The Last Plague

Investigating the Lethality of Pandemics Using Deterministic and Stochastic Epidemic Models

Co-Authors: Li Qing Wang & Uma Wu
Faculty Advisor: James D. Berger
SCI001 T2 Project
March 9, 2016
Abstract

This project investigates the effects of seven parameters on the spread of viral epidemics: infectivity, infectious period, mobility, incubation period, mortality, immunity and recovery period. Two algorithms were created to simulate the progression – one stochastic, and one deterministic. While the deterministic model simulates the spread through regulating the flux between compartments, the stochastic model simulates the spread by applying the parameters as a set of probabilities. The models were calibrated with current data on SARS and Ebola from WHO. Using optimization methods, we found the characteristics of viral diseases that would lead to the most lethal disease, marked by death rates of 92.26% of the world population in the deterministic model and 98.76% in the stochastic model. By investigating the effect of each parameter on the disease, we found that the key factors in regulating the spread of an epidemic are percentage immunity, mobility and recovery rate.

Introduction

Scientific question: what parameter values define the most deadly disease possible, and how do they each contribute to the epidemic?

An epidemic is defined as a sudden spread of a disease that affects an unexpectedly high number of people in a certain population [1]. The determinant factors for the lethality examined in this study include infectivity, infectious period, mobility, incubation period, mortality, immunity and recovery period (refer to Appendix Pages for definitions).

Although we are working with simulated diseases, the combination of parameters that define a “super viral disease” is important because any of these combinations may be present in a future infection. Therefore, by knowing the most dangerous aspects of an infectious disease, we can better focus our resources to counteract it.

This project has four phases: 0, I, II and III. The model is developed in Phase 0, calibrated in Phase I and used to answer our scientific question in Phases II and III. The models use the susceptible-infected-removed model with a few more compartments to provide insight into the effect of each on the spread of diseases.

The deterministic model is a closed model. The flux in and out of the compartments is represented as a series of differential equations. In order to optimize this program for certain desired outputs, the L-BFGS-B method was implemented [14]. The stochastic model is also closed, but it relies on a matrix system instead. To optimize this program for desired outputs, the Bayesian Optimization package Spearmint was implemented.

Different sets of parameters can contribute to the same death rate; therefore, further analysis was done to find the trend in those combinations.
Procedure

Phase 0

Principle Objective: To create the deterministic and stochastic models

Deterministic

Assumption 1: The parameters remain constant throughout the simulation.

Justification: This means the effects of borders, quarantines, cures, and other currently implementable disease-control tactics are unavailable. The implementation of these features would impede the observations on the spread of the disease amidst all of the other elements affecting it.

Stochastic

- See also Assumptions 1 for the Deterministic Model-

Assumption 1: The disease can only spread between adjacent individuals.

Justification: While it is possible to simulate the disease spreading as a function of the total number of infected/incubating individuals regardless of their position in the simulated matrix, it defeats the point of having a stochastic process.

Assumption 2: The parameter values produced by optimizing the program are scalable and will produce around the same proportions between individuals of different states regardless of population size.

Justification: The spread of the disease overall have the same progression as long as the population is >9, since the disease needs to be spreading between unique individuals. This has been verified.
Phase I

**Principle Objective:** Fitting and calibrating the program to real disease data

Cumulative data on the number of cases, recoveries and deaths for SARS and cumulative data on the number cases and deaths for Ebola were collected from the World Health Organization (WHO) website [7, 11].

Target points were chosen for the simulations to fit. The best trend line for each category of data was found by minimizing the coefficient of determination ($R^2$) to a value below a biological alpha of 5% with respect to the population using minimize function L-BFGS-B. For each disease, the category of data with the highest number of data points on the trend line was selected, and target points were chosen from these points. The death data and infected data for SARS and Ebola were used, respectively. The selected data points were representative of the trend of the disease, and minimized the noise in the raw data. [For the raw data used to fit the data points, refer to Appendix Pages 1-7.]

Phase II

**Principal Objective:** Finding the Parameters that result in the most deadly pandemics

In both models, the functions representing the simulation were reconfigured to return the negative maximum percentage of the population that is dead over the time frame of a year.

**Deterministic - L-BFGS-B**

The percentage mortality is maximized using the L-BFGS-B gradient evaluation method. The L-BFGS-B evaluates the derivative of the gradient from a given starting point and follows the slope to find a local minimum. To counteract the possibility of multiple minima within our function, multiple parameter sets were generated (~10000) to ideally produce as many minima
as possible. Among these, the ones with the highest death rates were selected for analysis.

**Stochastic - Spearmint**

The Spearmint package uses Bayesian Optimization in a global black-box operation in order to find the global minimum of a function. This experiment required no more than using Spearmint to find the global minimum of the Stochastic outputs while minimizing the noise that comes from the random factors that govern its processes.

**Assumption:** everyone was susceptible.

**Justification:** if a significant fraction of a population is pre-immune, the disease may not become an epidemic at all.

**Phase III**

**Principal Objective:** investigating the effect of each parameter on percentage death independently.

While keeping other parameters at the values that lead to the most deadly disease, each parameter in the deterministic program was varied independently from 0.1~1.0, in increments of 10%. The output percentage death, defined by the percentage of the world population dying from the disease, was recorded.

**Results**

Each of the following sets of parameters fit the respective disease data collected from WHO with the sum of residuals squared indicated.

**PHASE I RESULTS**

*Deterministic Fitting Results (PHASE I) (FIGURE 3)*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>EBOLA</th>
<th>SARS</th>
<th>SARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infectivity</td>
<td>62.9133%</td>
<td>12.2454%</td>
<td>63.4744%</td>
</tr>
<tr>
<td>Infection Rate</td>
<td>30.6023%</td>
<td>98.3275%</td>
<td>37.0586%</td>
</tr>
<tr>
<td>Mobility</td>
<td>99%</td>
<td>98.211%</td>
<td>76.4858%</td>
</tr>
<tr>
<td>Incubation Period</td>
<td>1.395</td>
<td>101.01</td>
<td>150.443</td>
</tr>
<tr>
<td>Incubation Rate</td>
<td>71.68%</td>
<td>99%</td>
<td>0.6647%</td>
</tr>
<tr>
<td>Mortality</td>
<td>46.2063%</td>
<td>42.6484%</td>
<td>29.8424%</td>
</tr>
<tr>
<td>Percent Immunity</td>
<td>0.1%</td>
<td>0.1%</td>
<td>0.9516%</td>
</tr>
<tr>
<td>Recovery Rate</td>
<td>10.6243%</td>
<td>43.7013%</td>
<td>32.6739%</td>
</tr>
<tr>
<td>Population</td>
<td>6990000000</td>
<td>1795595082</td>
<td>1407579615</td>
</tr>
<tr>
<td>Residual Sum</td>
<td>2.546% of Pop</td>
<td>0.0083558% of Pop</td>
<td>0.0081459% of Pop</td>
</tr>
</tbody>
</table>

*Stochastic Fitting Results (PHASE I) (FIGURE 4)*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>EBOLA</th>
<th>SARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infectivity</td>
<td>99%</td>
<td>43.14%</td>
</tr>
<tr>
<td>Infectious Period</td>
<td>25</td>
<td>11</td>
</tr>
<tr>
<td>Mobility</td>
<td>27.6382%</td>
<td>40.22%</td>
</tr>
<tr>
<td>Incubation Period</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>Parameter Set 1</td>
<td>Parameter Set 2</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Mortality</td>
<td>36.72%</td>
<td>13.79%</td>
</tr>
<tr>
<td>Percentage Immunity</td>
<td>50.266%</td>
<td>85.85%</td>
</tr>
<tr>
<td>Recovery Period</td>
<td>28</td>
<td>11</td>
</tr>
<tr>
<td>Population</td>
<td>Any</td>
<td>Any</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.022747</td>
<td>0.022434</td>
</tr>
</tbody>
</table>

**FIGURE 5.1:** Plots of epidemic progression as generated by Deterministic Model & SARS Data

**FIGURE 5.2:** Plots of epidemic progression as generated by Deterministic Model & SARS Data
FIGURE 6: Plot of epidemic progression as generated by Deterministic Model & Ebola Data

FIGURE 7: Plot of epidemic progression as generated by Stochastic Model & SARS Data
<table>
<thead>
<tr>
<th>Model</th>
<th>Disease</th>
<th>Estimated Basic Reproductive Number</th>
<th>$R_0$ from Literature [3, 15]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deterministic</td>
<td>SARS</td>
<td>2.25</td>
<td>0.24-2.47, 2.87</td>
</tr>
<tr>
<td>Deterministic</td>
<td>SARS</td>
<td>1.13</td>
<td>2.4-3.6 ($R_e$)</td>
</tr>
<tr>
<td>Deterministic</td>
<td>Ebola</td>
<td>2.88</td>
<td>1.50-2.67</td>
</tr>
<tr>
<td>Stochastic</td>
<td>Ebola</td>
<td>1.12</td>
<td></td>
</tr>
<tr>
<td>Stochastic</td>
<td>SARS</td>
<td>1.0</td>
<td>0.24-2.47, 3.87</td>
</tr>
</tbody>
</table>

**FIGURE 9: Estimation of Basic Reproduction Number**

**FIGURE 8: Plot of epidemic progression as generated by Stochastic Model & EBOLA Data**
### PHASE II RESULTS

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Deterministic</th>
<th>Stochastic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infectivity</td>
<td>19.82%</td>
<td>99%</td>
</tr>
<tr>
<td>Infectious Period</td>
<td>1.124</td>
<td>14</td>
</tr>
<tr>
<td>Infection Rate</td>
<td>90.53%</td>
<td>7.14%</td>
</tr>
<tr>
<td>Mobility</td>
<td>99.00%</td>
<td>65.61%</td>
</tr>
<tr>
<td>Incubation Period</td>
<td>7.731</td>
<td>4</td>
</tr>
<tr>
<td>Incubation Rate</td>
<td>18.65%</td>
<td>25%</td>
</tr>
<tr>
<td>Mortality</td>
<td>99.00%</td>
<td>90.45%</td>
</tr>
<tr>
<td>Percentage Immunity</td>
<td>1.0%</td>
<td>0.00%</td>
</tr>
<tr>
<td>Recovery Rate</td>
<td>1.0%</td>
<td>4%</td>
</tr>
<tr>
<td>Recovery Period</td>
<td>100</td>
<td>25</td>
</tr>
<tr>
<td>Percentage Dead</td>
<td>92.26%</td>
<td>98.76%</td>
</tr>
</tbody>
</table>

