Simulating Influenza Transmission with Network Data

By Henry Bongiovi

Background

Influenza has been studied extensively throughout history and its behavior and symptoms are well documented. Influenza spreads rapidly through moisture droplets during close proximity interactions, (Salathe et al., 2010). However, the rate of its spread through different environments is still difficult to track because not everyone reports having influenza and many people may not even know they are carrying it, thus it is difficult to determine when there is a risk of a true influenza epidemic. Models using network data are a relatively new way of simulating such circumstances without having to observe actual epidemics. In this case the network data is a network of face-to-face contact between two people for an extended period of time. In this sense network data is a perfect way to simulate the spread of influenza and other influenza like diseases. Schools are one of the places influenza is thought to spread very quickly. A study done by Salalthe has shown that schools have a high density contact network, meaning people in schools tend to have long durations of face-to-face contact with each other very often. This is a perfect environment for influenza to spread. Thus, it is important to understand when a school is in danger of exposing its students to pandemic. Using data based on the face-to-face contact of a school's students with each other and the staff, a more accurate simulation can be made to estimate the rate of transmission as well as the effectiveness of intervention strategies implemented during the outbreak.

Review of Related Study

This study is closely related to a study conducted by Potter, Handcock, Longini, and Halloran (2011), so we summarize that study in this section. Potter et al. compared the effect of epidemic simulation models using a social contact network estimated from data vs. a simulation models based on a random mixing network where contact with other people is simulated to occur with equal probability between all people.

The intent of this study is to better understand the transmission of influenza through contact networks, specifically using an ERGM (exponential random graph model). To do so, an ERGM of friends was compared to a conventional model of a contact network which assumes random mixing of all those in the network. Mathematically, ERGM models the sociomatrix (a matrix used to measure interpersonal relationships) for a network of fixed size as follows:

$$P(Y = y|\theta) = \frac{e^{\sum_{i=1}^{k} \theta^T g(y)}}{k(\gamma)}$$

Here, γ denotes the space of all possible networks of this size and k(γ) is a normalizing constant which ensures that the probability distribution sums to 1. Θ is a vector of parameters and g(y) is a vector of network statistics, such as the number of edges. However, ERGMs are out of the scope of this project. Potter et al. (2011) provides further insight and more in-depth descriptions of the mathematical properties of ERGM.

The study takes place in a school setting, derived from data received from 2 studies. The first being the Add Health study conducted by Harris (2009), a survey of health, demographic, and relational data from 80 high school/feeder schools in America. Grades ranged from 7th-12th grade during the 2009 school year. The second study was "A survey on Epidemics in High Schools," a contact survey

administered in two Virginia high schools in spring 2009 by Xia et al. (2010). In addition, the POLYMOD study [Potter et al (2011a), Mossong et al. (2008), Hens et al. (2009)] is also referenced as an alternative data source for modeling contact networks. The POLYMOD study was a diary based survey of contact behavior conducted in Europe.

The model created was based on the friendship data obtained in the Add Health network while using the epidemic survey to estimate the number of contacts, and preference of contacts to friends. The model captures a tendency to make contact, especially longer contact, with close friends, while also contacting other students, albeit, less frequently and for shorter periods of time throughout the day.

Each student is represented as a node and each contact as an edge. Contact is defined by face-to-face contact for ten minutes. For instance, if a student is in contact with another student for an hour, that is considered six contacts. A maximum of 38 contacts per any two students is permitted as a typical school day does not exceed 6 hours and 20 minutes.

Three different versions of the contact network model were compared. In the dynamic contact network model, students keep the same class contacts for the duration of the influenza season but a new break/lunch network is sampled daily. Within class contacts were modeled by assuming students take classes only within their own grade. Each student is then randomly assigned 2, 3, or 4 class neighbors with probabilities 1/9, 4/9, and 4/9 in each class. They then assumed that students have 40 minutes of continual contact with these neighbors while is in session. Lunch/break networks were modeled differently in that they estimated a cutoff of 30 average lunch contacts that range from 10, 20, 30, 40, or 50 minutes with all equal probability. In the static model, students contact the same people each day for the duration of the influenza season. In the friendship-only model, students only contact their friends. These models are calibrated so that the expected total number of contacts in all models is the same. The authors also considered a dynamic contact model, in which new break and lunch contact networks are simulated each day, but the classroom contact network remains fixed through the epidemic.

The authors simulated five hundred influenza epidemics over each of the three models. Essentially, when an infected student contacts a healthy one, the infected student may transmit influenza with a certain transmission probability, which increases with the duration of contact between them. The specification for the epidemic simulation is very similar to the one I preform in this project, and will be detailed in the methods section. The authors compared epidemic outcomes between the three models. The dynamic and static models were very similar because they were based on the same friendship network, and the added variety during the breaks/lunch seemed to have little effect on the dynamic model because contacts made where usually 10 minutes or less. However, the friendship only model appeared to be oversimplified as the expected peak date, final size, probability of epidemic and expected peak date by final size all appeared to underestimate the epidemic when compared to the other models. Therefore the static model was chosen for the final model.

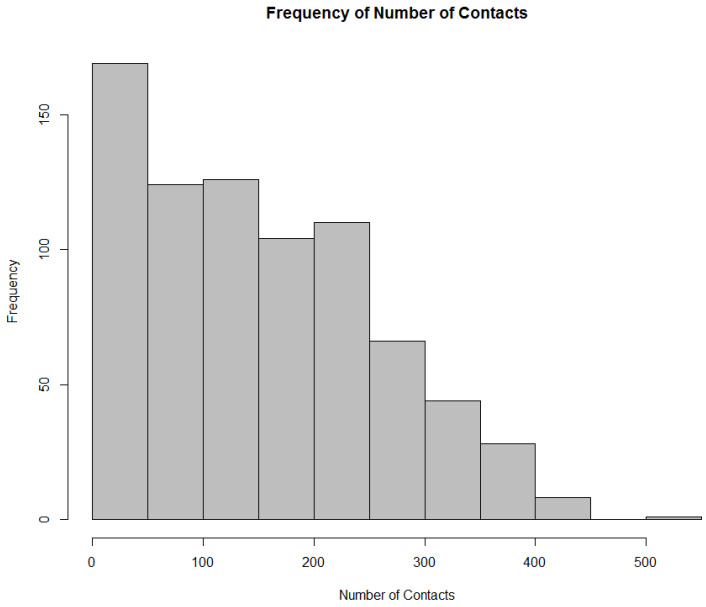
Next the simulated network model was compared to the empirical friendship network and the results were almost identical. This means that the current network model captures the structures relevant to disease transmission. Now this contact network model is compared to random mixing model. The random mixing model was calibrated to the same expected number of people contacted per student per day as well as duration as in the friendship model. First the models were simulated without intervention, then with a grade closure intervention. Finally a targeted antiviral prophylaxis intervention was tested on both models.

