Appendix 1

Explanation of model terms and conceptual experimental design

Transmission is the complex process of the successful establishment of a pathogen in a new host, which depends on the pathogen, host, and environment (Antonovics 2017). For a directly transmitted pathogen, this requires 1) contact between an infected host and a susceptible host, and then 2) depends upon the infectiousness and susceptibility of the infected and susceptible host, respectively, as well as the transmissibility of the pathogen itself (Fig. A1). The transmission rate ($\beta$) is a term used in mathematical models that describes how quickly (per unit time) susceptible individuals in a population become infected (Begon et al. 2002). It encompasses both behavioral ($\beta_c$) and physiological ($\beta_p$) aspects of transmission (Hawley et al. 2011), each of which may have non-linear components (McCallum et al. 2017).

Contact rate ($\beta_c$) is the frequency with which hosts interact with each other. In the specific context of pathogen transmission, a contact represents a possible transmission event between an infected and a susceptible host pair. Hosts within a population may be considered superspreaders if they have higher than average contact rates (Lloyd-Smith et al. 2005a). The contact rate depends upon both local processes like group size, mate choice, or sickness behaviors as well as broad-scale processes like resource availability, host density, and migration (Lloyd-Smith et al. 2005b). Contact rate may also be altered by pathogen-induced changes in behavior or sickness behaviors (Ezenwa et al. 2016); we consider this scenario in our third experiment: infection status versus contact rate. In all models in this manuscript, contact rate is governed by the underlying dynamic network model (Fig A2; Section 2.1).

The physiological component of transmission ($\beta_p$), which we incorporate into the model as probability of transmission given an eligible contact between hosts, can be affected by a variety of factors including a host’s innate physiological characteristics such as immunocompetence, shedding rate, latency period, and co-infection (Lehmer et al. 2010, Telfer et al. 2010, Hawley and Altizer 2011, Lass et al. 2013). We incorporate these possible factors into the model via host susceptibility, host infectiousness, and pathogen transmission efficiency (Fig A2). Susceptibility ($s$) describes a
host’s innate immune and physiological response to pathogen exposure that helps determines whether a transmission event is successful. Host susceptibility may be affected by body condition (Beldomenico et al. 2009) and coinfection (Cattadori et al. 2008). Some studies suggest that a host’s immunocompetence and sociality are linked, e.g. more extraverted individuals may be at higher risk of exposure, but may also have more active immune systems (Natoli et al. 2005, Vedhara et al. 2015). For the purposes of this paper, susceptibility only modifies the final probability of transmission, not the contact rate (section 2.4). Thus, we consider susceptibility to be a host-driven trait. Infectiousness (κ) describes how effectively an infected host transmits pathogens to susceptible hosts. Depending on the system, this might be behavioral via sneezing, coughing, or biting, but it could also correspond to the pathogen load or shedding rate. For the purposes of the model, we consider infectiousness to be an intrinsic, host-determined trait (section 2.5). Apart from the final experimental scenario of infection status versus contact rate (section 2.6), we do not consider any feedbacks of infectiousness on host behavior. Finally, we use transmission efficiency (τ) to describe transmissibility of the pathogen itself. This reflects the idea that, all else being equal and host physiology non-withstanding, some pathogens are more infectious than others.

Figure A1. A successful transmission event first requires contact between a susceptible and infected host (blue rectangle). Given contact, a variety of factors affecting host physiology and pathogen transmission efficiency may affect the likelihood of transmission (green rectangle). Infection can result in infection-induced behavioral changes that induce a positive or negative feedback on contact rate (black arrow).
Figure A2. A conceptual representation of the experimental design for this study. (A) Possible forms of covariation between the behavioral and physiological components of transmission, which are represented as contact rate ($\beta_c$) and susceptibility, infectiousness, and infection status ($\beta_p$). Positive covariation is shown in blue; negative covariation is shown in red. (B) The behavioral component of transmission ($\beta_c$) is incorporated in the dynamic network model via contact rate. This panel shows how contact rate is reflected as mean degree (average number of contacts per time step) in the dynamic network. From top to bottom, the focal node (black) has either higher than average, average, or a lower than average number of contacts (blue)—corresponding to the y-axis of panel A.