**FIGURE 10:** Parameters that constitute the most lethal disease in each model

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Average</th>
<th>Standard Deviation</th>
<th>Maximum</th>
<th>Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infectivity</td>
<td>19.82362286%</td>
<td>9.03404125%</td>
<td>37.5849%</td>
<td>1.5306%</td>
</tr>
<tr>
<td>Infectious Period</td>
<td>1.123803493%</td>
<td>0.1633870674</td>
<td>1.6784099416</td>
<td>1.010101</td>
</tr>
<tr>
<td>Infectious Rate</td>
<td>90.53117143%</td>
<td>11.04190317%</td>
<td>99%</td>
<td>59.5802%</td>
</tr>
<tr>
<td>Mobility</td>
<td>99%</td>
<td>0%</td>
<td>99%</td>
<td>0.99</td>
</tr>
<tr>
<td>Incubation Period</td>
<td>7.7306150843%</td>
<td>7.0662311461</td>
<td>38.6085479325</td>
<td>2.6623430216</td>
</tr>
<tr>
<td>Incubation Rate</td>
<td>18.64900571%</td>
<td>8.47061018%</td>
<td>37.5609%</td>
<td>2.5901%</td>
</tr>
<tr>
<td>Mortality</td>
<td>99%</td>
<td>0%</td>
<td>99%</td>
<td>99%</td>
</tr>
<tr>
<td>Percentage Immunity</td>
<td>1%</td>
<td>0%</td>
<td>1%</td>
<td>1%</td>
</tr>
<tr>
<td>Recovery Rate</td>
<td>1%</td>
<td>0%</td>
<td>1%</td>
<td>1%</td>
</tr>
<tr>
<td>Percentage Dead</td>
<td>92.25818%</td>
<td>8.622e-4%</td>
<td>92.2588%</td>
<td>92.2555%</td>
</tr>
</tbody>
</table>

**FIGURE 11:** PHASE II Deterministic Parameter Features
Deterministic Model
The data shows parameter values that contribute to the most deadly disease predicted by the deterministic model. (Figure 10)
The disease is infectious slightly over 1 day. A diseased individual may infect 99% of the healthy people he meets. Only 19.82% of infecteds will display symptoms after infection. Other individuals do not show any symptom until 7.7 days after infection. 99% of infected individuals will die from the disease, and only 1% will gain immunity. Infected individuals, if capable of recovering from the disease, take 100 days to do so.
Stochastic Model
The data shows parameter values that contribute to the most deadly disease predicted by the stochastic model. (Figure 10)
The disease incubates for 4 days. There is a 99% chance that an infected population will display symptoms of the disease, and the disease is infectious over a span of two weeks. A healthy individual has a 65.61% chance of contracting the disease from an infectious individual in close proximity. Infected individuals have a 90.45% chance of dying from the disease. No one can develop immunity against this disease, but individuals can recover from the disease over a period of 25 days.

FIGURE 13: Trend of optimized pandemic (Stochastic)
FIGURE 14.1: Incubation Period compared to Infectivity

FIGURE 14.2: Percentage Death compared to Incubation Period

FIGURE 14.3: Incubation Period compared to Infection Rate
**PHASE III RESULTS**

<table>
<thead>
<tr>
<th>Infectivity</th>
<th>Infectious Period</th>
<th>Mobility</th>
<th>Incubation Period</th>
<th>Mortality</th>
<th>Percentage Immunity</th>
<th>Recovery Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>≤30%</td>
<td>1-2 days</td>
<td>≥20%</td>
<td>5-10 days</td>
<td>any</td>
<td>≤70%</td>
<td>≤70%</td>
</tr>
</tbody>
</table>

FIGURE 15: Table of criteria for a significant drop in population as a result of an epidemic

**Analysis**

**Analysis of PHASE I Data**

*Estimation of Basic Reproduction Number*

The basic reproduction number characterizes the ability of a disease to spread, and is calculated by dividing the infection rate by recovery rate. On the other hand, the effect reproduction number \( R_e \) represents the observed number of secondary infections from a primary infection. Since the two are sometimes used interchangeably in literature, both values were obtained from literature as references.

Two sets of parameters were selected for SARS because both provided \( R_0 \) that were within the literature range. No statistical comparison was feasible between our results and the literature values due to the limitations of the model. However, it is clear that the \( R_0 \) estimated by both models (Figure 9) generally adhered to what was reported from the literature. The deterministic model generally predicts a higher \( R_0 \) than the stochastic model.

*Incubation Periods*

The stochastic model predicted a more reasonable incubation period for each disease compared to the deterministic one. The actual incubation periods for SARS and Ebola are up to two weeks [16] and up to 21 days [17], respectively. Both of these adhere significantly more to the stochastic results.

*Deterministic vs. Stochastic*

Comparing the shapes of the graphs of simulated SARS and Ebola produced by both models to the graphs of raw disease data, the stochastic model is clearly a better model than the deterministic one. The stochastic model is a better fit for the randomness in the nature of disease spread, unlike the deterministic.

Combining this with analyses of estimated \( R_0 \) and incubation period, the stochastic model seemed to have produced a lower residual for each disease (See Appendix Graphs).
Analysis of PHASE II Data

A Holistic Analysis of PHASE II Deterministic Data

This is done to deterministic only due to the various sets of optimized parameters it produced, all of which produced the desired maximal death rate. Of the 10000 parameter sets, 34 representative ones were selected for analysis.

The standard deviation for each parameter was computed using 34 sets of parameters that yield a percentage death of 92.26%. Since the standard deviation of mobility, mortality, percentage immunity and recovery are zero, these parameters must be at a specific value to result in the most lethal epidemic. Therefore, any changes made to those parameters alone will result in a death rate deviating from 92.26%.

The disease has high mobility (99%), high mortality (99%), low immunity (1%) and low recovery rate (1%). Infectivity can range from 1.53% to 37.58%, infection rate can range from 59.58% to 99% and incubation period can range from 2.66 ~ 38.61 days. A clear correlation between incubation period and infectivity can be observed below (Figure 14.1). When infectivity is below 20%, incubation period is less than a week. As infectivity goes beyond 20%, incubation period increases drastically and can be as high as 38 days.

Therefore, as shown in Figures 14.1-3, a combination of the following qualities is most likely to happen and will lead to a 92.26% death rate: 1.53%~26.21% infectivity, a short infectious period of 1~1.33 days, 99% mobility, within two weeks of incubation, 99% mortality, 1% immunity and 1% recovery.
Discussion

Accuracy & Precision

The accuracy of either models can be defined by how replicable the results are. The deterministic model is completely accurate since the same parameters will always yield the same output. Therefore, the accuracy of deterministic model is not representative of its uncertainty, since no model can have 0 uncertainty. However, for the stochastic model, standard deviations of outputs were determined. The standard deviation of the phase II result, (0.005512%), defines the uncertainty of the model since it is purely based on the outputs of the program and does not depend on any data. This small standard deviation indicates an accurate model.

The precision of either models can be defined as how well it fits to the real data in Phase I. This was obtained by fitting the simulation to the real data by minimizing the residuals. The stochastic model’s precision cannot be determined through its precision due to the restraint in runtime as a function of population (it can only fit the trend), and is therefore not representative of its uncertainty. The deterministic model does yield a wide array of residuals, however. In our fitting of the data, the lowest residual was 2.546% of Population with Ebola, while the two SARS fittings yielded 0.0083558% of population and 0.0081459% of population respectively. Since the percentages are less than 5%, the model is likely precise.

Limitations

Limitations of Collected Data

Inconsistency in WHO data
The cumulative number of SARS cases fluctuated for the last few days for which data were collected. SARS is a diagnosis of exclusion; as previous cases were further investigated, some patients were re-examined and re-diagnosed [6]. Since the fluctuation only involved a few individuals, we disregarded the occasional decrease, and recorded further data by adding the number of newly dead people to the cumulative number.

Limitation of Obtainable Categories of Data
While there are 7 parameters, only the aforementioned 3 had available data.

Limitations of Model

1. Quarantine & Borders
In April 2003, the Chinese government started implementing quarantine and restricting social interactions by methods such as closing theatres and implementing border control [4]. Quarantine would limit the mobility of the disease; however, this is not accounted for in the model. (See Assumptions)

2. Parameters change with time
The parameters in both models are constant over time, while in reality they do change with
time. For example, the mortality of both disease increased with time.

**Run Time Constraints**

**Deterministic**
The deterministic model does not take long to complete one function evaluation (~0.1s for Population = 7000000000) due to its nature as a linear-time computation, but its optimization took much longer due to the methodology of the L-BFGS-B method of evaluating the function multiple times until it reached a local minimum. Therefore, a global optimization would have been better for this instance.

**Stochastic**
The stochastic model takes a lot longer than the deterministic (~1s for Population = 400) to complete one function evaluation. With the Spearmint Optimization package, we were able to bypass that issue. Not only does Spearmint find the global minimum, it only takes around 5 hours to yield applicable and accurate results. However, it is still subject to the limitations of populations, since the run time increases linearly with the addition of an individual.

**Inferences**

**Deterministic Phase II**
Summary: The most deadly disease according to the deterministic model is a disease that doesn’t show symptoms immediately (19.82% infectivity), transmits quickly from person to person (99% mobility), almost certainly and immediately kills its host upon infection (99% mortality), and is almost impossible to develop immunity from (only 1% of the population can become immune). This disease can kill 92.26% of the world population in one year.

This set of parameters produces a greater incubating population than infectious population. There are many currently known viruses that can remain dormant in individuals (e.g. Rabies, STDs) for a relatively long time before manifesting symptoms. The disease may cover its lethality by displaying nonlethal typical flu symptoms as fever, cough and diarrhea when individuals are first infected, thus creating a high incubating population. This feature of the simulated disease greatly contributes to the risk of unknowingly getting infected since the number of susceptible people becoming incubating is dependent on both the number of carriers (infecteds) and susceptibles.

**Stochastic Phase II**
Summary: The most deadly disease according to the stochastic model is one that shows symptoms almost immediately (99% infectivity), transmits at a moderate rate (65.61%) from person to person, is infectious over two weeks, and is develop immunity. This disease can kill 98.76% of the world population in one year.

Instead of a high mobility, the disease becomes highly infective with a small incubation period, which means the patients would immediately show symptoms upon infection. A zero percent immunity means that even if a patient did manage to survive, they would become susceptible again with another chance of being infected and eventually succumbing to the
This disease resembles past epidemics such as AIDS, since it mutates very quickly and suppresses the immune system. The disease would exhibit symptoms that lead to more infection and eventually death, such as skin lesions (e.g. smallpox), discharging fluids (e.g. diarrhea), and even irregular behavior (e.g. rabies patients may bite other individuals).
Deterministic Phase III results and inferences
Since the purpose of Phase III is purely making inferences, its results will be shown here along with the inferences.

