

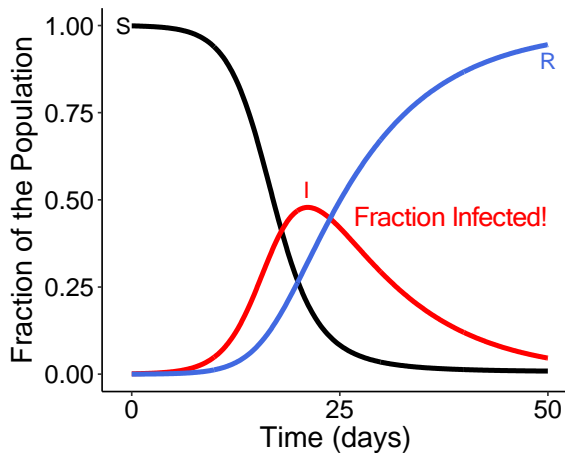
CS486C – Senior Capstone Design in Computer Science

Project Description

Project Title: A shared, interactive portal for exploring epidemiological model predictions	
Sponsor Information:	Dr. Joseph Mihaljevic, Assistant Professor School of Informatics, Computing, and Cyber Systems Email: joseph.mihaljevic@nau.edu Phone: (928) 523-5125

Project Overview:

Infectious disease is a primary threat to human, livestock, and wildlife health. Critically, new pathogens emerge all the time, and there is strong evidence that the rate of disease emergence has increased substantially over the last few decades. This poses a serious problem for epidemiologists, because critical decisions about how to mitigate disease must often be made with limited data and sometimes even limited understanding of how a pathogen is transmitted. For instance, during the recent Ebola virus outbreaks in western Africa, how the disease was spreading among populations was not well understood, and this likely resulted in a larger outbreak than if swift disease-control programs could have been implemented.



Mathematical models of how infectious diseases spread, known as *epidemiological models*, are vital tools in our goal of disease mitigation. These models typically predict how the number of infected individuals changes over time in a single population or across many populations in a region (e.g. spatial models). Models are regularly used, for example, to quantify what proportion of individuals need to be vaccinated to eliminate large outbreaks (e.g. childhood diseases), what factors cause diseases to cycle from year to year (e.g. seasonal influenza), or how many animals must be culled to stop an outbreak of disease in livestock to minimize economic losses (e.g. foot-and-mouth disease).

A persistent problem is that epidemiological models are not easily utilized by the general public or by disease management experts, for several reasons. First, models often keep track of many variables and how these variables change over time meaning that,

to make any sense of model output, one must select what variables one is interested and then display the result graphically (see graphic). In addition, some models have many parameters (e.g. assumptions about certain input values); others viewing the model will want to alter these parameters to understand how they affect the model predictions for their specific problem of interest. Another issue is that a model used to make predictions for one locality might need alternative input data to be useful in another locality, but this requires re-running the model. Finally, the code for many models is not open-source, and therefore attempting to reproduce model dynamics from verbal explanations of a model in the literature can be quite difficult. In short, there are *many* kinds of analyses/information that a given model can yield, meaning that researchers need to *interact* with models to explore their implications. In short, a model is much more than just a “snapshot” of some output; it is more a dynamic dataset that must be actively explored by viewers. **What is needed is an effective, secure information-sharing platform that provides researchers with the ability to make their models immediately accessible to the public and to disease management experts, provides all meta-information relevant to a model, and allows viewers to interact with the model to explore its implications graphically.**

Objective:

The objective of this project is to create a web-application that could serve as the cornerstone for an interactive community where researchers could share their models, and view and comment on models shared by others. In

particular, epidemiological researchers could register, search for interesting/relevant modeling results to view, could upload their own modeling results into the archive, and could share and discuss models in the archive with other users.

Basic set of features to even prove the concept (minimum viable product):

1. A secure, modern web application to host the information sharing portal, constructed using well-established technologies and frameworks, that allows uploading, searching, and viewing models by registered users.
2. A flexible architecture for hosting researchers' code for simulating their epidemiological models. The idea is that contributing researchers upload well-formatted "model driver modules" to the repository. Like records in a jukebox, these modules can then be indexed, searched, selected and "played" to create parameterized visualizations of the researcher's modeling results. The project sponsor can provide various examples of "models" to get the team started.
3. A well-designed graphical user interface that allows researchers to search for, select, and "play" the visualizations stored in the system. Running a visualization will also include (a) background information about their model, including equations, if necessary, (b) instructions for users, and (c) visualizations of their model output.
4. The GUI must provide an interactive exploratory experience driven by the stored "model driver modules": researchers depositing their model driver code can specify which parameters are adjustable (and value ranges); viewers can then change these parameters interactively to study changes in model output.
5. Handles code in C and R (two of the most common languages for epidemiological models), but the architecture is extensible to allow interpreters for other languages to be added later.

The basic system outlined above is essentially just a "plain old repository" and should be easily achievable; it is the minimum to prove the concept. To make this a truly useful basis for scientific interaction amongst epidemiologists, a number of additional features would need be added:

6. To create a basis for scientific interaction, the system must provide a secure user community, including: real user accounts with identify verification, user profiles, a strong permissions structure to allow submitters to control access to their submitted modules, and a "commentary" system that allows users to (a) contact others, e.g., contact the submitter of a module with questions, and (b) attach commentary/discussion threads to modules.
7. A more advanced system will support a wider array of coding languages for uploaded modules. Epidemiological models are written in many languages; the architecture should allow languages (e.g. C, C++, R, Python) to run and create visualizations. The architecture should support future "plug-ins" to be created for additional languages in future.

To support quick decision-making in rapidly-evolving disease outbreaks, scientists and policy-makers need a mechanism to quickly share and analyze models of disease transmission dynamics. This project will explore what such a platform would look like, and could have significant impact on the quality of disease management. If the prototype is successful, it might serve as a model for incorporation into the resource infrastructures provided by major organizations like the CDC and WHO.

Knowledge, skills, and expertise required for this project:

- Design and development of modern Web2.0 web applications. Familiarity with one or more webapp frameworks (e.g. Angular, Bootstrap, Django, etc.) would be useful.
- Basic database design, configuration, and installation knowledge to support the backend.
- Development of graphical user interfaces in web-based systems, including end-user testing/refinement.

Equipment Requirements:

- Ordinary development platform and software/tools freely available online.
- Client will provide access to appropriate datasets, models, and other domain-related elements needed.

Software and other Deliverables:

- Web-based scientific data sharing portal, as outlined above. Installed on a virtual machine (e.g. AWS) recommended by team and provided by sponsor.
- User manual written for researchers (i.e. those depositing code) and non-technical users, covering installation, configuration, and updating of the core database. Also covers operation of the app, including installation and configuration/connection to database.
- A strong as-built report detailing the design and implementation of the product in a complete, clear and professional manner. This document should provide a strong basis for future development of the product.
- Complete professionally-documented codebase, delivered both as a repository in GitHub, BitBucket, or some other version control repository; and as a physical archive on a USB drive.