References
Hawley, D. M. et al. 2011. Does animal behavior underlie covariation between hosts’ exposure to
infectious agents and susceptibility to infection? Implications for disease dynamics. – Integr. Comp. Biol. 51: 528–539.


Appendix 2

Methods supplement

Experiment 1: Susceptibility vs contact rate

Begin by loading the libraries and setting the random seed

```r
require(EpiModel)
require(parallel)
set.seed(4321)
```

Parameter specifications

```r
type.description<-"SvsCR" # Susceptibility vs. contact rate
degree.diff<-2 # degree difference: 2="low"; 4="high"
infection.p<-0.025 # infection probability: high=0.5, med=0.25, low=0.025
file_name<="SvsCR2low"
```

Initialize the networks

We create an undirected network with 525 individuals or nodes. We also set up other specifications for the simulation: how long each simulation should run, the number of repetitions per parameter set, and whether we'd like to run the simulations in parallel.

```r
num<-525; # number of individuals in simulation (ideally divisible by 3)
nw <- network.initialize(n = num, directed = FALSE) # initialize network
duration<-100 # duration of simulations
nsim<-10 # number of disease simulations to run per set of conditions
ncores<-detectCores() # number of cores (either on PC or to optimize on supercomputer)
```

Set up the vertex attributes for the network

Here we establish a gregarious attribute for our nodes which corresponds to $\beta_c$. The population is divided evenly into "low", "medium", and "high" contact rate groups of 175 individuals each.

```r
greg<-c("low","med","high") # add gregariousness attribute to nodes in network
# greg<sample(greg, num, replace=TRUE)
greg<rep(greg, times=num/length(greg))
nw <- set.vertex.attribute(nw, "gregarious", greg)
# check distribution of gregarious phenotype in population
sum(get.vertex.attribute(nw, "gregarious")=="low")
## [1] 175
sum(get.vertex.attribute(nw, "gregarious")=="med")
## [1] 175
sum(get.vertex.attribute(nw, "gregarious")=="high")
## [1] 175
```
Model fit

We set up our formation and dissolution equations for the STERGM in this section. The target stats are done in terms of edge numbers, so if you want to think in terms of mean degree you must convert your target stats accordingly, as we do below. Our dissolution formula is very simple and only has a mean edge duration rate. You can read more about ERGM terms here.

\[
\text{formation} \leftarrow \sim \text{edges} + \text{nodefactor(”gregarious”)}
\]
\[
\text{mean.degree} < - 4
\]
\[
\text{target.stats} \leftarrow \text{c}((\text{num}/2) \ast \text{mean.degree}, (\text{num}/3) \ast (\text{mean.degree}-\text{degree.diff}), (\text{num}/3) \ast \text{mean.degree}) \quad \# \text{mean number of edges (number of nodes/2*degree), number of edges for gregarious=low individuals (number of nodes/3*target degree), number of edges for gregarious=med}
\]
\[
\text{coef.diss} \leftarrow \text{dissolution_coefs(} \text{dissolution} = \sim \text{offset(} \text{edges}, \text{duration} = 2.5) \text{)}
\]

Model diagnostics

Before we simulate disease on the networks, we use the \textit{netest} function to fit a temporal ERGM using our specified formation and dissolution parameters. The \textit{netdx} function runs diagnostics on the fitted networks to see how well the simulated networks match the target specifications. We can view diagnostic results in table format and graphically.

\[
\text{est1} \leftarrow \text{netest(} \text{nw, formation, target.stats, coef.diss, edapprox = TRUE)}
\]
\[
\text{dx} \leftarrow \text{netdx(} \text{est1, nsims = 5, nsteps = 500, nwstats.formula = } \sim \text{edges} + \text{meandeg} + \text{nodefactor(”gregarious”, base = 0)} \text{)}
\]

## Network Diagnostics

### Formulation

#### Target Sim Mean Pct Diff Sim SD

<table>
<thead>
<tr>
<th></th>
<th>Target</th>
<th>Sim</th>
<th>Mean</th>
<th>Pct Diff</th>
<th>Sim</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>edges</td>
<td>1050</td>
<td>1096.762</td>
<td>0.045</td>
<td>34.802</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meandeg</td>
<td>NA</td>
<td>4.178</td>
<td>NA</td>
<td>0.133</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### EpiModel Network Diagnostics

