SEIR Simulations on a Human Interaction Network

Max Wu[a,1], **Kevin Zhang**[a,1], **Breitling Snyder**[b,1], **Laura Pasquale**[b,1], **and Randy Gonzalez**[b,1]

[a]Department of Computer Science, University of California, Los Angeles, 90024; [b]Department of Mathematics, University of California, Los Angeles, 90024

This manuscript was compiled on January 13, 2021

Using a temporal network that records face to face interactions, we implemented an SEIR model to predict the evolution of pathogen transmission and make conclusions about the potential impacts a viral disease might have on a population. We use Markovian data generation to expand our initial dataset, and modeled and compared different disease control methods by alteration of the parameters in our model.

Data Generation | Temporal Networks | Disease Modeling

The Susceptible-Exposed-Infected-Recovered model for spread of disease is one of the simplest models that can provide a relevant and applicable framework through which epidemics can be analyzed. When applied to a time-dependent network of human interactions, this model can represent the spread of an infection throughout a population in real time. The study of disease transmission on networks is increasingly relevant, as the spread of the Coronavirus disease 2019 (COVID-19) has majorly impacted the global population.

While simple mathematical and scientific observations had been made for the SEIR model by using probability alone, we decided to analyze this model with information provided to us from a network detailing social interactions over a period of time. This helps further examinations of disease spread in an applied setting by observing the model with its results using knowledge of node interactions and techniques of network analysis such as Louvain's Algorithm of community detection, centrality measures, and possible changes given certain nodes be removed from the network. To make this as insightful as possible, we also generated more raw data based on the original data by utilizing our own data extension method.

Haslemere Human Interaction Dataset

Firstly, we need to prepare a human interaction dataset in order to run our SEIR simulations on.

The Haslemere Human Interaction Dataset tracks contact distances between 469 people over a 3-day period in Haslemere, England. (1) The 3 days are split into 572 timesteps of 5-minute intervals (night-time interactions are omitted). At every timestep, if any pair of people are less than 50 meters apart, their distance is recorded as a row in the dataset.

Using this dataset, we can model the probability of disease exposure between two individuals as a function of distance, which will be discussed later.

We obtained this data directly from a publicly available GitHub repository published by the original researchers. (2)

Data Generation. Our dataset generation code is available here: github.com/legitmaxwu/math168-networks

In order to run SEIR simulations on our dataset, we needed at least a month's worth of interaction data, which is 10x more than the Haslemere dataset provides. The original collectors of the Haslemere dataset elected to continually cycle through the same dataset for simulations exceeding 3 days. However, this is unrealistic as people are unlikely to have the same interactions with the same people over and over again without any new contacts in a month-long period - even grocery shopping adds unknown people to a person's daily routine.

Temporal Graph Shuffling. An intuitive way to generate temporal graph data is to shuffle the base dataset in a variety of ways. There are 3 baseline methods for shuffling temporal graph data. (3) They are listed below, along with a potential drawback and their time complexity (with T and N being the number of timesteps and nodes in the base dataset):

1. *Link shuffling*: Preserve timelines for each edge, but shuffle entire edge timelines such that they occur between a different pair of nodes. **Drawbacks**: Doesn't alter temporal patterns. Doesn't preserve degree distribution. **Time complexity**: $O(N^2)$.

2. *Timeline shuffling*: For each edge timeline, shuffle the events in the timeline. **Drawback**: Doesn't preserve temporal correlations. (i.e. if two people interact over many consecutive timesteps, timeline shuffling will destroy this pattern) **Time complexity**: $O(N^2T)$.

3. *Sequence shuffling*: Shuffle snapshots of the network at every timestep. **Drawback**: Doesn't preserve temporal correlations. **Time complexity**: $O(T)$.