Figure 16.1 The percentage dead seems to decrease somewhat linearly with increased recovery rate when the recovery rate is lower than 70%. A recovery rate of 70% seems to be a threshold beyond which the disease is no longer an epidemic.

Figure 16.2 A disease with low mortality (10%) can have a high percentage death (~70%). The percentage death increase by ~10% in response to a 10 fold increase in mortality. This may indicate that the mortality of a disease is not the major determinant of its percentage death.
Figure 16.3 A disease for which an infected can only spread the infection to less than 20% of the people he contacts will not cause an epidemic. The percentage death increases drastically with the increase in mobility. This suggests that mobility is a major determinant in the spread of a disease.

Figure 16.4 As incubation rate increases, or as the length of incubation period decreases, the percentage death decreases.
Figure 16.5 The larger the fraction of the infected population that shows symptoms, the less deadly the disease becomes. This may suggest that a disease must not let the patients display symptoms immediately in order to let it spread unknowingly, and eventually kill a large percentage of the world population.

Figure 16.6 The disease is the most deadly when zero percent of the population is immune to it. The death rate decreases with increased immunity. There seems to be a threshold at 70% immunity, beyond which the disease is essentially no longer an epidemic.
**Key Inferences and Observations**

In agreement with what was observed in phase II, any deviation from the set of the most deadly parameters led to a decrease in percentage death. However, this phase allowed a closer investigation of the magnitude of such effect. For example, the mortality barely alters the percentage death while independent changes in mobility, recovery rate, and percentage immunity led to much greater changes. The effect of infection rate, incubation rate, and infectivity is somewhere in between. Some threshold values were observed in percentage immunity ($\leq 70\%$), mobility ($\geq 20\%$) and recovery rate ($\leq 70\%$). These three parameters are the major contributors to the percentage death of an epidemic; however, they must reach a certain threshold level for a disease to become an epidemic. These three parameters need to be regulated in order to prevent or control the spread of epidemic most efficiently.

As a summary of phase III, a chart of criteria for the most lethal epidemic predicted by the deterministic model was produced (figure 15). If an outbreak is suspected and several of its parameters fall into these values, it is likely to develop into a fairly lethal epidemic. Such disease can then be most effectively regulated by controlling percentage immunity, mobility and recovery rate.

![Percentage Death vs. Infection Rate](image)

Figure 16.7 A low infection rate such as 10% can lead to a death percentage of ~87%. The increase in infection rate from 10% to 70% causes the most increase in percentage death. Beyond that, the increase in infection rate has minute effect on percentage death.
**Conclusion**

Both the deterministic and stochastic models developed are able to fit realistic disease data and predict combinations of parameters that would lead to the most deadly epidemic. Both models also predict a high death rate when the chances of developing immunity are low, indicating the threat of an immune-suppressive disease is especially high.

Ranges of values for each parameter that constitute the most lethal epidemic outlined can be used to assess the lethality of a real disease. With further development, future potentials of this pair of models include investigation of other viral diseases, predicting the outcome of a disease in real time, and modelling the effect of prevention and control methods on the epidemic.

**References**


**Special Thanks**
James D. Berger for mentoring this project
Michael Gelbart for implementing Spearmint, suggesting fitting methods, and debugging
Pam Kalas for reviewing and advising this project
Omer Angel for providing insight on the data analysis
Costanza Piccolo for refining our parameters
Eric Cytrynbaum for refining the mechanics of the model
# Appendix

## Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SARS Raw Data: Cumulative Number of Deaths vs. Time (Fig1)</td>
<td>1</td>
</tr>
<tr>
<td>SARS Raw Data: Cumulative Number of Recovered Individuals vs. Time (Fig2)</td>
<td>1</td>
</tr>
<tr>
<td>SARS Extrapolated Data: Number of Alive &amp; Infected Individuals vs. Time (Fig3)</td>
<td>2</td>
</tr>
<tr>
<td>SARS Extrapolated Data: Cumulative Number of Deaths vs. Time Fitted (Fig4)</td>
<td>2</td>
</tr>
<tr>
<td>SARS Extrapolated Data: Currently Infectious Individuals vs. Time Fitted (Fig5)</td>
<td>3</td>
</tr>
<tr>
<td>EBOLA Raw Data: Cumulative Number of Infectious Individuals vs. Time (Fig6)</td>
<td>3</td>
</tr>
<tr>
<td>EBOLA Raw Data: Cumulative Number of Deaths vs. Time (Fig7)</td>
<td>4</td>
</tr>
<tr>
<td>EBOLA Raw Data: Cumulative Number of Deaths vs. Time Fitted (Fig7)</td>
<td>4</td>
</tr>
<tr>
<td>EBOLA Extrapolated Data: Cumulative Number of Infectious vs. Time Fitted (Fig8)</td>
<td>5</td>
</tr>
<tr>
<td>EBOLA Parameter Sets</td>
<td>6</td>
</tr>
<tr>
<td>SARS Parameter Sets</td>
<td>7</td>
</tr>
<tr>
<td>PHASE II Parameter Sets (Deterministic)</td>
<td>8</td>
</tr>
<tr>
<td>DeterministicEPIMOD PHASE I Model</td>
<td>10</td>
</tr>
<tr>
<td>DeterministicEPIMOD PHASE II Model</td>
<td>17</td>
</tr>
<tr>
<td>StochasticEPIMOD PHASE I Model</td>
<td>21</td>
</tr>
<tr>
<td>StochasticEPIMOD PHASE II Model</td>
<td>33</td>
</tr>
<tr>
<td>PHASE III Data Table</td>
<td>39</td>
</tr>
<tr>
<td>Definitions of Parameters</td>
<td>40</td>
</tr>
<tr>
<td>Phase I Target Points</td>
<td>41</td>
</tr>
</tbody>
</table>
Appendix Fig 1

Cumulative Number of Deaths vs. Time
SARS Raw Data

Appendix Fig 2

Cumulative Number of Recovered Individuals vs. Time
SARS Raw Data
Appendix Fig 3

Number of Alive and Infected Individuals vs. Time

Appendix Fig 4

Cumulative Number of Deaths vs. Time

$D(t)
\quad n=5$

$R^2 = 0.9990672921$
Currently Infectious Individuals vs. Time

![Graph of Currently Infectious Individuals vs. Time](image1)

$R^2 = 0.9983664682$

Appendix Fig5

Number of Infectious Individuals vs. Time

![Graph of Number of Infectious Individuals vs. Time](image2)

Ebola Raw Data

Appendix Fig6
Cumulative Infectious vs. Time

Ebola (n=22)

$R^2 = 0.9998275675$

Appendix Fig8
<table>
<thead>
<tr>
<th></th>
<th>Infectivity</th>
<th>Infection Rate</th>
<th>Mobility</th>
<th>Incubation Rate</th>
<th>Mortality</th>
<th>Percentage Immunity</th>
<th>Recovery Rate</th>
<th>Population</th>
<th>ResidualSum</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average</strong></td>
<td>0.99</td>
<td>0.001</td>
<td>0.988710041</td>
<td>0.989113625</td>
<td>0.8238342917</td>
<td>0.0010002083</td>
<td>0.0010002083</td>
<td>3470330353.2916</td>
<td>0.000287125</td>
</tr>
<tr>
<td><strong>Standard Deviation</strong></td>
<td>0</td>
<td>0</td>
<td>0.0044474239</td>
<td>0.0030006973</td>
<td>0.0039407441</td>
<td>1.02062072615966E-08</td>
<td>1.02062072615966E-08</td>
<td>2126288880.6962</td>
<td>5.50345741132434E-06</td>
</tr>
<tr>
<td><strong>Maximum Population</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>6866802093</td>
<td></td>
</tr>
<tr>
<td><strong>Minimum Population</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>729407782</td>
<td></td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>2218138142</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>2584033055</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>1901057942</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>3827085168</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>5857301234</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.982891</td>
<td>0.82749</td>
<td>0.001</td>
<td>0.001</td>
<td>6114060146</td>
<td>0.00029</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>930036007</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>729407782</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>801051285</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>6193655290</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>2750261610</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.977391</td>
<td>0.81404</td>
<td>0.001</td>
<td>0.001</td>
<td>3354761252</td>
<td>0.000299</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>6205577078</td>
<td>0.000285</td>
</tr>
<tr>
<td><strong>CHosen</strong></td>
<td>0.99</td>
<td>0.001</td>
<td>0.971673</td>
<td>0.989046</td>
<td>0.809716</td>
<td>0.0010005</td>
<td>0.0010005</td>
<td>1040879429</td>
<td>0.000307</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>1737874180</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.97679</td>
<td>0.830056</td>
<td>0.001</td>
<td>0.001</td>
<td>6085799484</td>
<td>0.000295</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.989994</td>
<td>0.824532</td>
<td>0.001</td>
<td>0.001</td>
<td>6866802093</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>4768289329</td>
<td>0.000285</td>
</tr>
<tr>
<td>0.99</td>
<td>0.001</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
<td>0.824537</td>
<td>0.001</td>
<td>0.001</td>
<td>2566033336</td>
<td>0.000285</td>
</tr>
<tr>
<td>SARS Parameter Sets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Infectivity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Infection Rate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mobility</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Incubation Period</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Incubation Rate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Mortality</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Percentage Immunity</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recovery Period</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Recovery Rate</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Population</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Residual Sum</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Estimated Basic Reproduction Number</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Standard Deviation

| 0.3224714094 | 0.2345012131 | 0.097017819 | 0.2934826039 | 0.0495100597 | 0.0636622241 | 0.0496357966 | 2350778466.15791 | 1.7285763331 |
| 0.486726 | 0.253035 | 0.866356 | 0.894551 | 0.215118 | 0.029584 | 16.0815657012 | 0.062183 | 4930935930 | 0.456655 |
| 0.851473 | 0.187879 | 0.868682 | 10.016125628 | 0.099839 | 0.233785 | 5.8622832421 | 0.170582 | 1959432004 | 0.492016 |
| 0.644418 | 0.26588 | 0.944756 | 1.489913287 | 0.67118 | 0.246621 | 7.9605158414 | 0.12562 | 114133266 | 0.370468 |
| 0.200633 | 0.124275 | 0.796056 | 1.474758763 | 0.678077 | 0.249025 | 21.1671570391 | 0.047243 | 781370871 | 0.276411 |
| 0.942965 | 0.970427 | 0.973074 | 1.0972913363 | 0.911335 | 0.265489 | 6.096631611 | 0.164025 | 4384338355 | 0.990407 |