#### Diagnostic Method: Dynamic

<table>
<thead>
<tr>
<th>Simulations: 5</th>
<th>Time Steps per Sim: 500</th>
</tr>
</thead>
</table>

### Formation Diagnostics

#### Target Sim Mean Pct Diff Sim SD

<table>
<thead>
<tr>
<th></th>
<th>Target</th>
<th>Sim</th>
<th>Mean</th>
<th>Pct Diff</th>
<th>Sim</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>edges</td>
<td>1050</td>
<td>1096.762</td>
<td>0.045</td>
<td>34.802</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meandeg</td>
<td>NA</td>
<td>4.178</td>
<td>NA</td>
<td>0.133</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## nodefactor.gregarious.high
\[ 1090.554 \quad \text{NA} \]
## nodefactor.gregarious.low
\[ 350 \quad 366.605 \quad 0.047 \quad 22.649 \]
## nodefactor.gregarious.med
\[ 700 \quad 736.365 \quad 0.052 \quad 31.200 \]

### Dissolution Diagnostics

<table>
<thead>
<tr>
<th>Target Sim</th>
<th>Mean</th>
<th>Pct</th>
<th>Diff</th>
<th>Sim</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Edge Duration</td>
<td>25.00</td>
<td>23.736</td>
<td>-0.051</td>
<td>23.129</td>
<td></td>
</tr>
<tr>
<td>Pct Edges Diss</td>
<td>0.04</td>
<td>0.040</td>
<td>0.001</td>
<td>0.006</td>
<td></td>
</tr>
</tbody>
</table>

```r
plot(dx, plots.joined=FALSE) #Remember for node factor plots, we are looking at number of edges in the network
```

```r
par(mfrow = c(1, 2))
plot(dx, type = "duration")
plot(dx, type = "dissolution")
```
Induce covariation

In order to induce covariation with the "gregarious" node factor, we write some custom modules that feed into EpiModel's \texttt{netsim} function. These functions modify the \texttt{dat} object (which contains all the network information) at time step \texttt{at}.

\textbf{#Positive correlation between susceptibility and contact rate}

\begin{verbatim}
suscept_positive <- function(dat, at) {
  ## Attributes
  if (at == 2) {
    n <- sum(dat$attr$active == 1)
    for (j in 1:n){
      if(dat$attr$gregarious[j]=="med"){
        dat$attr$susceptibility[j]<-1
      }
      if(dat$attr$gregarious[j]=="low"){
        dat$attr$susceptibility[j]<-0
      }
      if(dat$attr$gregarious[j]=="high"){
        dat$attr$susceptibility[j]<-2
      }
    }
    dat$nw <- set.vertex.attribute(dat$nw, "susceptibility", dat$attr$susceptibility)
    index.case<-which(dat$attr$status=="i")
  }
  ## Summary statistics
  if (at == 2) {
    
  }
}
\end{verbatim}
```r
# Negative covariation between susceptibility and contact rate
suscept_negative <- function(dat, at) {
  ## Attributes
  if (at == 2) {
    n <- sum(dat$attr$active == 1)
    for (j in 1:n) {
      if (dat$attr$gregarious[j] == "med") {
        dat$attr$susceptibility[j] <- 1
      }
      if (dat$attr$gregarious[j] == "low") {
        dat$attr$susceptibility[j] <- 2
      }
      if (dat$attr$gregarious[j] == "high") {
        dat$attr$susceptibility[j] <- 0
      }
    }
    dat$nw <- set.vertex.attribute(dat$nw, "susceptibility", dat$attr$susceptibility)
  }

  ## Summary statistics
  if (at == 2) {
    dat$epi$susceptibility <- mean(dat$attr$susceptibility, na.rm = TRUE)
    dat$epi$index.case <- index.case
    dat$epi$suscept.index.case <- dat$attr$susceptibility[index.case]
  }
  return(dat)
}

# Null covariation between susceptibility and contact rate
suscept_null <- function(dat, at) {
  ## Attributes
  if (at == 2) {
    n <- sum(dat$attr$active == 1)
    suscept <- c(0, 1, 2)
    dat$attr$susceptibility <- sample(suscept, n, replace = TRUE) # Susceptibility randomly assigned
  }

  dat$nw <- set.vertex.attribute(dat$nw, "susceptibility", dat$attr$susceptibility)
  index.case <- which(dat$attr$status == "i")
  return(dat)
}
```
Refine transmission function