For our purposes, link shuffling alone is not enough because it does nothing to alter patterns of interaction. However, the other two shuffling methods do a poor job of preserving temporal correlation. To alleviate these drawbacks, there exist algorithms to restrict how timesteps can be shuffled in both timeline and sequence shuffling. Shuffling algorithms can also be combined. However, these modifications often bring the

Significance Statement

The SEIR model is one of many compartmental models used to understand the spread of disease within a population over time. While observations of this model can be made based on probability alone, implementation on an empirical network of face to face interactions for a sample population can provide even more insight on how to control the spread of a virus. Using this network we are able to do a number of operations that help us quantify disease spread on a real world network and compare the effectiveness of disease control methods.

algorithm time complexity above $O(T)$, which could prevent them from scaling to generate from base datasets containing many timesteps.

Hence, we were motivated to design a different algorithm archetype: one that can run in $O(T)$ but can also modify temporal patterns without destroying temporal correlations.

Markovian Data Generation. Ultimately, we developed our own baseline approach for generating data for our experiments. The central idea behind this model is that, using human interaction from individual timesteps t, we can predict the probability that each pair of people will interact at time $t+1$:

$$P_{ij}^{\text{interact}}(t) = \begin{cases} P_{ij}^{\text{meet}} & \text{if } i \text{ and } j \text{ haven't interacted at } t-1 \\ P_{ij}^{\text{stay}} & \text{otherwise} \end{cases}$$

P_{ij}^{meet} and P_{ij}^{stay} are constants for each node pair (i, j), empirically predetermined from the dataset as follows: (let $I_{ij}(t)$ denote the event of nodes i and j interacting at timestep t)

$$P_{ij}^{\text{meet}} = P(I_{ij}(t) \mid \text{not } I_{ij}(t-1))$$

$$P_{ij}^{\text{stay}} = P(I_{ij}(t) \mid I_{ij}(t-1))$$

To get these actual values from our dataset, here are the more explicit equations:

$$P_{ij}^{\text{meet}} = \frac{\#\text{ cases where } I_{ij}(t) \text{ but not } I_{ij}(t-1)}{\#\text{ cases where not } I_{ij}(t)}$$

$$P_{ij}^{\text{stay}} = \frac{\#\text{ cases where } I_{ij}(t) \text{ and } I_{ij}(t-1)}{\#\text{ cases where } I_{ij}(t)}$$

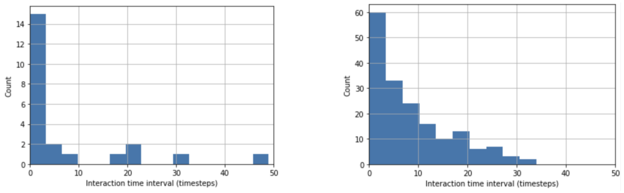
Using these formulas, we can calculate the probability that any two nodes interact at any timestep for data generation purposes.

Interaction isolation. One issue with this methodology is that if nodes i and j never interact in the base dataset, then P_{ij}^{meet} and P_{ij}^{stay} are both 0. In real life, this is not very realistic. To alleviate this, we bump all probability values equal to 0 by $1/T$, where T is the number of timesteps generated. While this is still relatively unrealistic, it helps the disease spread more evenly in our SEIR simulations.

Determining Interaction Distance. If two people interact, what is their distance of interaction? We use a simple method here, sampling distance $d_{ij}(t)$ from one of two normal distributions depending on whether nodes i and j are "friends" or not.

$$d_{ij} = \begin{cases} \lfloor d \rfloor \sim \mathcal{N}(6, 3) & \text{if } i \text{ and } j \text{ are "friends"} \\ \lfloor d \rfloor \sim \mathcal{N}(25, 10) & \text{otherwise} \end{cases}$$

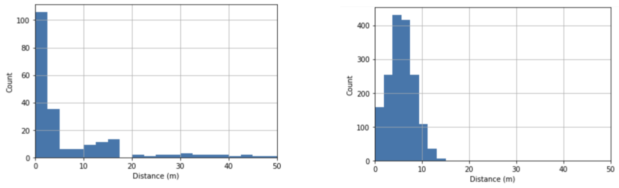
Sampled values are clamped between 0 and 50 to avoid unreasonable values. The intuition here is that "friends" are likely to stay closer to each other during their interactions than passerby. Our hardcoded distributions were based off of qualitative observations on the base dataset. With these in mind, the question still stands: How do we know if i and j are "friends"?