### Chosen Parameter set#1

| 0.942952 | 0.08657 | 0.255206 | 0.706624 | 0.1020136136 | 0.988129 | 0.283516 | 20.461195343 | 0.048873 | 5630901875 | 0.317624 |
| 0.08657 | 0.255206 | 0.706624 | 0.1020136136 | 0.988129 | 0.283516 | 20.461195343 | 0.048873 | 5630901875 | 0.317624 |
| 0.035323 | 0.142002 | 0.853813 | 1.779438571 | 0.561975 | 0.322865 | 7.2050781391 | 0.138791 | 1453683352 | 0.739969 |
| 0.245728 | 0.430487 | 0.826955 | 1.3681443994 | 0.730917 | 0.325093 | 7.381544662 | 0.135473 | 3293913925 | 0.625295 |
| 0.533429 | 0.366478 | 0.873111 | 1.2384805825 | 0.807441 | 0.338197 | 8.5397825771 | 0.117099 | 1607034240 | 0.233318 |
| 0.533066 | 0.387292 | 0.907111 | 1.589327349 | 0.629197 | 0.339495 | 15.5265037419 | 0.064406 | 5784967215 | 0.348509 |

### Chosen Parameter set#2

| 0.840515 | 0.539667 | 0.801637 | 13.8900463928 | 0.071994 | 0.347849 | 5.1692943913 | 0.19345 | 6631318516 | 0.819369 |
| 0.595789 | 0.538563 | 0.62483 | 1.2243318821 | 0.816772 | 0.349846 | 9.1958250954 | 0.108745 | 73865743 | 0.724964 |

**Note:** The table represents parameter sets for SARS with various values for different parameters such as infectivity, infection rate, mobility, incubation period, incubation rate, mortality, percentage immunity, recovery period, recovery rate, population, and residual sum. The estimated basic reproduction number (Basic Reproduction Number) is also provided for each set.
<table>
<thead>
<tr>
<th>Parameter Sets</th>
<th>Infectivity</th>
<th>Infectious Period</th>
<th>Infection Rate</th>
<th>Mobility</th>
<th>Incubation Period</th>
<th>Incubation Rate</th>
<th>Mortality</th>
<th>Percentage Immunity</th>
<th>Recovery Rate</th>
<th>Percentage Dead</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max</td>
<td>0.375849</td>
<td>1.6784099416</td>
<td>0.99</td>
<td>0.99</td>
<td>38.6085479325</td>
<td>0.375609</td>
<td>0.99</td>
<td>0.01</td>
<td>0.01</td>
<td>0.922588</td>
</tr>
<tr>
<td>Min</td>
<td>0.015306</td>
<td>1.0101010101</td>
<td>0.595802</td>
<td>0.99</td>
<td>2.6623430216</td>
<td>0.025901</td>
<td>0.99</td>
<td>0.01</td>
<td>0.01</td>
<td>0.922555</td>
</tr>
<tr>
<td>Average</td>
<td>0.1982362286</td>
<td>1.123803493</td>
<td>0.9053117143</td>
<td>0.99</td>
<td>7.7306120843</td>
<td>0.1864900571</td>
<td>0.99</td>
<td>0.01</td>
<td>0.01</td>
<td>0.9225818</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.0903404125</td>
<td>0.163870674</td>
<td>0.1104190317</td>
<td>0</td>
<td>7.0662311461</td>
<td>0.0847061018</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>8.6221329420E-06</td>
</tr>
</tbody>
</table>

8
DETERMINISTIC EPIDEMIC MODEL PHASE I
Uma Wu & LiQing Wang

~Ver.7 Updates~
-began implementation of fitting model

BLOCK DIAGRAM

#####PARAMETERS#####
I = Infectivity (percentage of people that move from INCUBATING to INFECTIOUS without IncP [OVERRIDE])
InfP = Length of Infectious Period (rate of individuals moving from INFECTIOUS to INCUBATING)
Mob = Mobility (rate of individuals moving from SUSCEPTIBLE to INCUBATING)
IncP = Length of Incubation Period (rate of individuals moving from INCUBATING to INFECTIOUS)
Mor = Morbidity (percentage of infectious individuals that move from INFECTIOUS to DEAD)
Imn = Immunity (percentage of recovered individuals that move from INFECTIOUS to IMMUNE)
RecP = Length of Recovery Period (rate of individuals moving from INFECTIOUS and INCUBATING to SUSCEPTIBLE)

#Note: Incubating refers to both the disease being dormant and the disease displaying nonconsequential symptoms.
Pop = Population (Total number of people in each compartment. Remains constant throughout simulation)

#####INITIAL VALUES#####
Pop = Population
T = Time of Simulation in Days
IMMUNE = Pop*PImmune  
SUSCEPTIBLE = Pop - Pop*PImmune  
INCUBATING = 1  
INFECTIOUS = 0  
DEAD = 0  

#####DIFFERENTIAL EQUATIONS FROM BLOCK#####  
IMMUNE = IMMUNE + INFECTIOUS*Imn + INCUBATING*Imn  
SUSCEPTIBLE = SUSCEPTIBLE + INFECTIOUS*RecP + INCUBATING*RecP -  
SUSCEPTIBLE*(INCUBATING/Population)Mob  
INCUBATING = INCUBATING + SUSCEPTIBLE*Mob + INFECTIOUS*InfP - INCUBATING*Imn -  
INCUBATING*I - INCUBATING*IncP - INCUBATING*RecP  
INFECTIOUS = INFECTIOUS + INCUBATING*I + INCUBATING*IncP - INFECTIOUS*Imn -  
INFECTIOUS*InfP - INFECTIOUS*Mor  
DEAD = DEAD + INFECTIOUS*Mor  

import numpy as np  
import random  
from scipy.optimize import minimize  
import matplotlib.pyplot as plt  

################FITTING##################  

SusFit = False  
IncFit = False  
InfFit = False  
ImmFit = False  
DeaFit = False  

if SusFit:  
    SUSCEPTIBLEquery = []  
    with open("SUSCEPTIBLEfit.csv", "r") as SUSCEPTIBLEfit:  
        for line in SUSCEPTIBLEfit:  
            SUSCEPTIBLEquery.append(tuple([int(x) for x in line[:-1].split(",")]))  
    preset = SUSCEPTIBLEquery  
    status = "SusFit"  

if IncFit:  
    INCUBATINGquery = []  
    with open("INCUBATINGfit.csv", "r") as INCUBATINGfit:  
        for line in INCUBATINGfit:  
            INCUBATINGquery.append(tuple([int(x) for x in line[:-1].split(",")]))  
    preset = INCUBATINGquery
status = "IncFit"

if InfFit:
    INFECTIOUSquery = []
    with open("INFECTIOUSfit.csv", "r") as INFECTIOUSfit:
        for line in INFECTIOUSfit:
            INFECTIOUSquery.append(tuple([int(x) for x in line[:-1].split(",")]))
    preset = INFECTIOUSquery
    status = "InfFit"

if ImmFit:
    IMMUNEquery = []
    with open("IMMUNEfit.csv", "r") as IMMUNEfit:
        for line in IMMUNEfit:
            IMMUNEquery.append(tuple([int(x) for x in line[:-1].split(",")]))
    preset = IMMUNEquery
    status = "ImmFit"

if DeaFit:
    DEADquery = []
    with open("DEADfit.csv", "r") as DEADfit:
        for line in DEADfit:
            DEADquery.append(tuple([int(x) for x in line[:-1].split(",")]))
    preset = DEADquery
    status = "DeaFit"

""
INPUTS:
time: the current time value
y: the current value of the function
query: the list of points (t, value) to fit against
OUTPUTS:
dy: the deviation from the query
""

fit = False
if fit:
    def Fit(time, y, query = preset):
        for i in query:
            if time == i[0]:
                dy = y - i[1] #Calculates the residuals
                return dy
            else:
                return "NONE"
status = "SusFit"

#SIMULATION

def DeterministicEPIMOD1(Parameters, T = 365, Plot = True, Print = True, fit = fit, status = status, dt = 0.1):

    IMMUNE = np.zeros(int((T+1)/dt))
    SUSCEPTIBLE = np.zeros(int((T+1)/dt))
    INCUBATING = np.zeros(int((T+1)/dt))
    INFECTIOUS = np.zeros(int((T+1)/dt))
    DEAD = np.zeros(int((T+1)/dt))

    #PARAMETER VALUES (INITIAL VALUES)
    I = Parameters[0]
    InfP = Parameters[1]
    Mob = Parameters[2]
    IncP = Parameters[3]
    Mor = Parameters[4]
    Imn = Parameters[5]
    RecP = Parameters[6]

    Pop = Parameters[7]

    #INITIAL VALUES
    IMMUNE[0] = 0 #1213 #Pop*PImmune
    INCUBATING[0] = 1 #1516
    INFECTIOUS[0] = 0 #286
    DEAD[0] = 0
    SUSCEPTIBLE[0] = Pop - IMMUNE[0] - INCUBATING[0] - INFECTIOUS[0] - DEAD[0]

    #FITTING
    residuals = []
    n = 0

    for t in range(1, int(float(T)/dt+1)):
        SUSCEPTIBLEin = INFECTIOUS[t-1]*RecP*dt + INCUBATING[t-1]*RecP*dt
        SUSCEPTIBLEout = SUSCEPTIBLE[t-1]*(INCUBATING[t-1]/float(Pop))*Mob*dt
        SUSCEPTIBLE[t] = SUSCEPTIBLE[t-1] + SUSCEPTIBLEin - SUSCEPTIBLEout

        INCUBATINGin = SUSCEPTIBLE[t-1]*(INCUBATING[t-1]/float(Pop))*Mob*dt +
        INFECTIOUS[t-1]*InfP*dt
        INCUBATINGout = INCUBATING[t-1]*Imn*dt + INCUBATING[t-1]*I*dt +
        INCUBATING[t-1]*IncP*dt + INCUBATING[t-1]*RecP*dt
        INCUBATING[t] = INCUBATING[t-1] + INCUBATINGin - INCUBATINGout
INFECTIONIn = INCUBATING[t-1]*I*dt + INCUBATING[t-1]*IncP*dt
INFECTIONOut = INCUBATING[t-1]*lnn*dt + INFECTIONIn[t-1]*InfP*dt + INFECTIONIn[t-1]*Mor*dt + INFECTIONIn[t-1]*RecP*dt
INFECTION[t] = INFECTIONIn[t-1] + INFECTIONIn - INFECTIONOut

DEADIn = INFECTIONIn[t-1]*Mor*dt
DEADOut = 0
DEAD[t] = DEAD[t-1] + DEADIn - DEADOut