We also update the standard transmission function from the `EpiModel` package so that it incorporates individual $\beta_p$ via susceptibility. In particular, we update the final transmission probability, $P(t)$.

```r
infection_susceptible <- function (dat, at) {
  active <- dat$attr$active
  status <- dat$attr$status
  susceptibility <- dat$attr$susceptibility
  modes <- dat$param$modes
  mode <- idmode(dat$nw) #id numbers for a bipartite network
  inf.prob <- dat$param$inf.prob
  inf.prob.m2 <- dat$param$inf.prob.m2
  act.rate <- dat$param$act.rate
  nw <- dat$nw
  tea.status <- dat$control$tea.status
  idsSus <- which(active == 1 & status == "s")
  idsInf <- which(active == 1 & status == "i" & susceptibility > 0) #changed here: if initially initially exposed individual has a susceptibility of "0"-- will not propagate infection
  nActive <- sum(active == 1)
  nElig <- length(idsInf)
  nInf <- nInfM2 <- totInf <- 0
  if (nElig > 0 && nElig < nActive) {
    del <- discord_edgelist(dat, idsInf, idsSus, at) #returns data frame with set of edges
    #in which the status of two partners is one is susceptible and one is infected
    if (!is.null(del)) { #if one or more such edges exist...
      inds <- which(get.vertex.pid(nw) %in% del$sus)
      # browser()
      # del$suscept <- susceptibility[inds]
      del$suscept <- susceptibility[del$sus]
      # any(get.vertex.pid(nw) == del$sus)
      # del$suscept <- dat$attr$susceptibility[get.vertex.pid(nw, del$sus)]
      del$infDur <- at - dat$attr$infTime[del$inf] #how long has each node been infected?
```
del$infDur[del$infDur == 0] <- 1
linf.prob <- length(inf.prob)
if (is.null(inf.prob.m2)) {
  del$transProb <- ifelse(del$infDur <= linf.prob, # if the length of
                        inf.prob[del$infDur], inf.prob[linf.prob])
} # then...if not, else....
else {
  del$transProb <- ifelse(del$sus <= nw %>% "bipartite",
                          inf.prob[del$infDur],
                          inf.prob[linf.prob]), ifelse(del$infDur <=
                          inf.prob.m2[del$infDur], inf.prob.m2[linf.prob]))
} # inter.eff - efficacy of an intervention
# inter.start - time at which intervention starts
# if there is an intervention and the current is after the intervention
# start time, then...
if (!is.null(dat$param$inter.eff) && at >= dat$param$inter.start) {
  del$transProb <- del$transProb * (1 - dat$param$inter.eff)
}
lact.rate <- length(act.rate) # act.rate-average number of transmissible acts per partnership per unit time
del$actRate <- ifelse(del$infDur <= lact.rate, act.rate[del$infDur],
                      act.rate[lact.rate])
# del$transProb<- apply(del[,c('transProb', 'susceptibility')], 1, function(x) {
# (x[1]) * (x[2])
})
del$finalProb <- 1 - (1 - del$transProb*del$suscept)^del$actRate
del$finalProb[which(del$finalProb>1)]<-1
# browser()
transmit <- rbinom(nrow(del), 1, del$finalProb)
del <- del[which(transmit == 1), ]
idsNewInf <- unique(del$sus)
nInf <- sum(mode[idsNewInf] == 1)
nInfM2 <- sum(mode[idsNewInf] == 2)
totInf <- nInf + nInfM2
if (totInf > 0) {
  if (tea.status == TRUE) {
    nw <- activate.vertex.attribute(nw, prefix = "testatus",
                                    value = "i", onset = at, termin
    us = Inf,
                                  v = idsNewInf)
  }
  dat$attr$status[idsNewInf] <- "i"
  dat$attr$infTime[idsNewInf] <- at
  form <- get_nwparam(dat)$formation
  fterms <- get_formula_terms(form)
if ("status" %in% fterms) {
    nw <- set.vertex.attribute(nw, "status", dat$attr$status)
}
if (any(names(nw$gal) %in% "vertex.pid")) {
    del$sus <- get.vertex.pid(nw, del$sus)
    del$inf <- get.vertex.pid(nw, del$inf)
}
if (totInf > 0) {
    del <- del![duplicated(del$sus), ]
    if (at == 2) {
        dat$stats$transmat <- del
    } else {
        dat$stats$transmat <- rbind(dat$stats$transmat, del)
    }
    if (at == 2) {
        dat$epi$si.flow <- c(0, nInf)
        if (modes == 2) {
            dat$epi$si.flow.m2 <- c(0, nInfM2)
        } else {
            dat$epi$si.flow[at] <- nInf
            if (modes == 2) {
                dat$epi$si.flow.m2[at] <- nInfM2
            }
        }
    }
    dat$nw <- nw
return(dat)
}