The results showed that the contact network estimated a smaller probability of epidemic and smaller final size than the random mixing model. This is due to the nature of friends infecting only other friends and the higher probability of mutual friends between friendships. Their research indicates that this network model is more realistic than compared to the random mixing model. However, they also

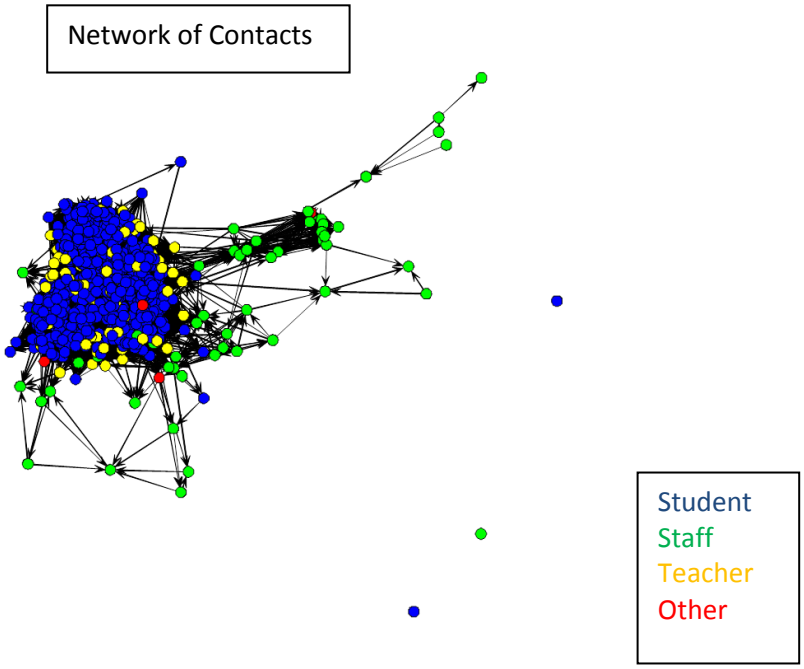
found that a static model sufficiently models epidemics in a high-school setting. However, in this model neither contact between students nor their duration were simulated. All the data used in the simulation were collected at an actual high school. Although these simulations are based only on one high school, using real network data should produce results that truly reflect the given population.

Data

The data used for this simulation was obtained from a paper study done by Salathe, Kazandjieva, Lee, Levis, Feldman, and Jones (2010). The study consisted of using RFID (Radio Frequency Identification) chips to collect data on social networks at schools, in order to simulate pandemics at schools. Their study covered 94% of a typical high-school that included 788 individuals. Of these individuals, 655 were students, 73 were teachers, 55 were staff and 5 were classified as other. The data was collected with RFID chips that recorded contact with another chip when they were within 3 meters of each other. The RFID chip would then record a contact approximately every 20 seconds they remained within 3 meters. The data itself consisted of 762,868 interactions with a mean duration of 2.8 contacts (~56 seconds). It should be noted that some of the measurements for contact durations were inconsistent. For instance, the device may have recorded the duration of a contact differently for each person in the contact. The data collection period was one day.



Above is a distribution of the number of contacts for each subject in the study. As shown, most people made between 50 and 250 contacts on the day of the study. One person made over 500 contacts on this day. It should be also noted that the distribution of contacts is not normal. The distribution resembles a right skewed distribution such as an exponential or Poisson. How this distribution could affect the epidemic might be useful in future studies.



Above is a visual representation of the network data from Salathe. However, it is of note to mention that this visualization does not include the duration of contacts between individuals. It only shows that a contact exists between individuals. As shown there are several individuals who are not connected to anyone and several others who appear to make contact with only a few people. This could affect results, especially if only a small number of simulations are being run.

Method

Before I could develop a simulation, I obtained the data from the Salathe study, converted it into a sociomatirx, and preformed descriptive analysis. The data comprised a list of all individual interactions in a single day, regardless of previous encounters with the same people. I condensed the list so that multiple interactions with the same people, in the same order, were combined. In addition the duration were recorded in 20 second intervals. In order to keep consistent to the Potter study, I simplified the contact durations to number of contacts per 10-minutes. The directionality in the data is due to measurement error. A future study addressing the effectiveness of the data collection method would be useful to obtain more accurate data on contacts and contact duration. There is no scientific reason to include the directionality. However, because the data was collected in this manner, keeping order of infector versus infected was the simplest approach to using the data in the simulation. This infector would have the potential to spread the virus the infected. This fact plays an important role in the calculation of probability of infection. The idea is to calculate the probability of becoming infected for an entire school day and repeating the process day-to-day until there is no one left to infect or everyone has become immune. The day-to-day probability calculation was based off of the same equation used in Potter et al. (2011). We let $p_{t,i}$ denote the per 10 minute transmission probability of person *i* on day *t*. The events that *i* transmits to *j* during two different 10-minute contacts are dependent since transmission during the earlier contact precludes transmission during the later. In addition $Y_{i,j}$ is the duration of contact duration between person *i* and person *j* Thus if *j* is susceptible,