IMMUNEIn = INFECTIONIn[t-1]*Imn*dt + INCUBATING[t-1]*Imn*dt
IMMUNEOut = 0
IMMUNE[t] = IMMUNE[t-1] + IMMUNEIn - IMMUNEOut

if fit == True:
    if status == "SusFit":
        y = SUSCEPTIBLE[t]
    elif status == "IncFit":
        y = INCUBATING[t]
    elif status == "InfFit":
        y = sum(INFECTION)
    elif status == "ImmFit":
        y = IMMUNE[t]
    elif status == "DeaFit":
        y = DEAD[t]
    else:
        print "ERROR: I DON'T KNOW WHAT IT IS BUT SOMETHING IS NOT RIGHT."
        if diffs != "NONE":
            residuals.append(diffs**2)
            # plt.plot(t, preset[n][1], "*r")
            # plt.plot(t, y, ".b")
            # print preset[n][1]
            # n += 1

if Print == True:
    Survivors = IMMUNE[t] + SUSCEPTIBLE[t] + INCUBATING[t] + INFECTION[t]
    print "TOTAL POPULATION = %i" %Pop
    print "REMAINING POPULATION = %f" %Survivors
    print "IMMUNE = %.0f" %round(IMMUNE[t])
    print "SUSCEPTIBLE = %.0f" %round(SUSCEPTIBLE[t])
    print "INCUBATING = %.0f" %round(INCUBATING[t])
print "INFECTIOUS = %.0f" %round(INFECTIOUS[t])
print "DEAD = %.0f" %round(DEAD[t])

if Plot == True:
    plt.plot(IMMUNE, label="IMMUNE")
    plt.plot(SUSCEPTIBLE, label="SUSCEPTIBLE")
    plt.plot(INCUBATING, label="INCUBATING")
    plt.plot(INFECTIOUS, label="INFECTIOUS")
    plt.plot(DEAD, label="DEAD")

    plt.legend(loc="best")
    plt.title("PHASEIIpandemic")
    plt.ylabel("Number of Individuals")
    plt.xlabel("Time/%f (in days)" %dt)

# plt.show()
plt.savefig("PHASEIIpandemicDET.pdf")

if fit == True:
    return abs(sum(residuals))
else:
    return -np.log(DEAD[t])
    # return abs(7452 -(IMMUNE[-1])) + abs(831 - DEAD[-1])

# print minimize(DeterministicEPIMOD, [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 5000], method = 'L-BFGS-B', bounds = ((0.01,(0.01, 1),(0.01, 1),(0.01, 1),(0.01, 1),(0.01, 1),(0.01, 1),(100, 7000000001)))

# print minimize(DeterministicEPIMOD1, [0.1982, 0.9053, 0.99, 0.1865, 0.99, 0.01, 0.01, 7000000000], Parameters)
# else:
#     print "ERROR: I DON'T KNOW WHAT IT IS BUT SOMETHING IS NOT RIGHT."

# data.write("Infectivity, InfectionRate, Mobility, IncubationRate, Mortality, PercentageImmunity, RecoveryRate, Population, ResidualSum")

# i = 0
# trials = 10000
# for n in range(trials):
#     print "\n"
#     print n
#     guess = np.random.rand(8)
#     guess[7] = random.randint(100, 7000000001)
#     Combination = minimize(DeterministicEPIMOD1, guess, method = 'L-BFGS-B', bounds = ((0.001, 0.99),(0.001, 0.99),(0.001, 0.99),(0.001, 0.99),(0.001, 0.99),(0.001, 0.99), (100, 7000000001)))
#     if Combination.fun <= 1:
#         i += 1
#         print "%i candidates found!" %i
#         l = Combination.x[0]
#         InfP = Combination.x[1]
#         Mob = Combination.x[2]
#         IncP = Combination.x[3]
#         Mor = Combination.x[4]
#         Imn = Combination.x[5]
#         RecP = Combination.x[6]
#         Population = Combination.x[7]
#         Residual = Combination.fun
#         data.write("\n" + "%f, %f, %f, %f, %f, %f, %f, %f" %(l, InfP, Mob, IncP, Mor, Imn, RecP, Population, Residual**2))

# else:
#     print "ERROR: I DON'T KNOW WHAT IT IS BUT SOMETHING IS NOT RIGHT."
DETERMINISTIC EPIDEMIC MODEL

Uma Wu & LiQing Wang

Discussed With: Michael Gelbart

~Ver.8 Updates~
-Began implementation of Spearmint
-Removed normalization

-------------------------------------------------------------------------------

BLOCK DIAGRAM

BLOCK DIAGRAM

###PARAMETERS###
Plimmune = Pre-Immunity (percentage of population immune to disease at initial time)
I = Infectivity (percentage of people that move from INCUBATING to INFECTIOUS without IncP [OVERRIDE])
InfP = Length of Infectious Period (rate of individuals moving from INFECTIOUS to INCUBATING)
Mob = Mobility (rate of individuals moving from SUSCEPTIBLE to INCUBATING)
IncP = Length of Incubation Period (rate of individuals moving from INCUBATING to INFECTIOUS)
Mor = Morbidity (percentage of infectious individuals that move from INFECTIOUS to DEAD)
Imn = Immunity (percentage of recovered individuals that move from INFECTIOUS to IMMUNE)
RecP = Length of Recovery Period (rate of individuals moving from INFECTIOUS and INCUBATING to SUSCEPTIBLE)

#Note: Incubating refers to both the disease being dormant and the disease displaying nonconsequential symptoms.

###INITIAL VALUES###
Pop = Population
T = Time of Simulation in Days
IMMUNE = Pop*PImmune
SUSCEPTIBLE = Pop - Pop*PImmune
INCUBATING = 0
INFECTIOUS = 1
DEAD = 0

#####DIFFERENTIAL EQUATIONS FROM BLOCK#####
IMMUNE = IMMUNE + INFECTIOUS*Imn + INCUBATING*Imn
SUSCEPTIBLE = SUSCEPTIBLE + INFECTIOUS*RecP - SUSCEPTIBLE*Mob
INCUBATING = INCUBATING + SUSCEPTIBLE*Mob + INFECTIOUS*InfP - INCUBATING*Imn - INCUBATING*I - INCUBATING*IncP
INFECTIOUS = INFECTIOUS + INCUBATING*I + INCUBATING*IncP - INFECTIOUS*Imn - INFECTIOUS*InfP - INFECTIOUS*Mor
DEAD = DEAD + INFECTIOUS*Mor

import numpy as np
from scipy.optimize import minimize
import scipy.optimize

def main(job_id, params):
    Parameters = [params["I"], params["InfP"], params["Mob"], params["IncP"],
                  params["Mor"], params["Imn"], params["RecP"]]
    return -StochasticEPIMOD(Parameters)

def DeterministicEPIMOD(Parameters, T = 365, Pop = 7000000000):
    dt = 0.1
    IMMUNE = np.zeros(int((T+1)/dt))
    SUSCEPTIBLE = np.zeros(int((T+1)/dt))
    INCUBATING = np.zeros(int((T+1)/dt))
    INFECTIOUS = np.zeros(int((T+1)/dt))
    DEAD = np.zeros(int((T+1)/dt))

    #PARAMETER VALUES (INITIAL VALUES)
    I = Parameters[0] #0.1 #0.011 is the rate of infection from data
    InfP = Parameters[1] #0.12 #max 0.12
    Mob = Parameters[2] #11.0/30.0 #THIS SHOULD DEPEND ON THE NUMBER OF INFECTED INDIVIDUALS AHHH
    IncP = Parameters[3] #7.0/15.0
    Mor = Parameters[4] #0.07 #0.096 is the death rate from data
    Imn = Parameters[5] #0.397 #1.0 - Mor
    RecP = Parameters[6] #0.468 is the recovery rate from data
#INITIAL VALUES
IMMUNE[0] = 0 #1213 #Pop*PImmune
INCUBATING[0] = 1 #1516
INFECTIOUS[0] = 0 #286
DEAD[0] = 0
SUSCEPTIBLE[0] = Pop - IMMUNE[0] - INCUBATING[0] - INFECTIOUS[0] - DEAD[0]

for t in range(1, int(float(T)/dt+1)):
    SUSCEPTIBLEin = INFECTIOUS[t -1]*RecP*dt + INCUBATING[t-1]*RecP*dt
    SUSCEPTIBLEout = SUSCEPTIBLE[t-1]*(INCUBATING[t-1]/float(Pop))*Mob*dt
    SUSCEPTIBLE[t] = SUSCEPTIBLE[t -1] + SUSCEPTIBLEin - SUSCEPTIBLEout

    INCUBATINGin = SUSCEPTIBLE[t-1]*(INCUBATING[t-1]/float(Pop))*Mob*dt +
                    INFECTIOUS[t-1]*InfP*dt
    INCUBATINGout = INCUBATING[t-1]*Imn*dt + INCUBATING[t-1]*I*dt +
                   INCUBATING[t-1]*IncP*dt + INCUBATING[t-1]*RecP*dt
    INCUBATING[t] = INCUBATING[t -1] + INCUBATINGin - INCUBATINGout

    INFECTIOUSin = INCUBATING[t -1]*I*dt + INCUBATING[t-1]*IncP*dt
    INFECTIOUSout = INFECTIOUS[t -1]*Imn*dt + INFECTIOUS[t -1]*InfP*dt +
                   INFECTIOUS[t-1]*Mor*dt + INFECTIOUS[t-1]*RecP*dt
    INFECTIOUS[t] = INFECTIOUS[t -1] + INFECTIOUSin - INFECTIOUSout

    DEADin = INFECTIOUS[t -1]*Mor*dt
    DEADout = 0
    DEAD[t] = DEAD[t -1] + DEADin - DEADout

    IMMUNEin = INFECTIOUS[t-1]*Imn*dt + INCUBATING[t-1]*Imn*dt
    IMMUNEout = 0
    IMMUNE[t] = IMMUNE[t -1] + IMMUNEin - IMMUNEout

    return -np.log(DEAD[t])

def Normalize(Parameters):
    return np.sum(Parameters) - 1

#################################################################
# print minimize(DeterministicEPIMOD, [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1], method = 'L-BFGS-B',
# bounds = ((0.01, 1),)*7)
# print scipy.optimize.fmin_slsqp(DeterministicEPIMOD, [0.3, 0.2, 0.1, 0.1, 0.1, 0.1, 0.1],
# bounds = ((0.01, 1),)*7, f_eqcons = Normalize)
#returns the seven best parameters if not printed

data = open('DeterministicPHASEII2.csv', 'w')
data.write("Infectivity, InfectionRate, Mobility, IncubationRate, Mortality, PercentageImmunity, RecoveryRate, PercentageDead")

trials = 50000
for n in range(trials):
    print n
    guess = np.random.rand(7)
    Combination = minimize(DeterministicEPIMOD, guess, method = 'L-BFGS-B', bounds = ((0.01, 0.99),)*7)
    I = Combination.x[0]
    InfP = Combination.x[1]
    Mob = Combination.x[2]
    IncP = Combination.x[3]
    Mor = Combination.x[4]
    Imn = Combination.x[5]
    RecP = Combination.x[6]
    Dead = np.exp(-DeterministicEPIMOD(Combination.x))/7000000000.0
    data.write("\n" + "%f, %f, %f, %f, %f, %f, %f" %(I, InfP, Mob, IncP, Mor, Imn, RecP, Dead))