Set up for running epidemic simulations

These components of the simulations are universal, so we set them up first here.

param <- param.net(inf.prob = infection.p, act.rate = 1) #Set infection probability and action rate
init<-init.net(i.num=1) #Start with one infected individual

Positive covariation

The control.net function is where the custom modules can be implemented. We have added a susceptibility.FUN which induces that type of covariation that we want. We also indicate that we want to use the customized transmission function from above rather than the default and that we would like to look at at summary statistics for each 'gregarious' class. If desired, the output can be saved as a .Rdata file.
control1 <- control.net(type = "SI", nsteps = duration, nsims= nsim, ncores=ncores, epi.by = "gregarious", infection.FUN=infection_susceptible, susceptibility.FUN=suscept_positive)
sim1 <- netsim(est1, param, init, control1)
#save(sim1, file=paste(file_name,"positive.RData", sep="_"))

Negative covariation
control2 <- control.net(type = "SI", nsteps = duration, nsims= nsim, ncores=ncores, epi.by = "gregarious", infection.FUN=infection_susceptible, susceptibility.FUN=suscept_negative)
sim2 <- netsim(est1, param, init, control2)
#save(sim2, file=paste(file_name,"negative.RData", sep="_"))

Null
Same contact structure, variability in susceptibility, and no covariation.
control3 <- control.net(type = "SI", nsteps = duration, nsims= nsim, ncores=ncores, epi.by = "gregarious", infection.FUN=infection_susceptible, susceptibility.FUN=suscept_null)
sim3 <- netsim(est1, param, init, control3)
#save(sim3, file=paste(file_name,"null.RData", sep="_"))

Control
Same contact structure, but no difference in susceptibility [i.e., susceptibility, s, always equals 1] and no covariation.
control4 <- control.net(type = "SI", nsteps = duration, nsims= nsim, ncores=ncores, epi.by = "gregarious")
sim4 <- netsim(est1, param, init, control4)
#save(sim4, file=paste(file_name,"control.RData", sep="_"))

Pull summary data from simulations
Here are a few functions to pull summary data from the simulated netsim objects.

colMax<-function(data) sapply(data, max, na.rm=TRUE) #What is the maximum value reached?
timeMax<-function(data) sapply(data, which.max) #When is max value reached?
CI95<-function(data) apply(data,1, quantile, probs=c(2.5, 97.5)/100) #What is 95% quantile CI?
suscept2<-function(simu, nsim){
suscept_2nd<-matrix(NA, nrow = nsim, ncol = 1)
for (i in 1:nsim){
  #store<-simu$network[[i]] #access stored dynamicNetwork object
  trans<-get_transmat(simu, sim=i) #access transmission matrix
  head(trans)
  if(length(trans)==0){
    suscept_2nd[i,1]<-NA
  }
}
#susceptibility of first cases (after index case)
else{
    trans[1,1] # time of first transmission event
    rowz<-which(trans[,1]==trans[1,1])
    suscept_2nd[i,1]<-mean(trans[rowz,4]) # mean of susceptibility of secondary cases
}
}
suscept_2nd<-as.matrix(suscept_2nd)
return(suscept_2nd)

