

EPYDEMIA: AN OPEN-SOURCE AGENT-BASED MODEL FOR INFECTIOUS DISEASE MODELING

Sebastian A. Rodriguez-Cartes

Edward P. Fitts Department of Industrial and Systems Engineering
North Carolina State University
915 Partners Way
Raleigh, NC 27606, USA

ABSTRACT

Agent-based models provide a flexible framework for the modeling of infectious diseases. We propose an open-source simulation framework, EPyDEMIA, that allows modeling multiple diseases infecting a population, implementing complex agent behaviors, and different interventions. The framework was designed as a discrete-event simulator and was implemented using Python. Infections throughout a population are driven using a network of multiple independent layers. We highlight the utility of our framework by showcasing a two-disease outbreak example. The proposed tool's modularity facilitates the implementation of disease transmission models, streamlining the analysis of the health impacts of infections.

1 INTRODUCTION

Recent challenges caused by the global Covid-19 pandemic have highlighted the need for effective tools that aid in forecasting and analyzing the impact of infectious diseases. Agent-based models have been widely used to model transmission dynamics within a population due to their ability to accurately represent the interactions between individuals that lead to epidemic outbreaks. Despite existing numerous simulation tools to build agent-based models, few of them allow for open-access while offering a simple and flexible framework (available open-access models include COVASIM, CppyABM, FRED). Hence, the objective of this study is to develop an open-source agent-based simulation library for the modeling of infectious diseases. We aim to contribute to the current state of knowledge by developing a framework flexible enough to incorporate complex interventions and behaviors, allowing the interaction of multiple diseases while maintaining a simple and intuitive interface for modelers to work with.

2 METHODS

The Epidemiological Python-based Disease Modeling and Infection Agent-based (EPyDEMIA) framework is an open-source simulation tool for infectious disease modeling. The library was implemented in Python as it is one of the leading open-source programming languages for scientific research. It allows for object-oriented programming, which facilitates the extension of models. The framework uses a discrete-event simulation approach. This paradigm allows users to define the desired complexity when modeling disease progression, interventions, and population dynamics. Interactions between agents are represented using a network of multiple independent layers supported by the `igraph` library (Csárdi and Nepusz 2006).

The simulation framework is driven by the `AgentBasedSim` class, which serves as a critical connection between simulation objects. This class holds the events scheduler and is responsible for executing them accordingly. Moreover, this class is meant to be the main connection to access different objects within the simulation. The `Disease` class defines a specific disease to be introduced among agents and requires the

user to at least define a probability of infection per contact β and the possible disease states that an agent could be. The framework allows for multiple diseases to be introduced simultaneously. Agents and their attributes are represented as arrays for computational efficiency. The `Population` class stores agents' information and handles interactions between them. Each instance is associated with an object from the `Network` class, which holds the information of contacts between agents. Multiple layers can be used to represent agents' connections across the population (i.e., households, community contacts, schools), with each `Layer` object containing independent graphs where agents are represented as vertices and their connections as edges. The possibility of an agent getting infected is modeled by defining β for each of its connections. Transmission of the disease is then driven by infectious agents being in contact with susceptible ones based on a user-defined `Step` (i.e., a daily step) event. The progression of the disease is modeled using sequential events defined by the user using the `ChangeStep` event type. When modeling multiple diseases, these events are responsible for handling their interactions. `Interventions` are also treated as events and are used to modify any simulation object, such as population attributes, disease attributes, and probability of infection.

3 TOY EXAMPLE: A MEASLES-RUBELLA OUTBREAK

To showcase the simulation framework, we present a toy example depicting the simultaneous outbreak of two highly infectious diseases: measles and rubella. Transmission probabilities were defined as 0.015 and 0.03, respectively, for illustrative purposes. A simple SEIR model was used to model the transmission dynamics of both diseases. The population size was 10,000 agents, and a single-layer network was created using the Barabasi algorithm with ten connections per node. We highlight the benefit of modeling both diseases together by analyzing the impact a quarantine intervention on measles has on the spreading of rubella. We assumed that an active case of measles had a probability p of being detected and sent to quarantine. Figure 1 shows the active cases of both diseases and the count of agents on quarantine for two different detection probabilities. As it is expected, the probability of detection has a significant impact on the peak of measles cases. The figure also shows that rubella cases are also affected, which would be more difficult to quantify if measles cases are not modeled jointly. Although this model is simple, it features the capacity of EPyDEMIA to develop simple models, which could be extended to incorporate more complex agent behaviors and interactions.

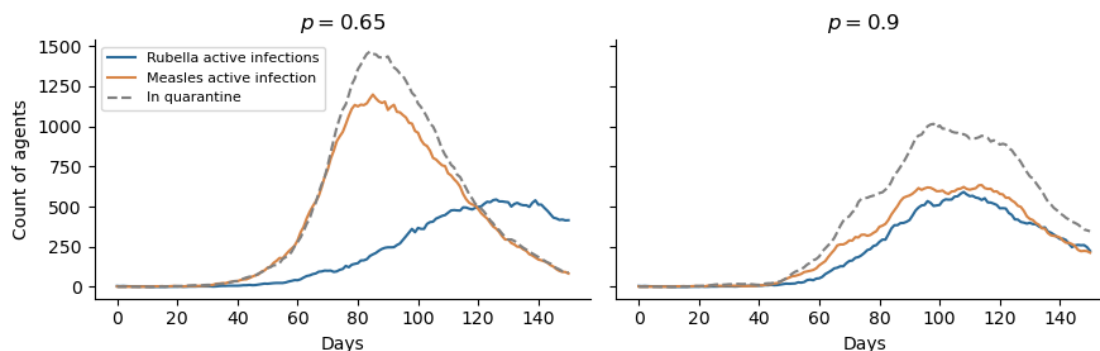


Figure 1: A measles and rubella model implemented with EPyDEMIA to quantify the impact of accurate detection of measles cases.

REFERENCES

Csárdi, G., and T. Nepusz. 2006. “The igraph Software Package for Complex Network Research”. *Inter-Journal Complex Systems* 1695.