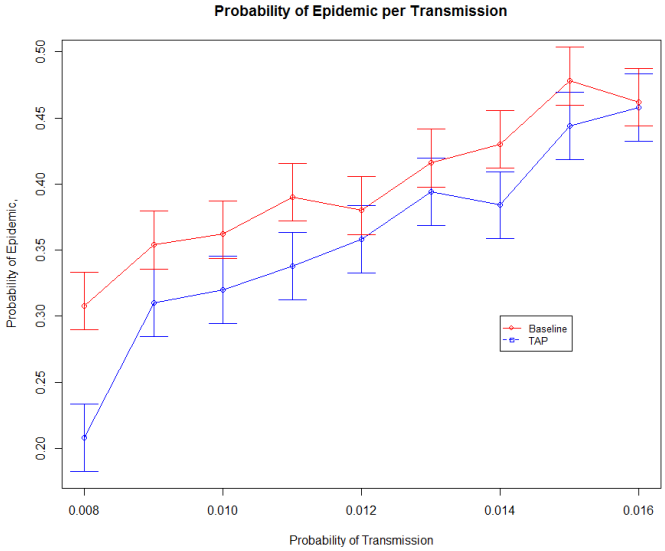
P(*j* escapes infection by person *i* on day *t*) = $(1\text{-}p_{t,i})^{Yij}$
so that, P(*j* infected on day *t*) = $1 - \prod_{i=1}^{n}(1 - p_{t,i})^{Y_{ij}}$

In addition, I incorporated several assumptions about the human nature of influenza outbreaks that affect transmission. These assumptions have been researched by Chao, Halloran and Longini (2010) and Elveback et al. (1976). The first assumption is that everyone has an incubation period of 1, 2, or 3 days with the respective probabilities 0.30, 0.50, and 0.20, and that everyone is infected for 6 days, after which they become immune to this particular strain of influenza. During incubation, subjects are infectious but not symptomatic. From Murphy et al. (1980) and Baccan et al. (2006) we also assume that around 67% of the students will become symptomatic after incubation, meaning they will become twice as infectious. However, this leaves 33% who will remain infectious at their standard rate without showing symptoms. After becoming symptomatic 75% of these students will withdraw to their homes. This 75% is further broken up in that 20.3% will retreat home on the first day symptoms appear, 39.7% on the second, and 15% on the third. These subjects will no longer be able to influence the school population.

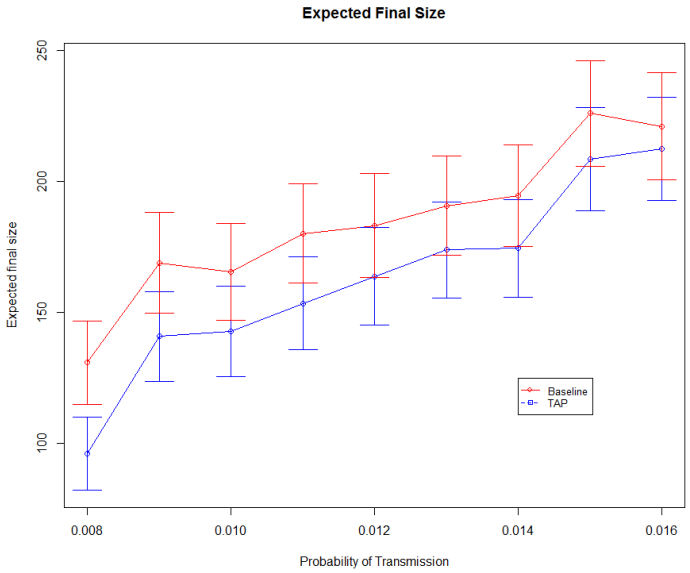
After the assumptions have been accounted for I simulated a range of infection probabilities in order to determine at which infectious rate epidemics will become likely. From the Potter study (2011), I based these ranges on 100 simulations per transmission probability and found that the probability of infection from 0.005 - 0.020 showed the most drastic changes. After settling on this range, I decided to further hone in on the subset 0.008 – 0.016 using a larger simulation of 500 epidemics per transmission probability. Every simulation starts with one infected individual, and there are times when an individual with a low number of contacts is selected. When this occurs, epidemic rates tend to be lower than expected and increase the variability of the simulated samples overall. This is the reason I increased the simulations to 500.

Next, I applied an intervention to the simulation that is based off of the *targeted antiviral prophylaxis* studied by Halloran et al. (2007). Under this model all those who are symptomatic are given antiviral treatment the day after symptoms occur. These treatments are given for 5 days. In addition, all those this person has had contact with during that day are given antiviral prophylaxis for 10 days. This intervention has a drastic effect on infectivity. While on antiviral prophylaxis, a subject is .44 times less likely to become symptomatic and the probability of becoming infected is reduced by a factor of .37. If the infectious person is on antiviral treatment (not prophylaxis) the chance of becoming infected is reduced by an additional factor of .85. After simulations had run on both models I graphed them and compared the results.