We use those functions to pull summary statistics from each of our four scenarios.

positive<-sim1$epi$i.num
index.case.positive<-sim1$epi$suscept.index.case
suscept_2nd.positive<-suscept2(sim1, nsim)
write.csv(positive,file=paste(file_name,"positive.csv", sep=" "))
mean_positive<-rowMeans(positive)
maximum_positive<-colMax(positive)
peak_time_positive<-timeMax(positive)
quants_positive<-t(as.matrix(CI95(positive)))

negative<-sim2$epi$i.num
index.case.negative<-sim2$epi$suscept.index.case
suscept_2nd.negative<-suscept2(sim2, nsim)
write.csv(negative,file=paste(file_name,"negative.csv", sep=" "))
mean_negative<-rowMeans(negative)
maximum_negative<-colMax(negative)
peak_time_negative<-timeMax(negative)
quants_negative<-t(as.matrix(CI95(negative)))

null<-sim3$epi$i.num
index.case.null<-sim3$epi$suscept.index.case
suscept_2nd.null<-suscept2(sim3, nsim)
write.csv(null,file=paste(file_name,"null.csv", sep=" "))
mean_null<-rowMeans(null)
maximum_null<-colMax(null)
peak_time_null<-timeMax(null)
quants_null<-t(as.matrix(CI95(null)))

control<-sim4$epi$i.num
index.case.control<-rep(1,nsim)
suscept_2nd.control<-rep(1,nsim)
write.csv(control,file=paste(file_name,"control.csv", sep=" "))
mean_control<-rowMeans(control)
maximum_control<-colMax(control)
peak_time_control<-timeMax(control)
quants_control<-t(as.matrix(CI95(control)))
Create summary data frames

Since the simulation objects themselves can be quite large, especially for longer runs or more repeats, these summary dataframes can be saved as .csv files for easy access later.

timecourse.df <- data.frame(
  mean_positive=mean_positive,
  positive_CI=quants_positive,
  mean_negative=mean_negative,
  negative_CI=quants_negative,
  mean_control=mean_control,
  control_CI=quants_control)

summary.df <- data.frame(
  maximum_positive,
  peak_time_positive,
  maximum_negative,
  peak_time_negative,
  maximum_null,
  peak_time_null,
  maximum_control,
  peak_time_control)

write.csv(timecourse.df, file=paste(file_name,"timecourse.csv", sep="_"))
write.csv(summary.df, file=paste(file_name,"summary.csv", sep="_"))

We also set up data frames in a long data format for random forest analysis and figure generation.

covariation<- c("positive", "negative", "null", "control")
covariation<- rep(covariation, each=nsim)
type<- rep(type.description, times=4*nsim)
infection.prob<- rep(infection.p, times=4*nsim)
maximum<- c(maximum_positive, maximum_negative, maximum_null, maximum_control)
peak_time<- c(peak_time_positive, peak_time_negative, peak_time_null, peak_time_control)
index.case<- as.numeric(c(index.case.positive, index.case.negative, index.case.null, index.case.control))
suscept_2nd<- as.numeric(c(suscept_2nd.positive, suscept_2nd.negative, suscept_2nd.null, suscept_2nd.control))
long.df<- data.frame(type, infection.prob, covariation, maximum, peak_time, index.case, suscept_2nd)
write.csv(long.df, file=paste(file_name,"long.csv", sep="_"))

Plot data

Last, but not least, we can do a quick visual check and plot our simulation data.