(a) Base dataset

(b) Generated dataset

Fig. 1. The D_{ij} (interaction duration) distribution for the base dataset and generated datasets, between nodes 1 and 390.



(a) Base dataset

(b) Generated dataset

Fig. 2. d_{ij} (interaction distance) distribution for base dataset and generated datasets, between nodes 1 and 390.

Determining "Friendship" between two people. We denote nodes i and j to be "friends" if $P_{ij}^{\text{stay}} > 0.7$. Every time i and j interact, let D_{ij} denote the duration of interaction, or the number of consecutive interaction timesteps before they part ways. If our threshold is satisfied, then we have

$$E(D_{ij}) > \sum_{k=0}^{\infty} k(0.7)^k(1-0.7)$$

$$E(D_{ij}) > \frac{1}{1-0.7} = 3.3$$

So, we have $E(D_{ij}) > 3.3$ timesteps, or 16.5 minutes. If i and j are expected to interact for 16.5 minutes, we assume that they're not usually just passing by each other, so they are "friends".

Evaluation. Using our models above, we generated 5,000 timesteps in addition to the 572 timesteps provided by the base dataset.

We believe that our Markovian data generation algorithm is a solid baseline approach that can generate new temporal patterns while preserving temporal correlations, as well as run in $O(N^2T)$ time complexity. Note that this is linear on T. However, how well does it preserve interaction duration distributions as well as distance? We investigate the interactions between person 1 and 390 (two people considered to be "friends") in the dataset. Figures 1 and 2 show the distributions of D_{ij} and d_{ij} for nodes 1 and 390 across the base dataset in comparison with the generated dataset.

Interaction duration. As expected, the interaction durations follow a geometric distribution in the generated dataset. It also emulates the distribution of the base dataset, though it is less concentrated on lower values (below 3 timesteps).

Interaction distance. Since i and j are friends, it's expected that d_{ij} follows a normal distribution centered around 6. However, this does not seem to be very accurate in capturing the base distribution. In particular, even if i and j are close, there are still many instances where their interaction distance is much higher than 6, which the normal distribution does not capture.

Future Considerations. Our model is extremely simple, and therefore has much room for improvement. We list a few considerations for future work below:

1. Systematically determining distance distributions to sample d_{ij} from the dataset instead of hard-coding. Also, using normal distributions does not seem ideal. (Another possible candidate is chi-squared)

2. Interaction isolation: Another way to alleviate this problem (described previously) is to introduce link shuffling across certain time intervals, which we could try in the future. This would not break linear time to T.

3. Distance rule: If nodes i and j are both close to k, then they should also be somewhat close to each other. Because our Markovian model does not take this into account, our generated data does not capture the full effect of people hanging out in large groups.

SEIR Model

The SEIR Model for spread of disease is traditionally governed by a system of differential equations. The rate of people going from susceptible to exposed is characterized as the exposure rate, which grows as the number of infected individuals increases. A standard hypothesis is that the exposure rate is the transmission rate (or rate of contact, β) multiplied by the probability of infection given that contact had been made, which would be the ratio between the infected population and the entire population N (4). The rate of people going from exposed to infected is known as the incubation rate (σ), and the rate of people going from infected to recovered is known as the recovery rate (γ). As such, we obtain the following system of equations dictating the rate of change for each of the four groups of the population:

$$\frac{dS}{dt} = -\frac{\beta SI}{N}$$

$$\frac{dE}{dt} = \frac{\beta SI}{N} - \sigma E$$

$$\frac{dI}{dt} = \sigma E - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

where $S(t)$ = the number of susceptible people, $E(t)$ = the number of exposed people, $I(t)$ = the number of infected people, $R(t)$ = the number of people who recovered, β = contact rate, σ = incubation rate, γ = recovery rate, and N = the observable population. Any value not dependent on time t is assumed to be constant.

