epiworld: A fast and flexible ABM framework for epidemiological simulations

George G. Vega Yon, Ph.D.

2024-04-05

epiworld is a fast and flexible agent-based modeling (ABM) framework for epidemiological simulations. Designed in C++, it can simulate large populations with complex interactions. The framework is designed to be modular, allowing users to extend and modify the model to suit their needs quickly.

## Features

**Header-only**: epiworld is a header-only template library, making it easy to integrate into existing projects. It is distributed as a collection of files and as [a single header file](https://github.com/UofUEpiBio/epiworld/blob/master/epiworld.hpp).

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

**Fast**: epiworld is designed for speed. It is implemented in C++ and uses efficient data structures and algorithms to ensure simulations dash. Furthermore, epiworld is designed to take advantage of multi-core processors, allowing simulations to be run in parallel.

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

## Example

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

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

The output could look something like the following:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Running the model...  
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||| done.  
 done.  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
SIMULATION STUDY  
  
Name of the model : Susceptible-Infected-Recovered (SIR)  
Population size : 100000  
Agents' data : (none)  
Number of entities : 0  
Days (duration) : 100 (of 100)  
Number of viruses : 1  
Last run elapsed t : 103.00ms  
Last run speed : 96.34 million agents x day / second  
Rewiring : off  
  
Global events:  
 (none)  
  
Virus(es):  
 - COVID-19 (baseline prevalence: 1.00%)  
  
Tool(s):  
 (none)  
  
Model parameters:  
 - Recovery rate : 0.3000  
 - Transmission rate : 0.9000  
  
Distribution of the population at time 100:  
 - (0) Susceptible : 99000 -> 2565  
 - (1) Infected : 1000 -> 366  
 - (2) Recovered : 0 -> 97069  
  
Transition Probabilities:  
 - Susceptible 0.96 0.04 0.00  
 - Infected 0.00 0.70 0.30  
 - Recovered 0.00 0.00 1.00

## Appendix

### Implementation

Core components:

* Model class
* States
* Tool
* Virus
* Agdnt
* GlobalEvents
* DataBase