSeq** >

Setting and accessing parameters from the model

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

The function `set_param()` can be used when the parameter already exists in the model.

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

Parameters

initial_val	
pname	Name of the parameter to add or to fetch

Returns

The current value of the parameter in the model.

- epiworld_double * **p0**
- epiworld_double * **p1**
- epiworld_double * **p2**
- epiworld_double * **p3**
- epiworld_double * **p4**
- epiworld_double * **p5**
- epiworld_double * **p6**
- epiworld_double * **p7**
- epiworld_double * **p8**
- epiworld_double * **p9**
- epiworld_double * **p10**
- epiworld_double * **p11**
- epiworld_double * **p12**
- epiworld_double * **p13**
- epiworld_double * **p14**
- epiworld_double * **p15**
- epiworld_double * **p16**
- epiworld_double * **p17**
- epiworld_double * **p18**
- epiworld_double * **p19**
- epiworld_double * **p20**
- epiworld_double * **p21**
- epiworld_double * **p22**
- epiworld_double * **p23**
- epiworld_double * **p24**
- epiworld_double * **p25**
- epiworld_double * **p26**
- epiworld_double * **p27**
- epiworld_double * **p28**
- epiworld_double * **p29**
- epiworld_double * **p30**
- epiworld_double * **p31**
- epiworld_double * **p32**
- epiworld_double * **p33**
- epiworld_double * **p34**
- epiworld_double * **p35**
- epiworld_double * **p36**
- epiworld_double * **p37**
- epiworld_double * **p38**
- epiworld_double * **p39**
- unsigned int **npar_used** = 0u
- epiworld_double **add_param** (epiworld_double initial_val, std::string pname)
- epiworld_double **set_param** (std::string pname)
- epiworld_double **get_param** (unsigned int k)
- epiworld_double **get_param** (std::string pname)
- epiworld_double **par** (unsigned int k)
- epiworld_double **par** (std::string pname)

3.5.1 Detailed Description

template<typename TSeq = bool>
class Model< TSeq >

Core class of epiworld.

The model class provides the wrapper that puts together `Person`, `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.)

3.5.2 Member Function Documentation

3.5.2.1 add_global_action()

```
template<typename TSeq = bool>
void Model< TSeq >::add_global_action (
             std::function< void(Model< TSeq > *)> fun,
             int date )
```

Set a global action.

Parameters

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

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

3.5.2.2 reset()

```
template<typename TSeq = bool>
void Model< TSeq >::reset ( )
```

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

3.5.2.3 reset_status_codes()

```
template<typename TSeq = bool>
void Model< TSeq >::reset_status_codes (
             std::vector< epiworld_fast_uint > codes,
             std::vector< std::string > names,
             bool verbose = true )
```

Reset all the status codes of the model.

The default values are those specified in the enum STATUS.

Parameters

codes	In the following order: Susceptible, Infected, Removed
names	Names matching the codes
verbose	When `true`, it will print the new mappings.

3.5.2.4 run_multiple()

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

Parameters

nexperiments	Multiple runs of the simulation

3.5.2.5 write_data()

```
template<typename TSeq = bool>
void Model< TSeq >::write_data (
             std::string fn_variant_info,
             std::string fn_variant_hist,
```

```
        std::string fn_total_hist,
        std::string fn_transmission,
        std::string fn_transition ) const
```

Wrapper of `DataBase::write_data`

Parameters

fn_variant_info	Filename. Information about the variant.
fn_variant_hist	Filename. History of the variant.
fn_total_hist	Filename. Aggregated history (status)
fn_transmission	Filename. Transmission history.
fn_transition	Filename. Markov transition history.

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

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

3.6 Person< TSeq > Class Template Reference

Public Member Functions

- void **init** (epiworld_fast_uint baseline_status)
- void **add_tool** (int d, Tool< TSeq > tool)
- void **add_virus** (Virus< TSeq > *virus)
- void **rm_virus** (Virus< TSeq > *virus)
- epiworld_double **get_susceptibility_reduction** (Virus< TSeq > *v)
- epiworld_double **get_transmission_reduction** (Virus< TSeq > *v)
- epiworld_double **get_recovery_enhancer** (Virus< TSeq > *v)
- epiworld_double **get_death_reduction** (Virus< TSeq > *v)
- int **get_id** () const
- unsigned int **get_index** () const
- std::mt19937 * **get_rand_endgine** ()
- Model< TSeq > * **get_model** ()
- Virus< TSeq > & **get_virus** (int i)
- PersonViruses< TSeq > & **get_viruses** ()
- Tool< TSeq > & **get_tool** (int i)
- PersonTools< TSeq > & **get_tools** ()
- void **mutate_variant** ()
- void **add_neighbor** (Person< TSeq > *p, bool check_source=true, bool check_target=true)
- std::vector< Person< TSeq > * > & **get_neighbors** ()
- void **update_status** ()
- void **update_status** (epiworld_fast_uint new_status)
- const epiworld_fast_uint & **get_status** () const
- const epiworld_fast_uint & **get_status_next** () const
- void **reset** ()
- void **set_update_susceptible** (UpdateFun< TSeq > fun)
- void **set_update_exposed** (UpdateFun< TSeq > fun)
- void **set_update_removed** (UpdateFun< TSeq > fun)
- bool **has_tool** (unsigned int t) const
- bool **has_tool** (std::string name) const
- bool **has_virus** (unsigned int t) const
- bool **has_virus** (std::string name) const
- bool **visited** () const
- void **toggle_visited** ()

Friends

- class **Model**< **TSeq** >
- class **Tool**< **TSeq** >
- class **Queue**< **TSeq** >

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

- include/epiworld/config.hpp
- include/epiworld/person-bones.hpp
- include/epiworld/person-meat.hpp

3.7 PersonTools< TSeq > Class Template Reference

List of tools available for the individual to.