# TESTING PURPOSES ONLY
# guess = np.random.rand(7)
# print scipy.optimize.fmin_slsqp(DeterministicEPIMOD, guess, bounds = ((0.01, 1),)*7, f_eqcons = Normalize)
STOCHASTIC EPIDEMIC MODEL

Uma Wu & LiQing Wang

Discussed With: Noah Bayless, Michael Gelbart

 плохо

PATIENT INFORMATION

xLoc = range(0, x_Dim)
yLoc = range(0, y_Dim)
Status = IMMUNE, SUSCEPTIBLE, INFECTIOUS, INCUBATING, or DEAD
tExist = time elapsed since start of simulation
tInc = time incubating
tInf = time infectious
tRec = time recovering

PARAMETERS

PImmune = Pre-Immunity (Probable Percentage of Population that is Pre-Immune to Disease, float)
I = Infectivity (Percentage Probability of Displaying Symptoms When Infected, float)
InfP = Length of Infectious Period (Time Range in which the Disease is Infectious, array)
Mob = Mobility (Percentage Probability of Contracting the Disease from an Adjacent Diseased Individual, float)
IncP = Length of Incubation Period (Time Range in which the Disease does not produce Symptoms In an Infective Individual), array)
Mor = Morbidity (Percentage Probability of Infective Individuals Dying from the Disease, float)
Imn = Immunity (Percentage Probability of Individuals of Gaining Immunity After Recovery, float)
RecP = Length of Recovery Period (Time Range in which the Disease is Present in an Individual, array)

########################DEFINING PATIENT CLASS#################################

import random
import matplotlib.pyplot as plt
import numpy as np

class Dossier(): #Creates an object (like list or array) that has its own modules
def __init__(self, x, y, PImmune, IncP, InfP, RecP): #Automatically run as soon as object is created
    #x and y - int
    #PImmune - float
    #IncP, InfP and RecP - lists
self.xLoc = x #Stores the inputted x location in the object
self.yLoc = y #Same as above

vaccination = random.random()
if vaccination <= PImmune: #If the "roll" is below the PImmune probability
    self.Status = "IMMUNE" #The Individual becomes immune
else: self.Status = "SUSCEPTIBLE" #If not, the individual becomes susceptible

self.IncLen = random.choice(IncP) #Picks a incubation rate out of the IncP list
self.InfLen = random.choice(InfP) #Same as above for infectious rate
self.RecLen = random.choice(RecP) #Same as above for recovery rate
self.IncState = False #Sets incubation state as false
self.InfState = False #Same as above for infectious rate
self.RecState = False #Same as above for recovery rate

def Incubation(self, I, Imn):
    self.IncLen -= 1 #When the function is called, it first decreases the incubation period
    if self.IncLen >= 0: #While the incubation period is more than 0, the IncState is set to
        self.IncState = True
    else: self.IncState = False #Otherwise, the IncState is False. In other words, the
        patient stops incubating.
    if self.IncState == False: #Once the patient finishes incubating..
        luck = random.random()
        if luck <= I: #If their "roll" is less than Infectivity...
            self.Status = "INFECTIOUS" #They become Infectious
        else:
            antibody = random.random() #If their "roll" is more than Immunity
            if antibody <= Imn: #If their "roll" is smaller than the Immunity
                self.Status = "IMMUNE" #They become Immune.
            else:
                self.Status = "SUSCEPTIBLE" #Otherwise, they become susceptible.

def Infection(self, Mor): #Same concepts as incubation.
    self.InfLen -= 1
    if self.InfLen >= 0:
        self.InfState = True
    else: self.InfState = False
    if self.InfState == False:
        hopesndreams = random.random()
if hopesndreams <= Mor:
    self.Status = "DEAD"
else: self.Status = "INCUBATING"

def Recovery(self, Imn): #Activated when patient goes into incubation or infection
    self.RecLen -= 1
    if self.RecLen >= 0:
        self.RecState = True
    else: self.RecState = False

    if self.RecState == False:
        antibody = random.random()
        if antibody <= Imn:
            self.Status = "IMMUNE"
        else: self.Status = "SUSCEPTIBLE"

def Copy(self):
    new_pat = Dossier(self.xLoc, self.yLoc, 0, range(2,7), range(2,7), range(2,7))
    #^These input values doesn't matter cuz we're gonna update it anyway
    new_pat.Status = self.Status
    new_pat.IncLen = self.IncLen
    new_pat.InfLen = self.InfLen
    new_pat.RecLen = self.RecLen
    new_pat.IncState = self.IncState
    new_pat.InfState = self.InfState
    new_pat.RecState = self.RecState
    return new_pat

#define defining functions

def Spread(Mob, NBR):
    if NBR.Status == "INFECTIOUS":
        contagion = random.random()
        if contagion <= Mob: #If the "roll" is lower than mobility, then the patient is infected by the virus
            return True
        else:
            return False
    return False

def Plot(Patient, Population):
    for P in range(1, Population+1):
x = Patient[P].xLoc
y = Patient[P].yLoc

if Patient[P].Status == 'SUSCEPTIBLE':
    colour = "white"
elif Patient[P].Status == 'IMMUNE':
    colour = "blue"
elif Patient[P].Status == 'INFECTIOUS':
    colour = "red"
elif Patient[P].Status == 'INCUBATING':
    colour = "yellow"
elif Patient[P].Status == 'DEAD':
    colour = "black"

if colour == "white":
    plt.plot(x, y, "wo")
elif colour == "blue":
    plt.plot(x, y, "bo")
elif colour == "red":
    plt.plot(x, y, "ro")
elif colour == "yellow":
    plt.plot(x, y, "yo")
elif colour == "black":
    plt.plot(x, y, "ko")
plt.show()

def DeepCopy(Old): #Takes in dict, returns deep copy
    New = dict()
    for i in range(1, len(Old)+1):
        New[i] = Old[i].Copy()
    return New

if __name__ == '__main__':
    SusFit = False
    IncFit = False
    InfFit = False
    ImmFit = False
    DeaFit = False

    if SusFit:
        SUSCEPTIBLEquery = []
        with open("SUSCEPTIBLEfit.csv", "r") as SUSCEPTIBLEfit:
for line in SUSCEPTIBLEfit:
    SUSCEPTIBLEquery.append(tuple([int(x) for x in line[:-1].split(",")]))
    preset = SUSCEPTIBLEquery
    status = "SusFit"

if IncFit:
    INCUBATINGquery = []
    with open("INCUBATINGfit.csv", "r") as INCUBATINGfit:
        for line in INCUBATINGfit:
            INCUBATINGquery.append(tuple([int(x) for x in line[:-1].split(",")]))
    preset = INCUBATINGquery
    status = "IncFit"

if InfFit:
    INFECTIOUSquery = []
    with open("INFECTIOUSfit.csv", "r") as INFECTIOUSfit:
        for line in INFECTIOUSfit:
            INFECTIOUSquery.append(tuple([int(x) for x in line[:-1].split(",")]))
    preset = INFECTIOUSquery
    status = "InfFit"

if ImmFit:
    IMMUNEquery = []
    with open("IMMUNEfit.csv", "r") as IMMUNEfit:
        for line in IMMUNEfit:
            IMMUNEquery.append(tuple([int(x) for x in line[:-1].split(",")]))
    preset = IMMUNEquery
    status = "ImmFit"

if DeaFit:
    DEADquery = []
    with open("DEADfit.csv", "r") as DEADfit:
        for line in DEADfit:
            DEADquery.append(tuple([int(x) for x in line[:-1].split(",")]))
    preset = DEADquery
    status = "DeaFit"

""
INPUTS:
time: the current time value
y: the current value of the function
query: the list of points (t, value) to fit against
OUTPUTS:
dy: the deviation from the query
fit = False
if fit == True:
    def Fit(time, y, query = preset):
        for i in query:
            if time == i[0]:
                dy = y - i[1]
            else:
                dy = "NONE"
        return dy
else:
    fit = False

#INITIALIZE SIMULATION

def main(job_id, params):
    Parameters = [params["I"], params["InfP"], params["Mob"], params["IncP"],
                  params["Mor"], params["Imn"], params["RecP"]]
    return StochasticEPIMOD(Parameters)

def StochasticEPIMOD(Parameters, xDimension = 20, yDimension = 20, tElapsed = 365, flux =
                      True, move_range = 0.5, Plot = True, Print = True, fit = fit):
    #INITIAL VALUES
    Population = xDimension*yDimension

    #PARAMETERS
    PImmune = 0.0
    I = Parameters[0] #Percentage
    InfP = range(int(Parameters[1])-3, int(Parameters[1])+4)
    Mob = Parameters[2] #Percentage
    IncP = range(int(Parameters[3])-3, int(Parameters[3])+4)
    Mor = Parameters[4] #Percentage
    Imn = Parameters[5] #Percentage
    RecP = range(int(Parameters[6])-3, int(Parameters[6])+4)

    #PATIENTS
    Patient = dict()
    ID = 0
    Pos2Pat = np.zeros((yDimension, xDimension))
    for x in range(xDimension):
        for y in range(yDimension):
ID += 1
Patient[ID] = Dossier(x, y, Plimmune, IncP, InfP, RecP)
Pos2Pat[y,x] = ID

#GENERATING PATIENT ZERO
x = xDimension/2
y = yDimension/2
for P in range(1, ID+1):
    if Patient[P].xLoc == x and Patient[P].yLoc == y:
        Patient[P].Status = "INFECTIOUS"

Record = DeepCopy(Patient)

SUSCEPTIBLEtrend = []
INCUBATINGtrend = []
INFECTIOUStrend = []
IMMUNEtrend = []
DEADtrend = []
residuals = []