Implementation. Our simulation implementation is available here: github.com/legitmaxwu/math168-networks

To simulate the SEIR model over our human interaction network, each node on the network is assigned an initial state at the first timestep, from potential states S = Susceptible, E = Exposed, I = Infected, and R = Recovered. In this case, nodes are either assigned S = Susceptible or I = Infected at the onset of the simulation. Then, at each subsequent timestep t a node will either change states or remain in the same state, depending on exposure. The way in which a node changes states at a time t is as follows:

For each susceptible node i, If a susceptible node has a shared edge with an Infected node, then the susceptible node may or may not become Exposed. This depends on the node's probability of exposure, which will be discussed in detail later. During this Exposed period the node is considered infectious, and therefore can spread the disease to other nodes. This is comparable to an Asymptomatic period, which is common for diseases such as COVID-19.

Additionally, at the onset of the simulation both the Exposed and Infected states are assigned durations. We set $\zeta(E) = 270$, which is approximately 1.5 days, and $\zeta(I) = 542$, which is approximately 3 days. All Exposed nodes remain in state E for $\zeta(E)$ timesteps, and then enter state I. After $\zeta(I)$ timesteps, the nodes in state I are considered recovered. For the purpose of this model, state R represents nodes that have either recovered from the disease or died due to it, as in both scenarios the node cannot be infected or infect others.

Probability of Exposure. As we're simulating on an interaction dataset, we would like to determine probability of exposure as a function of the distances between each node and its neighbors. (The lower the distance, the higher the probability of exposure). The determining equations are as follows:

$$P_i(t) = 1 - e^{-F_i}$$

$$F_i = \sum_j \lambda_{ij}$$

$$\lambda_{ij} = \begin{cases} ae^{-d_{ij}/\rho} & d_{ij} \leq \xi \text{ and} \\ & i \text{ is susceptible and } j \text{ is infectious} \\ 0 & \text{otherwise} \end{cases}$$

The force of infection (the higher the force, the more likely an infection occurs) between i and j, λ_{ij}, exponentially decays with distance d_{ij}. a and ρ are tuning constants affecting the magnitude and decay rate, and ξ is a cutoff point; distances past ξ do not create any force of infection. In our simulations, for simplicity, we chose $a = 1, \rho = 10$ and $\xi = 20$ just like the publishers of the Haslemere dataset. (2)

The force of infection on an individual F_i is simply the sum of the force of infections due to each of its infectious neighbors. $P_i(t)$ asymptotically increases with F_i.

For our simulations, we use $P_i(t)$ for each person as the actual probability that the person gets infected at time t. We use `np.random.random()` to choose a uniformly random number $p \in [0, 1]$. If $p < P_i(t)$, then the Susceptible node becomes Exposed.

It's interesting to note that when implementing this method in the code, using vectorized matrix operations resulted in a runtime that was 100 times faster than using traditional `if/else` logic.

Quarantining Implementation. Because a node's state is reassessed at each timestep, state Q = Quarantined can be added to the model to examine the benefits of quarantining during an epidemic. This was utilized through a boolean `hasQ`, which was passed through the function for the SEIR simulation. When `hasQ` is true, nodes in state I are no longer infectious. The node is therefore able to spread the disease during the exposed period, but will remain in the quarantined state until it is recovered. (5)