`#include <persontools-bones.hpp>`

Public Member Functions

- void **add_tool** (int date, Tool< TSeq > tool)
- epiworld_double **get_susceptibility_reduction** (Virus< TSeq > *v)
- epiworld_double **get_transmission_reduction** (Virus< TSeq > *v)
- epiworld_double **get_recovery_enhancer** (Virus< TSeq > *v)
- epiworld_double **get_death_reduction** (Virus< TSeq > *v)
- 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)
- size_t **size** () const
- Tool< TSeq > & **operator()** (int i)
- Person< TSeq > * **get_person** ()
- Model< TSeq > * **get_model** ()
- void **reset** ()
- bool **has_tool** (unsigned int t) const
- bool **has_tool** (std::string name) const

Friends

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

3.7.1 Detailed Description

template<**typename TSeq = bool**>
class PersonTools< **TSeq** >

List of tools available for the individual to.

Template Parameters

TSeq	

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

- include/epiworld/config.hpp
- include/epiworld/persontools-bones.hpp
- include/epiworld/persontools-meat.hpp

3.8 PersonViruses< TSeq > Class Template Reference

Set of viruses in host.

`#include <personviruses-bones.hpp>`

Public Member Functions

- void **add_virus** (epiworld_fast_uint new_status, Virus< TSeq > v)
- size_t **size** () const
- int **size_active** () const
- Virus< TSeq > & **operator()** (int i)
- void **mutate** ()
- void **reset** ()
- void **deactivate** (Virus< TSeq > &v)
- Person< TSeq > * **get_host** ()
- bool **has_virus** (unsigned int v) const
- bool **has_virus** (std::string vname) const

Friends

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

3.8.1 Detailed Description

template<**typename TSeq = bool**>
class PersonViruses< **TSeq** >

Set of viruses in host.

Template Parameters

TSeq	Type of sequence

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

- include/epiworld/person-bones.hpp
- include/epiworld/personviruses-bones.hpp
- include/epiworld/personviruses-meat.hpp

3.9 Progress Class Reference

A simple progress bar.

`#include <progress.hpp>`

Public Member Functions

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

3.9.1 Detailed Description

A simple progress bar.

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

- include/epiworld/progress.hpp

3.10 Queue< TSeq > Class Template Reference

Controls which agents are verified at each step.

`#include <queue-bones.hpp>`

Public Member Functions

- void **operator+=** (Person< TSeq > *p)
- void **operator-=** (Person< TSeq > *p)
- epiworld_fast_int **operator[ ]** (unsigned int i) const
- void **set_model** (Model< TSeq > *m)
- void **update** ()

3.10.1 Detailed Description

template<typename TSeq = bool>
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/model-bones.hpp
- include/epiworld/queue-bones.hpp

3.11 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

3.12 Tool< TSeq > Class Template Reference

Tools for defending the host against the virus.

`#include <tools-bones.hpp>`

Public Member Functions

- **Tool** (std::string name="unknown tool")
- void **set_sequence** (TSeq d)
- void **set_sequence_unique** (TSeq d)
- void **set_sequence** (std::shared_ptr< TSeq > d)
- std::shared_ptr< TSeq > **get_sequence** ()
- TSeq & **get_sequence_unique** ()
- void **set_name** (std::string name)
- std::string **get_name** () const
- Person< TSeq > * **get_person** ()
- unsigned int **get_id** () const

Get and set the tool functions

Parameters

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

Returns

epiworld_double

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

Friends

- class **PersonTools**< **TSeq** >
- class **Person**< **TSeq** >
- class **Model**< **TSeq** >

3.12.1 Detailed Description

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

Tools for defending the host against the virus.

Template Parameters

TSeq	Type of sequence

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

- include/epiworld/config.hpp
- include/epiworld/tools-bones.hpp
- include/epiworld/tools-meat.hpp

3.13 UserData< TSeq > Class Template Reference

Public Member Functions

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

- epiworld_double & **operator()** (unsigned int i, unsigned int j)
- epiworld_double & **operator()** (unsigned int i, std::string name)
- std::vector< std::string > & **get_names** ()
- std::vector< int > & **get_dates** ()
- std::vector< epiworld_double > & **get_data** ()
- void **get_all** (std::vector< std::string > *names=nullptr, std::vector< int > *date=nullptr, std::vector< epiworld_double > *data=nullptr)
- unsigned int **nrow** () const
- unsigned int **ncol** () const
- void **write** (std::string fn)
- void **print** () const

Friends

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

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

3.14 vecHasher< T > Struct Template Reference

Public Member Functions

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

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

- include/epiworld/misc.hpp

3.15 Virus< TSeq > Class Template Reference

Virus.

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

Public Member Functions

- **Virus** (std::string name="unknown virus")
- void **mutate** ()
- void **set_mutation** (MutFun< TSeq > fun)
- const TSeq * **get_sequence** ()
- void **set_sequence** (TSeq sequence)
- Person< TSeq > * **get_host** ()
- Model< TSeq > * **get_model** ()
- void **set_date** (int d)
- int **get_date** () const
- void **set_id** (int idx)
- int **get_id** () const
- bool **is_active** () const
- void **deactivate** ()
- void **set_name** (std::string name)
- std::string **get_name** () const
- std::vector< epiworld_double > & **get_data** ()

Get and set the tool functions

Parameters

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

Returns

epiworld_double

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

Friends

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

3.15.1 Detailed Description

template<typename TSeq = bool>
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 host.

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

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

Index