###########################SIMULATION BEGINS###########################
for t in range(tElapsed):
    # Plot(Patient, Population)
    for P in range(1, ID+1):
        #STATUS PRIORITY: DEAD = IMMUNE > INFECTIOUS = INCUBATING > SUSCEPTIBLE
        if Record[P].Status == "SUSCEPTIBLE":
            n = 8
            x = Patient[P].xLoc
            y = Patient[P].yLoc
            while Patient[P].Status != "INFECTIOUS" and Patient[P].Status != "INCUBATING" and n > 0:
                if n == 8:
                    infection = Spread(Mob, Record[Pos2Pat[y,(x+1)%xDimension]])
                #Right
                if infection == True:
                    infection = random.random()
                if infection <= 1:
                    Patient[P].Status = "INFECTIOUS"
                else: Patient[P].Status = "INCUBATING"
            elif n == 7:
                infection = Spread(Mob, Record[Pos2Pat[y,(x-1)%xDimension]])
if infection == True:
    infection = random.random()
    if infection <= I:
        Patient[P].Status = "INFECTIOUS"
    else: Patient[P].Status = "INCUBATING"

elif n == 6:
    infection = Spread(Mob, Record[Pos2Pat[(y+1)%yDimension,x]])

else if n == 5:
    infection = Spread(Mob, Record[Pos2Pat[(y-1)%yDimension,x]])

else if n == 4:
    infection = Spread(Mob, Record[Pos2Pat[(y+1)%yDimension,(x+1)%xDimension]]) #Upper Right
    if infection == True:
        infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else: Patient[P].Status = "INCUBATING"

else if n == 3:
    infection = Spread(Mob, Record[Pos2Pat[(y-1)%yDimension,(x-1)%xDimension]]) #Lower Left
    if infection == True:
        infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else: Patient[P].Status = "INCUBATING"

else if n == 2:
    infection = Spread(Mob, Record[Pos2Pat[(y+1)%yDimension,(x-1)%xDimension]]) #Upper Left
    if infection == True:
        infection = random.random()
        if infection <= I:
elif n == 1:
    infection = Spread(Mob, Record[Pos2Pat[(y-1)%yDimension,(x+1)%xDimension]]) #Lower Right
    if infection == True:
        infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else: Patient[P].Status = "INCUBATING"
        n -= 1

elif Record[P].Status == "INFECTIOUS":
    Patient[P].Recovery(Imn)
    if Patient[P].RecState == True: #If the patient is still infectious...
        Patient[P].Infection(Mor)

elif Record[P].Status == "INCUBATING":
    Patient[P].Recovery(Imn)
    if Patient[P].RecState == True: #If the patient is still incubating...
        Patient[P].Incubation(I,Imn)

eelif Record[P].Status == "IMMUNE": #Removed from population
    pass

eelif Record[P].Status == "DEAD": #Removed from population
    pass

#UPDATE MOVING
if flux == True:
    for x in range(xDimension):
        for y in range(yDimension):
            if Patient[Pos2Pat[y,x]].Status != "DEAD":
                move = random.random()
                if move <= move_range:
                    swap = random.randint(1, 4)
                    if swap == 1: #switch with above
                        oriID = Pos2Pat[y, x]
                        newID = Pos2Pat[(y+1)%yDimension, x]
                        Patient[oriID].yLoc = (y+1)%yDimension
                        Patient[newID].yLoc = y
                        Pos2Pat[y,x] = newID
                        Pos2Pat[(y+1)%yDimension, x] = oriID
elif swap == 2: #switch with below
    oriID = Pos2Pat[y, x]
    newID = Pos2Pat[(y-1)%yDimension, x]
    Patient[oriID].yLoc = (y-1)%yDimension
    Patient[newID].yLoc = y
    Pos2Pat[y,x] = newID
    Pos2Pat[(y-1)%yDimension, x] = oriID

elif swap == 3: #switch with left
    oriID = Pos2Pat[y, x]
    newID = Pos2Pat[y, (x-1)%yDimension]
    Patient[oriID].xLoc = (x-1)%xDimension
    Patient[newID].xLoc = x
    Pos2Pat[y,x] = newID
    Pos2Pat[y, (x-1)%yDimension] = oriID

elif swap == 4: #switch with right
    oriID = Pos2Pat[y, x]
    newID = Pos2Pat[y, (x+1)%yDimension]
    Patient[oriID].xLoc = (x+1)%xDimension
    Patient[newID].xLoc = x
    Pos2Pat[y,x] = newID
    Pos2Pat[y, (x+1)%yDimension] = oriID

SUSCEPTIBLE = 0
INFECTIOUS = 0
INCUBATING = 0
IMMUNE = 0
DEAD = 0

for P in range(1, ID+1):
    if Patient[P].Status == "SUSCEPTIBLE":
        SUSCEPTIBLE += 1
    elif Patient[P].Status == "INFECTIOUS":
        INFECTIOUS += 1
    elif Patient[P].Status == "INCUBATING":
        INCUBATING += 1
    elif Patient[P].Status == "IMMUNE":
        IMMUNE += 1
    elif Patient[P].Status == "DEAD":
        DEAD += 1

SUSCEPTIBLEtrend.append(SUSCEPTIBLE)
INFECTIOUStrend.append(INFECTIOUS)
INCUBATINGtrend.append(INCUBATING)
IMMUNEtrend.append(IMMUNE)
DEADtrend.append(DEAD)

if fit == True:
    if status == "SusFit":
        y = SUSCEPTIBLE
    elif status == "IncFit":
        y = INCUBATING
    elif status == "InfFit":
        y = INFECTIOUS
    elif status == "ImmFit":
        y = IMMUNE
    elif status == "DeaFit":
        y = DEAD

diffs = Fit(t, y)
if diffs != "NONE":
    residuals.append(diffs**2)

Record = DeepCopy(Patient)

if Plot:
    plt.plot(SUSCEPTIBLEtrend, "0.5", label = "SUSCEPTIBLE")
    plt.plot(INCUBATINGtrend, "y", label = "INCUBATING")
    plt.plot(INFECTIOUStrend, "r", label = "INFECTIOUS")
    plt.plot(IMMUNEtrend, "b", label = "IMMUNE")
    plt.plot(DEADtrend, "k", label = "DEAD")

    plt.legend(loc="best")
    plt.title("ebolaSTOinfSIM")
    plt.ylabel("Number of Individuals")
    plt.xlabel("Time (in days)")
    # plt.show()
    plt.savefig("ebolaSTOinfSIM2.pdf")

if Print:
    print "SUSCEPTIBLE = %i" %SUSCEPTIBLE
    print "INFECTIOUS = %i" %INFECTIOUS
    print "INCUBATING = %i" %INCUBATING
    print "IMMUNE = %i" %IMMUNE
    print "DEAD = %i" %DEAD
    print "Percent Dead = %f" %(float(DEAD)/float(Population))

    return abs(sum(residuals))
# PercentDead = float(DEAD)/float(Population)
# return PercentDead

if __name__ == '__main__':
    StochasticEPIMOD([0.6834, 14, 0.0163, 19, 0.9512, 0.8861, 22])
STOCHASTIC EPIDEMIC MODEL
Uma Wu & LiQing Wang

Discussed With: Noah Bayless, Michael Gelbart

---

#####PATIENT INFORMATION#####

xLoc = range(0, x_Dim)
yLoc = range(0, y_Dim)
Status = IMMUNE, SUSCEPTIBLE, INFECTIOUS, INCUBATING, or DEAD
tExist = time elapsed since start of simulation
tInc = time incubating
tInf = time infectious
tRec = time recovering

#####PARAMETERS#####

PImmune = Pre-Immunity (PERCENTAGE PROBABILITY of Population that is Pre-Immune to Disease, float)
I = Infectivity (PERCENTAGE PROBABILITY of Displaying Symptoms When Infected, float)
InfP = Length of Infectious Period (TIME RANGE in which the Disease is Infectious, array)
Mob = Mobility (PERCENTAGE PROBABILITY of Contracting the Disease from an Adjacent Diseased Individual, float)
IncP = Length of Incubation Period (TIME RANGE in which the Disease does not produce Symptoms (In an Infective Individual), array)
Mor = Morbidity (PERCENTAGE PROBABILITY of Infective Individuals Dying from the Disease, float)
Imn = Immunity (PERCENTAGE PROBABILITY of Individuals of Gaining Immunity After Recovery, float)
RecP = Length of Recovery Period (TIME RANGE in which the Disease is Present in an Individual, array)

---

from StochasticEPIMOD1 import Dossier
from StochasticEPIMOD1 import Spread
from StochasticEPIMOD1 import Plot
from StochasticEPIMOD1 import DeepCopy

import random
import matplotlib.pyplot as plt
import numpy as np
from scipy.optimize import minimize
def main(job_id, params):
    Parameters = [params["I"], params["InfP"], params["Mob"], params["IncP"],
    params["Mor"], params["Imn"], params["RecP"],
    return -StochasticEPIMOD(Parameters)

def StochasticEPIMOD(Parameters, xDimension = 20, yDimension = 20, tElapsed = 365, flux =
    True, move_range = 0.5, Plot = False, Print = False):
    #INITIAL VALUES
    Population = xDimension*yDimension

    #PARAMETERS
    PImmune = 0.0
    I = Parameters[0] #Percentage
    InfP = range(int(Parameters[1]) -3, int(Parameters[1])+4)
    Mob = Parameters[2] #Percentage
    IncP = range(int (Parameters[3])-3, int(Parameters[3])+4)
    Mor = Parameters[4] #Percentage
    Imn = Parameters[5] #Percentage
    RecP = range(int(Parameters[6]) -3, int(Parameters[6])+4)

    #PATIENTS
    Patient = dict()
    ID = 0
    Pos2Pat = np.zeros((yDimension, xDimension))
    for x in range(xDimension):
        for y in range(yDimension):
            ID += 1
            Patient[ID] = Dossier(x, y, PImmune, IncP, InfP, RecP)
            Pos2Pat[y,x] = ID

    #GENERATING PATIENT ZERO
    x = xDimension/2
    y = yDimension/2
    for P in range(1, ID+1):
        if Patient[P].xLoc == x and Patient[P].yLoc == y:
            Patient[P].Status = "INFECTIOUS"

    Record = DeepCopy(Patient)