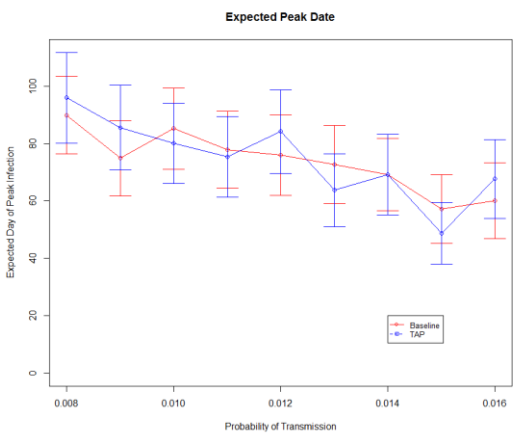
Results

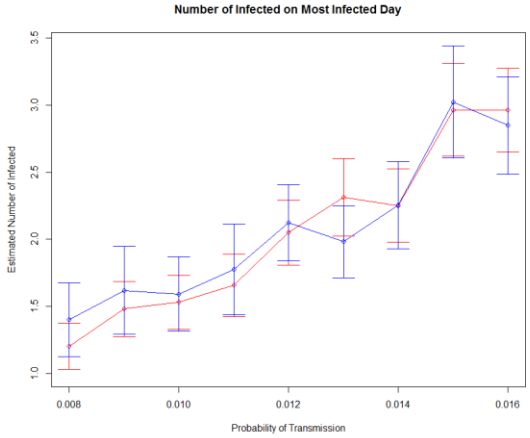


After running 500 simulations per probability of transmission (0.008-0.016) I graphed the probability that the outbreak of influenza reached an epdidemic, 200 or more people infected. When the probability of transmission is .013 the probability of an epidemic is around 40%. As show, when a TAP intervention is effect the probability of epidemic is around 37%. However as noted above, the error bars indicate that for mid range values there is some overlap. As is the case at .013 in this case we cannot say that TAP is significantly more effective. Probability of transmissions above .016 tended to increase very slowly, the most drastic change seemed to be within this range. However, TAP maintained a lower average probability of epidemic by around 8% throughout with error bars slightly overlapping. Next I graphed the expected final size of the epidemic per transmission probability. As seen, the graph is similar to the previous in that TAP maintains a difference of about 40 people. However, the overalp in error bars here is more evidient throughout, again showing that TAP is not necessarily more effective here.

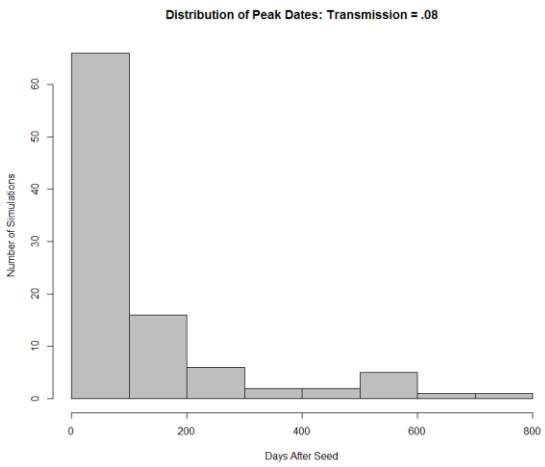


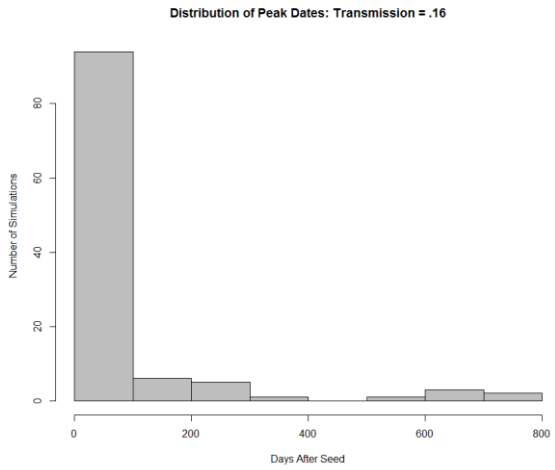
The most counter intuitive graph is the graph of expected day of peak infection versus the probability of transmission. Although it seems that TAP is preforming worse than no treatment, in actuality this is not true. If we compare this graph to a graph that shows the number of infected on these peak days we see that for every given peak day TAP has less infected. For the graphs below, simulations where the epidemic did not spread beyond the seed were omitted.





Given that the range of transmission probabilities gave expected final epidemics of less than 300 people it is not unreasonable to see that the expected peak dates of infection only infected 1.5 to 4 people a day. This is especially evident given the length of epidemics were typically dragged on by the fact that the epidemic could be carried on by a few lingering people who would then infect another person late in their infectivity. This chain of one-to-one infections sometimes lasted a long time and gave simulations that lasted up to 800 days. From this we can infer that the peak day of infection is not necessarily related to the final size of the epidemic. However, this seems counter intuitive as typically we would expect to see earlier peak dates with smaller epidemics. To investigate this further I graphed the distribution of peak infection days by transmission probabilities 0.08 and 0.16.





From the above graphs we can see that even after excluding peak dates of 0 (indicating no spread) both distributions are heavily right skewed. However we can see that the number of extreme values for transmissions of 0.08 is higher than the number of extreme values for a transmission probability of 0.16. Therefore the means are being artificially inflated for transmissions with lower probabilities.

Discussion

At first I was skeptical of my results but the more I looked at the network data the more comfortable I felt with the results. In order to achieve epidemic sizes with comparable frequencies as the Potter study, the transmissions for the simulation needed to be much higher. However, I attribute this to the unique network of the single high school that the Salathe data was based on. Some of the students in school had very little contact with others that day and I suspect there may be some error in data collection. This put a heavy dependency on seed selection in order for the virus to thrive. In many simulations the seed did not succeed in transmitting the virus.I would be very interested to see more data of this type collected on more high schools. If data on more high school's contact networks were available I would like to see what the optimal range of probability transmissions are for epidemics, across all schools.

This study has just been a glimpse of what can be accurately modeled using real network data. Although this paper only touched on one intervention method, TAP, any other known method can be tested using similar simulations, assuming the contact network it is based on is reliable. From just this simulation we see that TAP is an effective method to quell epidemics of influenza. Understanding and simulating as realistic network data as possible will be key to developing new strategies to combat diseases without having to test them in real situations.

However, because this simulation is based only on contact data from one American high school, it is hard to tell weather similar results would occur elsewhere. That being said it would be interesting to get more studies like Salalthe's conducted across the nation. By collecting more data on real face-to-face contact networks we will be able to better generalize this information to a larger population.

That being stated, there are already some studies that apply similar methods on a larger, country-wide scale. One such has been conducted by Chao et al. (2010). By combing simulation methodology from this Chao's and real network data a realistic simulation of a country-wide epidemic is very possible. By collecting more data world-wide there is no reason an accurate simulation of a world-wide epidemic is not plausible.

This area of research has great potential in education as well. Knowing very little about the theory and mathematics around networks (graph theory) I was able to create a relatively accurate simulation of a real life epidemic. Simulations such as these are easy to understand and can taught to a wide audience. By showing a simulation such as this, people can easily see the factors that affect the spread and longevity of an easily prevented epidemic. Not only can this save immediate lives by testing which interventions are most effective, it can raise awareness of effective outbreak prevention lifestyles and habits.

References

Baccan, P., Beauchemin, C., Macken, C. A., Hayden, F. G. and Perelson, A. S. (2006) Kinetics of influenza A virus infection in humans. J.*Virol.* **80** 7590-599.