Community Detection and Partitioning

Louvain's Algorithm. In order to examine the spread of disease using the SEIR model on individual partitions, Louvain's Algorithm for community detection was utilized to identify the communities within the network. Louvain's Algorithm is a greedy algorithm for modularity that separates the nodes within a network into supercommunities. The modularity of the network is defined as

$$Q = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(g_i, g_j)$$

where A_{ij} is the weight of the edge between node i and j, $\delta(i,j)$ is defined as the Kronecker delta function-that is 1 if the group of node i and node j are in the same group or classification(i.e interaction community) and 0 otherwise-$m = \frac{1}{2}\sum_{ij} A_{ij}$ is the total number of edges, k_i is the degree of node i. This algorithm tries to achieve a maximization of the modularity in order to find communities in a graph. The algorithm works in as an *agglomerative* algorithm by performing join operations on each individual state of the algorithm. The algorithm has two phases. In the first phase, each node starts off with its own community designation $c(i)$. For each node i, we attempt to put node i into the communities of neighboring nodes and calculate the change in modularity ΔQ to the entire network. This change in modularity by putting a node i into a new community C can be computed in the following equation :

$$\Delta Q = [\frac{\sum_{in} + k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum in}{2m})^2 - (\frac{k_i}{2m})^2]$$

where $\sum_{in}$ is the sum of the weights of the links inside a community C, $\sum_{tot}$ is the sum of the weights of links incident to nodes in C, k_i is the sum of the weights of the links incident to node i, $k_{i,in}$ is the sum of weights of links from node i to nodes in community C, and $m = \sum_{ij} A_{ij}$ (6).

The first phase stops when a local maxima of modularity is attained and no movement of a node to another community can increase the modularity. In the second phase of the algorithm, we build a new network whose nodes are now the communities in the first phase. Between these nodes or "communities", we add the weight of the links as the sum of the weight of the links between nodes in the original communities. Links between nodes of the same community lead to self-loops for the community in the new network and the phase will be completed once the entire network is constructed. The combination of these two phases is run multiple times until no more changes in modularity occurs. The final resulting network gives the major communities within the network. From this, the communities can be separated into distinct partitions on which an SEIR simulation can be run. We used Louvain's algorithm on the graph we generated to obtain 40 communities on 469 nodes. We then applied our model to each of the largest 5 partitions to compare results of the individual partitions in comparison with the the model on the entire dataset.

Centrality Measures

Centrality measures are measures that quantify the importance of a node with respect to some metric. In our research, we hypothesized that since disease transfer was directly correlated to the strength of connections(inverse of the distance) between nodes or people, the betweenness centrality and eigenvector centrality measurements would be very useful in devising a method to stem the spread of infection.

Betweenness Centrality. Betweenness centrality is used to calculate the importance of a individual nodes in connecting other nodes in the graph. We calculate the betweenness centrality for each individual node i in a weighted, undirected network as follows:

$$x_i = \sum_{st} \frac{n^i_{st}}{g_{st}}$$

where n^i_{st} is the number of shortest paths between node s and node t that pass through node i, and g_{st} is defined as the total number of shortest s-t paths. (7) By running an SEIR simulation on the graph while removing a number of nodes, specifically up to 5, with the highest betweenness centrality, we can model a particular method of disease control. Betweenness centrality for a given node measures the number of shortest paths between nodes that the given node lies on. We hypothesize that removing the nodes with the highest betweenness centrality would effectively "disconnect" the graph on its shortest path between multiple nodes and this would stem the spread of the infection. In other words, we assume that an individual with high betweenness centrality may play a much more important role in indirectly passing pathogens than others in the population so removing them would cause our total number of infected individuals to decrease. Furthermore, this method also tests the effectiveness of betweenness centrality in quantifying a node's importance in indirectly connecting different nodes in the graph.

Eigenvector Centrality. Eigenvector centrality for a particular node i is defined by the following equation:

$$x_i = \kappa^{-1} \sum_{j=1}^{n} A_{ij} x_j$$

where κ is a proportionality constant, A_{ij} is the weight of the edge between node i and j, and x is the leading right eigenvector of A.