    #****************************SIMULATION
    BEGINS*****************************
    for t in range(tElapsed):
        for P in range(1,ID+1):
            #STATUS PRIORITY: DEAD = IMMUNE > INFECTIOUS = INCUBATING >
if Record[P].Status == "SUSCEPTIBLE":
    n = 8
    x = Patient[P].xLoc
    y = Patient[P].yLoc
    while Patient[P].Status != "INFECTIOUS" and Patient[P].Status != "INCUBATING" and n > 0:
        if n == 8:
            infection = Spread(Mob, Record[Pos2Pat[y,(x+1)%xDimension]])
        # Right
        else if infection == True:
            infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else:
            Patient[P].Status = "INCUBATING"
        elif n == 7:
            infection = Spread(Mob, Record[Pos2Pat[y,(x-1)%xDimension]])
        # Left
        else if infection == True:
            infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else:
            Patient[P].Status = "INCUBATING"
        elif n == 6:
            infection = Spread(Mob, Record[Pos2Pat[(y+1)%yDimension,x]])
        # Above
        else if infection == True:
            infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else:
            Patient[P].Status = "INCUBATING"
        elif n == 5:
            infection = Spread(Mob, Record[Pos2Pat[(y-1)%yDimension,x]])
        # Below
        else if infection == True:
            infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else:
            Patient[P].Status = "INCUBATING"
        elif n == 4:
            infection = Spread(Mob, Record[Pos2Pat[(y+1)%yDimension,(x+1)%xDimension]])
        # Upper Right
            if infection == True:
                infection = random.random()
if infection <= I:
    Patient[P].Status = "INFECTIOUS"
else: Patient[P].Status = "INCUBATING"

elif n == 3:
    infection = Spread(Mob, Record[Pos2Pat[(y-1)%yDimension,(x-1)%xDimension]]) #Lower Left
    if infection == True:
        infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else: Patient[P].Status = "INCUBATING"

elif n == 2:
    infection = Spread(Mob, Record[Pos2Pat[(y+1)%yDimension,(x-1)%xDimension]]) #Upper Left
    if infection == True:
        infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else: Patient[P].Status = "INCUBATING"

elif n == 1:
    infection = Spread(Mob, Record[Pos2Pat[(y-1)%yDimension,(x+1)%xDimension]]) #Lower Right
    if infection == True:
        infection = random.random()
        if infection <= I:
            Patient[P].Status = "INFECTIOUS"
        else: Patient[P].Status = "INCUBATING"

    n -= 1

elif Record[P].Status == "INFECTIOUS":
    Patient[P].Recovery(lmn)
    if Patient[P].RecState == True: #If the patient is still infectious...
        Patient[P].Infection(Mor)

elif Record[P].Status == "INCUBATING":
    Patient[P].Recovery(lmn)
    if Patient[P].RecState == True: #If the patient is still incubating...
        Patient[P].Incubation(l,lmn)

elif Record[P].Status == "IMMUNE": #Removed from population
    pass
elif Record[P].Status == "DEAD": #Removed from population
    pass
#UPDATE MOVING
if flux == True:
    for x in range(xDimension):
        for y in range(yDimension):
            if Patient[Pos2Pat[y,x]].Status != "DEAD":
                move = random.random()
                if move <= move_range:
                    swap = random.randint(1, 4)
                    if swap == 1: #switch with above
                        oriID = Pos2Pat[y, x]
                        newID = Pos2Pat[(y+1)%yDimension, x]
                        Patient[oriID].yLoc = (y+1)%yDimension
                        Patient[newID].yLoc = y
                        Pos2Pat[y,x] = newID
                        Pos2Pat[(y+1)%yDimension, x] = oriID
                    elif swap == 2: #switch with below
                        oriID = Pos2Pat[y, x]
                        newID = Pos2Pat[(y-1)%yDimension, x]
                        Patient[oriID].yLoc = (y-1)%yDimension
                        Patient[newID].yLoc = y
                        Pos2Pat[y,x] = newID
                        Pos2Pat[(y-1)%yDimension, x] = oriID
                    elif swap == 3: #switch with left
                        oriID = Pos2Pat[y, x]
                        newID = Pos2Pat[y, (x-1)%yDimension]
                        Patient[oriID].xLoc = (x-1)%xDimension
                        Patient[newID].xLoc = x
                        Pos2Pat[y,x] = newID
                        Pos2Pat[y, (x-1)%yDimension] = oriID
                    elif swap == 4: #switch with right
                        oriID = Pos2Pat[y, x]
                        newID = Pos2Pat[y, (x+1)%yDimension]
                        Patient[oriID].xLoc = (x+1)%xDimension
                        Patient[newID].xLoc = x
                        Pos2Pat[y,x] = newID
                        Pos2Pat[y, (x+1)%yDimension] = oriID
SUSCEPTIBLE = 0
INFECTIOUS = 0
INCUBATING = 0
IMMUNE = 0
DEAD = 0

for P in range(1, ID+1):
    if Patient[P].Status == "SUSCEPTIBLE":
        SUSCEPTIBLE += 1
    elif Patient[P].Status == "INFECTIOUS":
        INFECTIOUS += 1
    elif Patient[P].Status == "INCUBATING":
        INCUBATING += 1
    elif Patient[P].Status == "IMMUNE":
        IMMUNE += 1
    elif Patient[P].Status == "DEAD":
        DEAD += 1

Record = DeepCopy(Patient)

PercentDead = float(DEAD)/float(Population)
return PercentDead

def Average(Parameters, tests = 1000):
    total = 0.0
    results = np.zeros(tests)
    for n in range(tests):
        results[n] = float(StochasticEPIMOD(Parameters))
        print n
    print np.mean(results)
    print np.std(results)
    print results
    return np.mean(results), np.std(results), results

data = open("StochasticEPIMOD1SARSresultsDEAD.txt", "w")

Combination = [0.2860000, 9, 0.256050, 5, 0.34989, 0.462340, 17]
average, stdev, results = Average(Combination)
data.write("Average: %f" %average)
data.write("Standard Deviation: %f" %stdev)
for i in range(len(results)-1):
    data.write("%f" %results[i])
<table>
<thead>
<tr>
<th>Infectivity</th>
<th>Percentage Death</th>
<th>Infection Rate</th>
<th>Mobility</th>
<th>Incubation Rate</th>
<th>Mortality</th>
<th>Percentage Death</th>
<th>Immunity</th>
<th>Recovery Rate</th>
<th>Percentage Death</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.587594</td>
<td>0.63814</td>
<td>0.92235</td>
<td>0.92314</td>
<td>1</td>
<td>0.57929</td>
<td>0.92235</td>
<td>0.00002</td>
<td>0.57964</td>
</tr>
<tr>
<td>0.9</td>
<td>0.7782</td>
<td>0.72177</td>
<td>0.91542</td>
<td>0.91627</td>
<td>1</td>
<td>0.86648</td>
<td>0.91666</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>0.8</td>
<td>0.92258</td>
<td>0.82672</td>
<td>0.91895</td>
<td>0.90518</td>
<td>0.9</td>
<td>0.92258</td>
<td>0.92258</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>0.7</td>
<td>0.88648</td>
<td>0.88673</td>
<td>0.92223</td>
<td>0.91854</td>
<td>0.8</td>
<td>0.92223</td>
<td>0.92223</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>0.6</td>
<td>0.91566</td>
<td>0.85485</td>
<td>0.92223</td>
<td>0.91566</td>
<td>0.7</td>
<td>0.91566</td>
<td>0.91566</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>0.5</td>
<td>0.91525</td>
<td>0.79698</td>
<td>0.92223</td>
<td>0.91525</td>
<td>0.6</td>
<td>0.91525</td>
<td>0.91525</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>0.4</td>
<td>0.91566</td>
<td>0.68413</td>
<td>0.92223</td>
<td>0.91566</td>
<td>0.5</td>
<td>0.91566</td>
<td>0.91566</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>0.3</td>
<td>0.91525</td>
<td>0.4279</td>
<td>0.92223</td>
<td>0.91525</td>
<td>0.4</td>
<td>0.91525</td>
<td>0.91525</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>0.2</td>
<td>0.91566</td>
<td>0.2</td>
<td>0.92223</td>
<td>0.91566</td>
<td>0.3</td>
<td>0.91566</td>
<td>0.91566</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>0.1</td>
<td>0.91525</td>
<td>0.0</td>
<td>0.92223</td>
<td>0.91525</td>
<td>0.2</td>
<td>0.91525</td>
<td>0.91525</td>
<td>0.00002</td>
<td>0.81512</td>
</tr>
<tr>
<td>PARAMETERS</td>
<td>STOCHASTIC</td>
<td>DETERMINISTIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Initially Susceptible Population/ Total Pop</td>
<td>N/A: There was not enough computational power to simulate population as a parameter.</td>
<td>(PHASE I only) Total number of people in each compartment. Remains constant throughout simulation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Infectivity (I)</td>
<td>Percentage Probability of Population that Displays Symptoms When Infected</td>
<td>Percentage of Population that Displays Symptoms When Infected</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length of Infectious Period (InfP)</td>
<td>(+/-3) Possible Time Range in which the Disease is Infectious</td>
<td>Time in which the Disease is Infectious</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mobility (Mob)</td>
<td>Percentage Probability of Contracting the Disease from an Adjacent Diseased Individual</td>
<td>Number of Healthy Individuals that a Diseased Individual may Infect</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length of Incubation Period (IncP)</td>
<td>(+/-3) Possible Time Range in which the Disease does not produce Symptoms (In an Infective Individual)</td>
<td>Time in which the Disease does not produce Symptoms (In an Infective Individual)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Length of Recovery Period (RecP)</td>
<td>(+/-3) Possible Time Range in which the Disease is Present in an Individual</td>
<td>Time in which the Disease is Present in an Individual</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mortality (Mor)</td>
<td>Percentage Probability of Infective Individuals Dying from the Disease</td>
<td>Percentage of Infective Individuals Dying from the Disease</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Immunity (Imn)</td>
<td>Percentage Probability of Individuals of Gaining Immunity After Recovery</td>
<td>Percentage of Individuals that Gain Immunity Upon Recovery</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SARS Data points for Fitting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>78</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>98</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>103</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>165</td>
<td>144</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>154</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>186</td>
<td>461</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>187</td>
<td>478</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>188</td>
<td>495</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>189</td>
<td>506</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>202</td>
<td>666</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>204</td>
<td>689</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>210</td>
<td>750</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>211</td>
<td>754</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>216</td>
<td>772</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>221</td>
<td>784</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>228</td>
<td>799</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>229</td>
<td>799</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>231</td>
<td>804</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>238</td>
<td>809</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>239</td>
<td>810</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>242</td>
<td>811</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>243</td>
<td>812</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>244</td>
<td>812</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>249</td>
<td>812</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>252</td>
<td>812</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ebola Data Points for Fitting</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>279</td>
<td>12713</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>312</td>
<td>13697</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>347</td>
<td>14487</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>382</td>
<td>15151</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>413</td>
<td>15854</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>446</td>
<td>16236</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>479</td>
<td>16470</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>510</td>
<td>16763</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>