par(mfrow = c(1,1))
plot(sim1, ylim=c(0,num+100), y = "i.num", sim.lines = FALSE, qnts = 1)
  mean.col = "firebrick", qnts.col = "firebrick", add = TRUE)
plot(sim2, y = "i.num", sim.lines = FALSE, qnts = 1,
  mean.col = "black", qnts.col = "black", add = TRUE)
plot(sim3, y = "i.num", sim.lines = FALSE, qnts = 1,
  mean.col = "purple", qnts.col = "purple", add = TRUE)
legend("topright", c("Positive Correlation", "Negative Correlation", "Null",
  "Control"), lty = 1, lwd = 3,
  col = c("steelblue", "firebrick", "black", "purple"), cex = 0.9, bty = "n")
par(mfrow = c(1, 1), mar = c(3.5, 4, 1, 1), mgp = c(2.25, 1, 0), cex.axis=1.5, cex.lab=1.5)
par(lwd=2)
plot(sim1, y = "i.num", sim.lines = FALSE, qnts=FALSE)
plot(sim2, y = "i.num", sim.lines = FALSE, qnts=FALSE, mean.col = "firebrick", add = TRUE)
plot(sim3, y = "i.num", sim.lines = FALSE, qnts=FALSE, mean.col = "black", add = TRUE)
plot(sim4, y = "i.num", sim.lines = FALSE, qnts=FALSE, mean.col = "purple", add = TRUE)
legend("topleft", c("Positive Correlation", "Negative Correlation", "Null", "Control"), lty = 1, lwd = 3, col = c("steelblue", "firebrick", "black", "purple"), cex = 1.25, bty = "n")
Session Information

## Record version information used in this analysis

```r
print(sessionInfo(), locale = TRUE)
```

## R version 3.3.2 (2016-10-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 14393)

## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252

## attached base packages:
## [1] parallel stats graphics grDevices utils datasets methods

## other attached packages:
## [1] EpiModel_1.2.7 tergm_3.4.0 ergm_3.6.0
## [4] statnet.common_3.3.0 networkDynamic_0.9.0 network_1.13.0
## [7] deSolve_1.13

## loaded via a namespace (and not attached):
## [1] Rcpp_0.12.6 knitr_1.14 magrittr_1.5
## [4] MASS_7.3-45 doParallel_1.0.10 ape_3.5
## [7] lattice_0.20-34 foreach_1.4.3 stringr_1.1.0
<table>
<thead>
<tr>
<th>Package</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>tools_3.3.2</td>
<td>grid_3.3.2</td>
</tr>
<tr>
<td>nlme_3.1-128</td>
<td>coda_0.18-1</td>
</tr>
<tr>
<td>htmltools_0.3.5</td>
<td>lazyeval_0.2.0</td>
</tr>
<tr>
<td>digest_0.6.10</td>
<td>Matrix_1.2-7.1</td>
</tr>
<tr>
<td>formatR_1.4</td>
<td>codetools_0.2-15</td>
</tr>
<tr>
<td>robustbase_0.92-6</td>
<td>evaluate_0.9</td>
</tr>
<tr>
<td>stringi_1.1.1</td>
<td>compiler_3.3.2</td>
</tr>
<tr>
<td>DEoptimR_1.0-6</td>
<td></td>
</tr>
</tbody>
</table>
Appendix 3

Derivation of logistic growth equation

We sought to measure the rate at which the pathogen was spreading through the theoretical population, or functionally, a realized transmission rate or ‘realized $\beta$’. We start with an underlying SI model, but with the addition of some fraction of the susceptible pool, $r$, that cannot be infected due to social isolation or lack of susceptibility (as a result of experimental design with variable $\beta_c$ and/or $\beta_p$):

$$\frac{dS}{dt} = -\beta (1 - r)SI$$

$$\frac{dI}{dt} = \beta (1 - r)SI$$

We further assume that the carrying capacity, $K$ (in this context, the maximum number of individuals that can be infected), can vary and reflects the sum of the number of eligible susceptible infected individuals and infected individuals:

$$K = (1 - r)S + I \rightarrow S = \frac{K - I}{1 - r}$$

We than solve for $\frac{dI}{dt}$ in terms of $I$:

$$\frac{dI}{dt} = \beta I (1 - r) \frac{K - I}{1 - r} = \beta I (K - I)$$

We can put this in the recognizable, logistic growth form by multiplying the right side by $K/K$:

$$\frac{dI}{dt} = \beta IK \left( \frac{K - I}{K} \right) = \beta I \left( K \left( 1 - \frac{I}{K} \right) \right)$$