By running an SEIR simulation on the graph with removal of the node with highest eigenvector centrality, we can model yet another method of disease control. Eigenvector centralities for a given node measures the centrality of the node given the centralities of its neighbors. This method isolates the individual with the highest number of interactions by considering not only the number of direct interactions they have but also by the number of direct interactions their contacts have. That is to say by isolating this individual we assume they may play a more important role in passing pathogens both directly and indirectly through secondary contact. Since eigenvector centrality quantifies the influence of a node on a network, we hypothesize that isolating nodes with high eigenvector centrality will reduce the effectiveness of the spread since the nodes with access to the highest number of other nodes is no longer present. Furthermore, evaluating the results of removing these nodes can help to test the effectiveness of eigenvector centrality in quantifying said "influence".

Basic Reproduction Number

In order to quantify the results of the SEIR simulations used to test various disease control methods, observe the basic reproduction number, R_0. The basic reproduction number is the expected number of secondary cases directly generated by one case in a population where all individuals are susceptible to infection. It is a dimensionless quantity that gives insight into the infectiousness of a given disease. If $R_0 < 1$, then the virus cannot spread throughout the population. If $R_0 > 1$, then the virus will be able to spread throughout the population. Note that R_0 is not a constant value for any given viral disease; its value varies depending on other factors as well, including environmental conditions and behavior of the affected population. Calculation of R_0 for the base simulation will give a benchmark value for the disease modeled. Comparison of this value to the R_0 values for the simulations of each disease control method will aid in evaluating the effectiveness of these methods.

Estimation of R_0. There are a number of ways to estimate the value of R_0, based on different mathematical models of disease spread. In deciding which would be appropriate for the conditions of our model, we also considered the code-level limitations in retrieving data needed to make the calculations. It was decided to use the following estimation, which is derived from the SIR model:

$$R_0 = -\frac{\ln(1-f)}{f}$$

where f is the total fraction of the population that was infected at any point during the outbreak (8). This is equal to the fraction of recovered individuals at the end of the simulation. This equality holds true so long as the fraction of infected individuals at the end is zero, which we found to be the case for all of our simulations. It is important to note that $R_0 \in [1, \infty]$ as $f \in [0, 1]$, so in this case there is the assumption that each infected individual is on average already spreading the virus at least one person, which might not be the case all of the simulations. As previously mentioned, this estimation is based on the SIR model whereas our simulations are based on an SEIR model, therefore there is no consideration of the exposed period within this calculation. However note that within the simulation nodes which are in the exposed state are still infectious, which is not typical of a normal SEIR model, so we hypothesize that this estimation will still be accurate.

Comparison of R_0. While the estimation used for R_0 values might lead to assumptions not representative of our model, we can still quantify the success of the different disease control methods through comparison of these values. The calculations of R_0 for the base case, as well the models with quarantining and node removal, are shown in the table below.

Results and Conclusions

The following are graphs that resulted from various SEIR simulations under the different circumstances listed above. First, a regular SEIR simulation with no changes in our generated dataset is run and the results of the simulation are used as a base case to compare against the simulations with changes occurring in them. SEIR simulations on the five largest partitions that we obtain through Louvain's Algorithm are run thereafter. Next, we ran SEIR simulations on the dataset, first after removing the five nodes with the highest eigenvector centralities, then after removing the five nodes with the highest betweenness centralities, and lastly after removing both sets of nodes. Finally, an SEIR simulation is run on the dataset considering quarantine procedures that would be implemented after the discovery of an infection.

Basic Reproduction Values		
Method of Disease Control	f	R_0
Base	0.957	3.295
Quarantine Parameter	0.676	1.667
Removal of top 5 BC nodes	0.936	2.937
Removal of top 5 EC nodes	0.919	2.735
Removal of both centralities	0.917	2.713
Removal of bottom 5 BC nodes	0.955	3.247
Removal of bottom 5 EC nodes	0.958	3.309
Removal of bottom centralities	0.953	3.208

Table 1. Table of Reproduction Values Using Each of the Simulation Methods Described

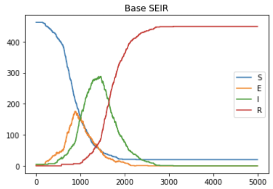


Fig. 3. Graph of a simulation of the SEIR model over 5,000 time-steps with 469 people interacting over the observed period.