Chao, D. L., Halloran, M. E. and Longini, I. M., Jr. (2010). School opening dates predict pandemic influenza A(H1N1) outbreaks in the United States. *J. Infect. Diseases* **202** 877-880.

Elveback, L. R., Fox, J. P., Ackerman, E., Langworthy, A., Boyd, M. and Gate-Wood, L. (1976) An influenza simulation model for immunization studies. *Am. J. Epidemiol.* **103** 152-165.

Halloran, M. E., Hayden, F. G., Yang, Y., Longini, I. M., Jr. and Monto, A. S., (2007) Antiviral effects on influenza viral transmission and pathogenicity: Observations from house-hold based trials. *Am. J, Epidemiol.* **165** 212-221.

Harris, K. M. (2009). The national longitudinal study of adolescent health (add health), waves I and II, 1994-1996; wave III, 2001-2002; wave IV, 2007-2009. [Machine readable data file and documentation] Carolina Population Center, Univ. North Carolina at Chapel Hill.

Hens, N., Goeyvaerts, N., Aerts, M., Shkedy, Z., Damme, P. V. and Beutels, P. (2009) Mining social mining patterns for infectious disease models based on a two-day population survey in Belgium. *BMC Infect. Dis.* **9** 5.

Salathe, M., Kazandjieva, M., Lee, J. W., Levis, P., Feldman, M. W. and Jones, J. H.. (2010). A high-resolution human contact network for infectious disease transmission. *Proc. Natl. Acad. Sci. USA* **107** 2202022025.

Massong, J., Hens, N., Jit, M., Beutels, P. Auranen, K., Mikolajczyk, R., Massari, M., Salmaso, S., Tomba, G. S., Wallinga, J., Heijne, J., Sadkowska-Todys, M., Rosinska, M. and Edmunds, W. J. (2008) Social contacts and mixing patterns relavent to the spread of infectious disease. *PLoS Medicine* **5** 0381-0391.

Murphy, B. R., Rennels, M. B., Douglas, R. G., Jr., Betts, R. F., Couch, R. B., Cate, T. R., Jr., Chanock, R. M., Kendal, A. P., Maassab, H. F., Suwanagool, S., Sotman, S. B., Cisneros, L. A., Anthony, W. C., Nalin, D. R., and Levine, M. M. (1980). Evaluation of influenza A/Hong Kong/123/77 (H1N1) ts-1A2 and cold-adapted recombinant viruses in seronegative adult volunteers. *Infect. Immune.* **29** 348-355.

Potter, G. E., Handcock, M. S., Longini, I. M., Jr. (2012). Estimating within-school contact networks to Understand influenza transmission. *The annals of Applied Statistics Vol. 6 No. 1* 1-26.

Xia, H., Chen, J., Marathe, M. V. and Mortveit, H. S. (2010). Synthesis and embedding of subnetworks for Individual-based epidemic models. NDSSL Technical Report 10-469, Univ. North Carolina at Chapel Hill.

Appendix

I have decided to include the code used throughout the report. The code is written in R and is divided up into sections.

Data prep – This section imports the data from the Salathe study and creates the necessary variables, matrices, lists, and objects that the main functions use.

Day-to-Day Sim function – This section creates the function used to calculate the probability of infection per day for each individual. This function takes a matrix of transmission probabilities, a matrix of contact durations, and a dummy matrix of infected individual (denoted in binary).
The function returns a list containing an updated version of the given matrix of infected and a vector of newly infected individuals.

Epidemic Sim function – This function runs the Day-to-Day function until everyone has been infected or is all immune. It takes a matrix of transmission probabilities, a matrix of contact durations, a dummy matrix of dummy infected individuals (denoted in binary), a data frame that contains vital information on all individuals in the simulation, and an indicator (T or F) of whether or not to use TAP.
The function will return a list containing the number of days the epidemic lasted and an updated data frame that contains the final variables and status of each individual in the study.

Mode function – This function calculates the mode of a vector. This function takes a vector.
This function will return the mode of the given vector.

Simulation – This section of code is the code I used to run the simulation for each probability transmission 500 times.