After integrating with partial fractions and using $I(t = 0) = I_0$ as the initial condition, we get:

$$I(t) = \frac{K}{1 + \frac{(K-I_0)}{I_0} e^{-\beta K t}}$$
Appendix 4

Additional time course simulations

Figure A3. Time course of simulated epidemics for susceptibility vs. contact rate for an infection probability of $\tau = 0.25$. Columns correspond to the difference in mean degree tested, and rows correspond to the mechanism of covariation: control (no variability in susceptibility, no covariation), null (variability in susceptibility, no covariation), positive covariation, and negative covariation. Individual trials are shown as semi-transparent lines and the colors—black, purple, blue, and red—correspond to control, null, positive, and negative covariation cases respectively. Percentages in the lower right hand corner of each panel describe the percentage of epidemics fading out. The dashed lines in each panel correspond to the expected maximum prevalence based on contact structure. For higher variations in contact rate, one-third of the population has a $\beta_c = 0$ limiting maximum prevalence to 0.66.
Figure A4. Time course of simulated epidemics for susceptibility vs. contact rate for an infection probability of $\tau = 0.5$. Columns correspond to the difference in mean degree tested, and rows correspond to the mechanism of covariation: control (no variability in susceptibility, no covariation), null (variability in susceptibility, no covariation), positive covariation, and negative covariation. Individual trials are shown as semi-transparent lines and the colors – black, purple, blue, and red – correspond to control, null, positive, and negative covariation cases respectively. Percentages in the lower right hand corner of each panel describe the percentage of epidemics fading out. The dashed lines in each panel correspond to the expected maximum prevalence based on contact structure. For higher variations in contact rate, one-third of the population has a $\beta_c = 0$ limiting maximum prevalence to 0.66.
Figure A5. Time course of simulated epidemics for infectiousness vs. contact rate for the medium infection probability tested of $\tau = 0.25$. Columns correspond to the difference in mean degree tested, and rows correspond to the mechanism of covariation: control (no variability in infectiousness, no covariation), null (variability in infectiousness, no covariation), positive covariation, and negative covariation. Individual trials are shown as semi-transparent lines and the colors – black, purple, blue, and red – correspond to control, null, positive, and negative covariation cases respectively. Percentages in the lower right hand corner of each panel describe the percentage of epidemics fading out. The dashed lines in each panel correspond to the expected maximum prevalence based on contact structure. For higher variations in contact rate, one-third of the population has a $\beta_c = 0$ limiting maximum prevalence to 0.66.
Figure A6. Time course of simulated epidemics for infectiousness versus contact rate for the high infection probability tested of $\tau = 0.5$. Columns correspond to the difference in mean degree tested, and rows correspond to the mechanism of covariation: control (no variability in infectiousness, no covariation), null (variability in infectiousness, no covariation), positive covariation, and negative covariation. Individual trials are shown as semi-transparent lines and the colors – black, purple, blue, and red – correspond to control, null, positive, and negative covariation cases respectively. Percentages in the lower right hand corner of each panel describe the percentage of epidemics fading out. The dashed lines in each panel correspond to the expected maximum prevalence based limiting maximum prevalence to 0.66.
Figure A7. Time course of simulated epidemics for infection status versus contact rate for the medium infection probability tested of $\tau = 0.25$. Columns correspond to how infectiousness was modelled (either in the exponent or the product of the final transmission probability), and rows correspond to the mechanism of covariation: control (all infection statuses have equal mean degree), null (variability in susceptibility, no covariation), positive covariation, and negative covariation. Individual trials are shown as semi-transparent lines and the colors – black, purple, blue, and red – correspond to control, null, positive, and negative covariation cases respectively. Percentages in the lower right hand corner of each panel describe the percentage of epidemics fading out. The dashed lines in each panel correspond to the expected maximum prevalence based on contact structure.
Figure A8. Time course of simulated epidemics for infection status versus contact rate for the high infection probability tested of $\tau = 0.5$. Columns correspond to how infectiousness was modelled (either in the exponent or the product of the final transmission probability), and rows correspond to the mechanism of covariation: control (all infection statuses have equal mean degree), null (variability in susceptibility, no covariation), positive covariation, and negative covariation. Individual trials are shown as semi-transparent lines and the colors – black, purple, blue, and red – correspond to control, null, positive, and negative covariation cases respectively. Percentages in the lower right hand corner of each panel describe the percentage of epidemics fading out. The dashed lines in each panel correspond to the expected maximum prevalence based on contact structure.