A. Base Case. The figure details the SEIR model simulation before any changes were made. Note that the simulation begins with a population that is mostly susceptible, however as the number of exposed people increase, the number of susceptible people decrease and the number of infected individuals increase alongside it. However, at around two-thirds the height of the curve of the infected individuals, the number of exposed people decrease while the number infected continues to increase and the number of recovered people starts to decrease all while the number of susceptible people increase. At the height of infection, the number of susceptible, exposed, and recovered are around the same, and as the susceptible and exposed population decreases while the infected population begins to decline, we see how the recovered population starts to increase almost exponentially. This occurs until we reach a point where both S, E, and I groups are steadily diminishing while the R group consists of almost the entire population. At this time, not much more noticeable change occurs for the rest of the observable time-frame.

B. Centrality. As can be seen in 4 and 1, we saw a small decrease in the basic reproduction number associated with removing the nodes with the highest centrality. We did not continue removing more nodes from the graph because at a certain point, the decrease in R_0 can also be attributed to a smaller network size which will distort the results. However, we did find that our initial hypothesis that the betweenness centrality removal would be less useful than the eigenvector centrality removal was contrary to what the data revealed.

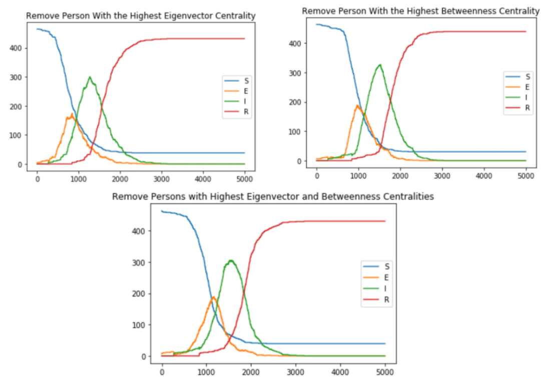
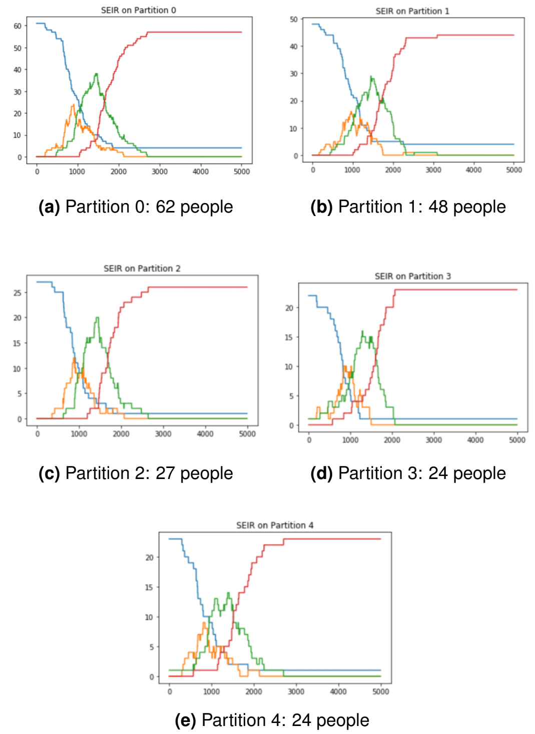


Fig. 4. Graphs of simulations of the SEIR model after removing the people with the highest eigenvector and betweenness centralities.