Graphs – This section of code is the code I used to produce the graphs used in the report.

```
##############################
## Epidemic Simulation       ##
## Henry Bongiovi            ##
##############################



######### Data Prep ############

# adjacency matrix creation (contact matrix with durations)

library('statnet');
netTest=
read.table('http://www.pnas.org/content/suppl/2010/12/09/1009094108.DCSupplemental/sd02.txt' );
netSorted = sort(netTest);
netMatrix = as.matrix(netSorted);

sum = 0;
uniqueNet = cbind(0,0,0);
for(i in 2:nrow(netMatrix))
{
   if(netMatrix[i,1] == netMatrix[(i-1),1] && netMatrix[i,2] == netMatrix[(i-1),2])
   {
      sum = sum + netMatrix[(i-1), 3];
   }
```

```
   else
   {
      sum = sum + netMatrix[(i-1),3];
      idDur = cbind(netMatrix[(i-1),1], netMatrix[(i-1),2], sum);
      sum = 0;
      uniqueNet = rbind(uniqueNet, idDur);
   }
}
idDur = cbind(netMatrix[i,1], netMatrix[i,2], sum);
uniqueNet=rbind(uniqueNet, idDur);
uniqueNet = uniqueNet[-1,];
net = network(uniqueNet, matrix.type = 'edgelist');
set.edge.attribute(net,'weight',uniqueNet[,3]);
adjMat = as.matrix(net,matrix.type='adjacency',attrname='weight');
adjMat=adjMat/30;

# event Matrix creation

seed = sample(1:nrow(adjMat),1);

eventMat = matrix(data=0,nrow=nrow(adjMat), ncol=ncol(adjMat));
eventMat[seed,] = 1;

# transmission Matrix
#original data used .001-.007 where .001-.002 were too small to start an epidemic.
idTable =
read.table('http://www.pnas.org/content/suppl/2010/12/09/1009094108.DCSupplemental/sd03.txt');
transMat = matrix(data=.004,nrow=nrow(adjMat), ncol=ncol(adjMat));
diag(transMat) = 0;

# create the Person object

createPerson=function(adjMat,seed)
{
   incubationTimes = NULL;
   symptomatic = NULL;
   withdrawl = NULL;
   for(i in 1:nrow(adjMat))
   {
      incubationTimes[i] = sample(x=(1:3), size=1, prob=c(.3,.5,.2));
      symptomatic[i] = sample(x=c(T,F), size=1, prob=c(.67,.33));
      if(symptomatic[i])
      {
         withdrawl[i] = sample(x=c(-999,1,2,3), size=1, prob=c(.25,.203,.397,.15));
      }
      else
      {
         withdrawl[i] = -999;
      }
   }
   dayInfected = NULL;
   treatment = NULL;
   prophylaxis = NULL;
   treatmentDay=NULL;
   prophylaxisDay=NULL;
   for(i in 1:nrow(adjMat))
   {
      dayInfected[i] = 0;
      treatment[i]=F;
      prophylaxis[i]=F
      treatmentDay[i]=5;
      prophylaxisDay[i]=10;
   }
   infected=vector(length=nrow(adjMat));
   immune = vector(length=nrow(adjMat));
   person = data.frame(infected, dayInfected, immune, incubationTimes, symptomatic, withdrawl,
treatment,
                       treatmentDay, prophylaxis, prophylaxisDay);
   person$infected[seed] = T;
   return(person);
}
```

```
######### Day-to-Day SIM function ############

   # Will run through one day of infections and update the contactMat of who is infected.
   # will return the new updated eventMat

infectionSim = function(transMat, adjMat, eventMat)
{
   infection = 1-apply((1-transMat)^(adjMat*eventMat), 2, prod);
#    print(infection);
   new=NULL;
   for(i in 1:length(infection))
   {
      new[i] = sample(x=c(T,F), size=1, prob=c(infection[i], 1-infection[i]));
   }
   if(length(which(new)) != 0)
   {
      for(i in 1:length(which(new)))
      {
         eventMat[which(new)[i],] = 1;
      }
   }
   else
   {
      new =NULL;
   }
   eventList = list(eventMat, new);
   return(eventList);
}


######## Epidemic SIM function ##############

   # Will run through an entire epidemic, simulating unil all have been infected.
   # prints the number of days the infection lasted.

epidemicSim = function(transMat, adjMat, eventMat, person, intervention)
{

   days=1;
   tempList = NULL;

   while(length(unique(person$infected)) != 1)
   {
      tempList = infectionSim(transMat, adjMat, eventMat);
      eventMat = tempList[[1]];
      if(is.vector(tempList[[2]]) == T)
      {
         for(i in 1:length(which(tempList[[2]])))
         {
            if(person$infected[which(tempList[[2]])[i]] == F &&
person$immune[which(tempList[[2]])[i]] == F)
            {
               person$infected[which(tempList[[2]])[i]] = T;
               person$dayInfected[which(tempList[[2]])[i]] = days;
               eventMat[which(tempList[[2]])[i],] = 0;
            }

         }
      }

      for(i in 1:nrow(adjMat))
         if(intervention)
         {
            if(person$prophylaxis[i])
            {
               person$prophylaxisDay[i] = person$prophylaxisDay[i]-1;
               if(person$prophylaxisDay[i] ==0)
               {
                  transMat[i,] = transMat[i,]/.37;
               }
```

```
            }
            if(person$treatment[i])
            {
               if(days == person$dayInfected[i] && person$symptomatic[i] == T)
               {
                  person$symptomatic[i] = sample(x=c(T,F), size=1, prob=c(.56,.44));
               }
               person$treatmentDay[i]=person$treatmentDay[i]-1;
               if(person$treatmentDay[i]==0)
               {
                  transMat[i,]=transMat[i,]/.85
               }
            }
         }
         if(person$infected[i])
         {
            if((days - person$dayInfected[i]) == 6)
            {
               person$immune[i] = T;
               eventMat[i,] = 0;
               person$infected[i] = F;
            }
            person$incubationTimes[i] = person$incubationTimes[i] -1;