Eigenvector centrality was actually more useful in reducing the spread of the virus. We attribute this to the fact that betweennness centrality does not disconnect two groups from contact but instead only disconnects one possible shortest path in the graph and this may not affect the overall connectivity of the graph as opposed to removing nodes that has a high influence on many of the nodes in the graph. Overall, we did prove that the centrality measures were effective in quantifying a node's importance because there is a statistically significant correlation between the decrease of R_0 and the removal of centrality nodes that was not exhibited when removing the nodes with least centrality measures. However, the overall plots of S, E, I, and R of the simulation with the detailed mutations introduced did not make a noticeable impact.

C. Partitioning. Comparing the simulations run over the partitions with the simulation from the Base Case, it can be seen that the distribution of S, E, I, and R over time is relatively similar to that of the Base Case. This result shows that if we separated each community while each community individually has their own disease spread, it would spread out in a similar manner throughout each community separately just as it would if the communities were connected in the base case. We applied individual simulations but did not see a statistically significant difference between the R_0 values of large communities versus smaller communities. Furthermore, we tried removing entire communities from the simulation but we couldn't differentiate the results from the effect of removing a large part of the entire graph from consideration from the effect of removing a cohesive community. However, we did determine that each individual community operates like a microcosm of the larger ecosystem.

D. Quarantine. Finally, we found that implementing quarantine procedures upon discovery of infection provided significantly different results. The overall percentage of people infected was substantially lower than that of all the other simulations, with about one third of the population remaining healthy as compared to what would usually be about one tenth of the population. From this result, as well as the result from the analysis of the R_0 values, we find that a quarantine procedure proves to be the most effective strategy for reducing the spread of disease as opposed to other strategies such as partitioning communities and removing nodes with the highest centralities. However, we do acknowledge that our method of quarantine might not be extremely viable on a large scale. For starters, quarantining can still allow certain levels of infection since quarantining is not absolute in most countries. Secondly, having a person quarantine by himself as soon as the infection is discovered is also not a realistic approach. However, knowing that we are making certain non-realistic assumptions, we still believe that our measurements are statistically significant enough to conclude that quarantining is a very effective measure against spread of diseases that transfer through contact.



(a) Partition 0: 62 people

(b) Partition 1: 48 people

(c) Partition 2: 27 people

(d) Partition 3: 24 people

(e) Partition 4: 24 people

Fig. 5. Graphs of simulations of the SEIR model after partitioning the communities within the network with Louvain's Algorithm.

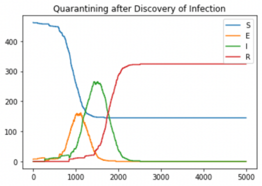


Fig. 6. Graph of a simulation of the SEIR model taking quarantine measures into consideration.

ACKNOWLEDGMENTS. We would like to thank Professor Mason Porter and TA Abby Hickock for their valuable insight and advice throughout this project.

1. S Kissler, P Klepac, M Tang, A Conlan, J Gog, Sparking "the bbc four pandemic": Leveraging citizen science and mobile phones to model the spread of disease (2018).
2. S Kissler, P Klepac, M Tang, A Conlan, J Gog, Supplemental information and data for the manuscript sparking "the bbc four pandemic": Leveraging citizen science and mobile phones to model the spread of disease (https://github.com/skissler/haslemere) (2020).
3. L Gauvin, et al., Randomized reference models for temporal networks (2020).
4. J Gopalakrishnan, The seir model of infectious diseases (2020).
5. V Giallonardo, et al., The impact of quarantine and physical distancing following covid-19 on

mental health: Study protocol of a multicentric italian population trial. *Front. Psychiatry* **11**, 533 (2020).

6. VD Blondel, JL Guillaume, R Lambiotte, E Lefebvre, Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008).
7. MEJ Newman, *Networks: an introduction.* (Oxford University Press, Oxford; New York), (2010).
8. MJ Keeling, P Rohani, *Modeling Infectious Diseases in Humans and Animals.* (Princeton University Press), (2011).