            person$withdrawl[i] = person$withdrawl[i]-1;
            if(person$withdrawl[i] == 0)
            {
               eventMat[i,]=0;
            }

            if(days - person$dayInfected[i] == 1 )
            {
               eventMat[i,]=1;
            }
            if(person$incubationTimes[i] == 0)
            {
               if(person$symptomatic[i])
               {
                  transMat[i,] = transMat[i,] * 2;
               }
            }
            if(intervention)
            {
               if(person$symptomatic[i] == T && person$incubationTimes[i] == -1)
               {
                  person$treatment[i] = T;
                  transMat[i,] = transMat[i,] * .85;
                  for(j in 1:length(which(adjMat[i,] > 0)))
                  {
                     vec=which(adjMat[i,] > 0);
                     if(length(vec) > 1)
                     {
                        if(is.vector(vec[j]))
                        {
                           if(person$prophylaxis[vec[j]] == F)
                           {
                              person$prophylaxis[vec[j]] = T;
                              person$prophylaxisDay[vec[j]] = 10;
                              transMat[vec[j],] = transMat[vec[j],]*.37;
                           }
                        }
                     }
                  }
               }
            }

         }
      }

      days= days+1;
   }
#    print(days);
```

```
#    print (person);
   results = list(person, days);
   return(results);
}

################# Mode function ########################
Mode = function(x)
{
   ux = unique(x)
   ux[which.max(tabulate(match(x, ux)))]
}

#########################################################
############## Simulation ######################
temp=NULL;
tempAvgDays=NULL;
tempMode = NULL;
tempSum =NULL;
tempFreq=NULL;


tempT=NULL;
tempAvgDaysT=NULL;
tempModeT = NULL;
tempSumT =NULL;
tempFreqT=NULL;
avgDaysSD = NULL;
avgDaysTDS= NULL;
totalInfectedSD= NULL;
totalInfectedTSD= NULL;
modeSD= NULL;
modeTSD= NULL;
freqSD= NULL;
freqTSD= NULL;


for(i in 8:16)
{
   print(i);
   yeild = c('base', i*.001);
   yeild[2] = substr(yeild[2],2,5);
   yeild = paste(yeild, collapse='');
   avgD = c('avgDays', i*.001);
   avgD[2] = substr(avgD[2],2,5);
   avgD = paste(avgD, collapse='');
   modeD = c('modeDays', i*.001);
   modeD[2] = substr(modeD[2],2,5);
   modeD = paste(modeD, collapse='');
   totalInf = c('totalInfected', i*.001);
   totalInf[2] = substr(totalInf[2],2,5);
   totalInf = paste(totalInf, collapse='');
   freqD=c('freq',i*.001);
   freqD[2]=substr(freqD[2],2,5);
   freqD=paste(freqD,collapse='');
   probD=c('probEp', i*.001);
   probD[2]=substr(probD[2],2,5);
   probD=paste(probD,collapse='');

   yeildT = c('baseTAP', i*.001);
   yeildT[2] = substr(yeildT[2],2,5);
   yeildT = paste(yeildT, collapse='');
   avgDT = c('avgDaysTAP', i*.001);
   avgDT[2] = substr(avgDT[2],2,5);
   avgDT = paste(avgDT, collapse='');
   modeDT = c('modeDaysTAP', i*.001);
   modeDT[2] = substr(modeDT[2],2,5);
   modeDT = paste(modeDT, collapse='');
   totalInfT = c('totalInfectedTAP', i*.001);
   totalInfT[2] = substr(totalInfT[2],2,5);
   totalInfT = paste(totalInfT, collapse='');
   freqDT=c('freqTAP',i*.001);
```

```
   freqDT[2]=substr(freqDT[2],2,5);
   freqDT=paste(freqDT,collapse='');
   probDT=c('probEpT', i*.001);
   probDT[2]=substr(probDT[2],2,5);
   probDT=paste(probDT,collapse='');

   for(j in 1:500)
   {

      transMat=matrix(data=(i*.001),nrow=nrow(adjMat), ncol=ncol(adjMat));
      diag(transMat) = 0;
      seed = sample(1:nrow(adjMat),1);
      person = createPerson(adjMat,seed);
      eventMat = matrix(data=0,nrow=nrow(adjMat), ncol=ncol(adjMat));
      eventMat[seed,] = 1;

      temp[[j]] = epidemicSim(transMat, adjMat, eventMat, person, intervention=F);
      tempAvgDays[j] = temp[[j]][[2]];
      tempSum[j] = sum(temp[[j]][[1]]$immune);
      tempInfected = sum(temp[[j]][[1]]$infected);
      tempMode[j] = Mode(which(temp[[j]][[1]]$dayInfected > 0));
      tempFreq[j] = sum(temp[[j]][[1]]$dayInfected == tempMode[j]);

      tempT[[j]] = epidemicSim(transMat, adjMat, eventMat, person, intervention=T);
      tempAvgDaysT[j] = tempT[[j]][[2]];
      tempSumT[j] = sum(tempT[[j]][[1]]$immune);
      tempModeT[j] = Mode(which(tempT[[j]][[1]]$dayInfected > 0));
      tempFreqT[j] = sum(tempT[[j]][[1]]$dayInfected == tempMode[j]);
   }
   assign(yeild, temp);
   assign(avgD, mean(tempAvgDays));
   assign(totalInf, mean(tempSum, na.rm=T));
   assign(modeD, mean(tempMode, na.rm=T));
   assign(freqD, mean(tempFreq, na.rm=T));
   assign(probD, sum(tempSum >199)/500);

   assign(yeildT, tempT);
   assign(avgDT, mean(tempAvgDaysT));
   assign(totalInfT, mean(tempSumT, na.rm=T));
   assign(modeDT, mean(tempModeT, na.rm=T));
   assign(freqDT, mean(tempFreqT, na.rm=T));
   assign(probDT, sum(tempSumT >199)/500);

}

avgDaysSD[i-7] = sd(tempAvgDays, na.rm=T);
avgDaysTDS[i-7] = sd(tempAvgDaysT, na.rm=T);
totalInfectedSD[i-7] = sd(tempSum, na.rm=T);
totalInfectedTSD[i-7] = sd(tempSumT, na.rm=T);
modeSD[i-7] = sd(tempMode, na.rm=T);
modeTSD[i-7] = sd(tempModeT, na.rm=T);
freqSD[i-7] = sd(tempFreq, na.rm=T);
freqTSD[i-7] = sd(tempFreqT, na.rm=T);

avgDays = c(avgDays.008,avgDays.009,avgDays.01,
            avgDays.011,avgDays.012,avgDays.013,avgDays.014,avgDays.015,avgDays.016);
            ,avgDays.037,avgDays.038,

avgDays.039,avgDays.04,avgDays.041,avgDays.042,avgDays.043,avgDays.044,avgDays.045,avgDays.046,
            avgDays.047,avgDays.048,avgDays.049,avgDays.05);

sumAll=c(totalInfected.008,totalInfected.009,totalInfected.01,totalInfected.011,totalInfected.012
,
         totalInfected.013,totalInfected.014,totalInfected.015,totalInfected.016);
         ,totalInfected.062,

totalInfected.063,totalInfected.064,totalInfected.065,totalInfected.066,totalInfected.067,

totalInfected.068,totalInfected.069,totalInfected.07,totalInfected.071,totalInfected.072,
         totalInfected.073,totalInfected.074,totalInfected.075);
```

```
modeAll=c(modeDays.008,modeDays.009,modeDays.01,modeDays.011,
          modeDays.012,modeDays.013,modeDays.014,modeDays.015,modeDays.016);
         ,modeDays.037,modeDays.038,

modeDays.039,modeDays.04,modeDays.041,modeDays.042,modeDays.043,modeDays.044,modeDays.045,
          modeDays.046,modeDays.047,modeDays.048,modeDays.049,modeDays.05);
probEpi=c(probEp.008,probEp.009,probEp.01,probEp.011,probEp.012,probEp.013,probEp.014,probEp.015,
probEp.016);
         ,probEp.017,probEp.018,
          probEp.019,probEp.02,probEp.021,probEp.022,probEp.023,probEp.024,probEp.025);
freqAll=c(freq.008,freq.009,freq.01,freq.011,freq.012,freq.013,freq.014,freq.015,freq.016);
         ,freq.015,freq.016);
         ,freq.017,freq.018,freq.019,freq.02,freq.021,freq.022,freq.023,freq.024,
          freq.025);

############### Graphs ######################


avgDaysTAP = c(avgDaysTAP.008,avgDaysTAP.009,avgDaysTAP.01,

avgDaysTAP.011,avgDaysTAP.012,avgDaysTAP.013,avgDaysTAP.014,avgDaysTAP.015,avgDaysTAP.016);
               ,avgDaysTAP.017,avgDaysTAP.018,avgDaysTAP.019,avgDaysTAP.02);
sumAllTAP=c(totalInfectedTAP.008,totalInfectedTAP.009,totalInfectedTAP.01,totalInfectedTAP.011,
            totalInfectedTAP.012,totalInfectedTAP.013,totalInfectedTAP.014,totalInfectedTAP.015,
            totalInfectedTAP.016);
            ,totalInfectedTAP.015,

totalInfectedTAP.016,totalInfectedTAP.017,totalInfectedTAP.018,totalInfectedTAP.019,totalInfected
TAP.02);
modeAllTAP=c(modeDaysTAP.008,modeDaysTAP.009,modeDaysTAP.01,

modeDaysTAP.011,modeDaysTAP.012,modeDaysTAP.013,modeDays.014,modeDaysTAP.015,modeDaysTAP.016);

modeDaysTAP.015,modeDaysTAP.016,modeDaysTAP.017,modeDaysTAP.018,modeDaysTAP.019,modeDaysTAP.02
probEpiTAP=c(probEpT.008,probEpT.009,probEpT.01,probEpT.011,probEpT.012,probEpT.013,probEpT.014,p
robEpT.015,
             probEpT.016);
            ,probEpT.017,probEpT.018,
             probEpT.019,probEpT.02);
freqAllTAP=c(freqTAP.008,freqTAP.009,freqTAP.01,freqTAP.011,freqTAP.012,
             freqTAP.013,freqTAP.014,freqTAP.015,freqTAP.016);
            ,freqTAP.015,freqTAP.016,freqTAP.017,freqTAP.018,freqTAP.019,freqTAP.02)


yvec=c(8:16);
yvec=yvec*.001;


ax = range(100, sumAll-(2*totalInfectedSD)/sqrt(500), sumAllTAP+(2*totalInfectedTSD)/sqrt(500));
plot(yvec,sumAll, type='o', col="red", axes=T, ann=T, ylab="Expected final size",
xlab="Probability of Transmission",
     main="Expected Final Size", ylim=ax);
lines(yvec,sumAllTAP, col='blue');
points(yvec, sumAllTAP, col='blue');
arrows(yvec, sumAll-(2*totalInfectedSD)/sqrt(500), yvec, sumAll+(2*totalInfectedSD)/sqrt(500),
code=3, angle=90, col='red');
arrows(yvec, sumAllTAP-(2*totalInfectedTSD)/sqrt(500), yvec,
sumAllTAP+(2*totalInfectedTSD)/sqrt(500), code=3, angle=90, col='blue');
legend(.014, 125, c("Baseline","TAP"), cex=0.8, col=c("red","blue"), pch=21:22, lty=1:2);

ax= range(0, modeAllTAP+(2*modeTSD)/sqrt(500), modeAllTAP-(2*modeTSD)/sqrt(500));
plot(yvec,modeAll, type='o',col='red', ylab='Expected Day of Peak Infection', xlab='Probability
of Transmission',
     main='Expected Peak Date',ylim=ax);
lines(yvec,modeAllTAP, col='blue');
points(yvec,modeAllTAP,col='blue');
arrows(yvec, modeAll-(2*modeSD)/sqrt(500), yvec, modeAll+(2*modeSD)/sqrt(500), code=3, angle=90,
col='red');
arrows(yvec, modeAllTAP-(2*modeTSD)/sqrt(500), yvec, modeAllTAP+(2*modeTSD)/sqrt(500), code=3,
angle=90, col='blue')
legend(.014, 20, c("Baseline","TAP"), cex=0.8, col=c("red","blue"), pch=21:22, lty=1:2);
```

```
ax=range(1,freqAll, freqAllTAP+(2*freqTSD)/sqrt(500));
plot(yvec,freqAll,type='o',col='red',ylab='Estimated Number of Infected', xlab='Probability of
Transmission',
     main='Number of Infected on Most Infected Day',ylim=ax);
lines(yvec,freqAllTAP, col='blue');
arrows(yvec, freqAll-(2*freqSD)/sqrt(500), yvec, freqAll+(2*freqSD)/sqrt(500), code=3, angle=90,
col='red');
arrows(yvec, freqAllTAP-(2*freqTSD)/sqrt(500), yvec, freqAllTAP+(2*freqTSD)/sqrt(500), code=3,
angle=90, col='blue')
points(yvec,freqAllTAP, col='blue');

ax = range(.25, probEpi+(2*sd(probEpiSD))/sqrt(500), probEpiTAP-(2*sd(probEpiTSD))/sqrt(500));
plot(yvec, probEpi, type='o',col='red',ylab='Probability of Epidemic,', xlab='Probability of
Transmission',
     main='Probability of Epidemic per Transmission', ylim=ax);
lines(yvec,probEpiTAP, col='blue');
points(yvec,probEpiTAP,col='blue');
arrows(yvec, probEpi-(2*sd(probEpiSD))/sqrt(500), yvec, probEpi+(2*sd(probEpiSD))/sqrt(500),
code=3, angle=90, col='red');
arrows(yvec, probEpiTAP-(2*sd(probEpiTSD))/sqrt(500), yvec,
probEpiTAP+(2*sd(probEpiTSD))/sqrt(500), code=3, angle=90, col='blue')
legend(.014, .3, c("Baseline","TAP"), cex=0.8, col=c("red","blue"), pch=21:22, lty=1:2);
```