



epinet: An R Package to Analyze Epidemics Spread across Contact Networks

Chris Groendyke
Robert Morris University

David Welch
University of Auckland

Abstract

We present the R package **epinet**, which provides tools for analyzing the spread of epidemics through populations. We assume that the relationships among individuals in a population are modeled by a contact network described by an exponential-family random graph model and that the disease being studied spreads across the edges of this network from infectious to susceptible individuals. We use a susceptible-exposed-infectious-removed compartmental model to describe the progress of the disease within each host. We describe the functionality of the package, which consists of routines that perform simulation, plotting, and inference. The main inference routine utilizes a Bayesian approach and a Markov chain Monte Carlo algorithm. We demonstrate the use of the package through two examples, one involving simulated data and one using data from an actual measles outbreak.

Keywords: **epinet**, Markov chain Monte Carlo, exponential-family random graph model, ERGM, epidemic, network.

1. Introduction

The **epinet** package (Groendyke and Welch 2018) for R (R Core Team 2017) allows users to infer network models from time-series epidemic data and simulate epidemics over networks. The software will be of use to statisticians, epidemiologists, biologists or others who wish to investigate the population processes involved in infectious disease transmission either via inference using empirical data obtained from a relatively small outbreak or via *in silico* simulation. The package is available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=epinet>.

Networks have been used to model the contact structure across which disease causing pathogens spread since at least the mid-20th century (Bailey 1957) for the simple reason that they are very natural models of this system. Networks allow a fine-scale modeling of the heterogeneity

seen in the number of contacts that individuals are involved in. Nodes in the network represent host individuals and edges between nodes represent possible paths for transmission so that heterogeneity in contact numbers is represented as heterogeneity in the degree of nodes in the network. Accurately modeling this heterogeneity is crucial to understanding the dynamics of epidemics and therefore to planning any control strategies (see [Danon *et al.* 2011](#), for a full discussion of the benefits of network modeling in epidemiology).

While the simple epidemic network models are well-studied using analytic tools (e.g., [Kenah and Robins 2007](#), and references therein) and many more complex models have been proposed and studied using simulation ([Welch, Bansal, and Hunter 2011](#); [Moslonka-Lefebvre, Pautasso, and Jeger 2009](#); [Melnik, Hackett, Porter, Mucha, and Gleeson 2011](#); [Badham and Stocker 2010](#)), there are few tools for the empirical scientist who obtains epidemiological data and wishes to reconstruct the contact network across which it spread.

Package **epinet** estimates the parameters of a parametric model for the contact network from temporal data obtained from an epidemic. All estimates are made in a Bayesian context using Markov chain Monte Carlo (MCMC) techniques, which produce samples from the posterior distribution of all unknowns in the model. The list of parameters estimated are the network itself, the transmission tree, and scalar parameters controlling the contact formation process (the network) and the transmission process.

All of the major software packages that we are aware of for fitting time series data to epidemic models are R packages. These packages include **EpiEstim** ([Cori 2013](#)), **surveillance** ([Meyer, Held, and Höhle 2017](#)), **R0** ([Boelle and Obadia 2015](#)) and **epicalc** ([Chongsuvivatwong 2012](#)). None of these explicitly place an unknown contact network at the heart of their model.

There are several existing software packages that fit network models to observed network data, such as **ergm** ([Handcock, Hunter, Butts, Goodreau, Krivitsky, and Morris 2017](#); [Hunter, Handcock, Butts, Goodreau, and Morris 2008](#)), **Bergm** ([Caimo and Friel 2014](#)) and **hergm** ([Schweinberger, Handcock, and Luna 2017](#)). These packages all take as input a fixed observed network or networks, and estimate the parameters of a specified model. The class of models available in these packages are a superset of those available in **epinet**. Package **epinet** is unique in taking time-series epidemic data as input and producing parameter estimates of a network model.

Package **epinet** also provides simulation tools to simulate networks from certain network models and to simulate epidemics across given networks. The simulation tools are particularly useful when used in conjunction with the inference tools to perform, for example, posterior predictive modeling and assessing model fit.

The remainder of the paper is structured as follows. In Section 2, we describe the network and epidemic models available in package **epinet** and the form of the data required by the inference engine. In Section 3, we describe the primary functions and interface used in the package and in Section 4 we go through two detailed examples of inference, one using simulated data, the other using real data. We conclude in Section 5 with a brief discussion of future work on this package.

2. Models and data

Before we describe in detail the models we use, we look at three basic modeling assumptions made in package **epinet**. The first is that the pattern of contacts within the population is modeled as a static network and the population size is known (so that the number of nodes in

the network is known and fixed). The second is that the spread of the pathogen over a given network is modeled as a stochastic compartmental model. The third is that the observed data consists of event times at which individuals moved from one compartment to another (for example, times of infection or removal) and that it is known which individuals are infected during the course of the epidemic and which are not.

The contact network is a static, undirected network in which the individuals are vertices (nodes) and an edge (tie) exists between two vertices when there is sufficient contact between the individuals for the pathogen to be transmitted if it is present. Exactly what constitutes contact depends on the pathogen being modeled: the network associated with a sexually transmitted pathogen will be very different to a network associated with a pathogen spread by respiration. It is important to note that the presence of a contact does not necessarily mean that transmission occurs, so (a, b) could be an edge in the contact network and a gets infected but b never gets infected. For example, someone could have sufficient contact with a person so that transmission is possible but, by chance, it never occurs.

Contrast a static network with the actual contact process which is dynamic in that contacts between individuals – times when transmission can occur – are fleeting and may or may not reoccur. The static contact network is typically considered as an approximation of these dynamic networks where contacts that pass some threshold (frequency, length or a combination of the two) are recorded, while contact below that threshold is considered unimportant. A static network assumption is most appropriate where the time scale of a typical epidemic is much faster than the time scale of changes in the population, such as birth, death or migration, and when transmissions of the pathogen are primarily across persistent or recurrent rather than fleeting contacts.

The use of a stochastic compartmental model for the infection is widespread and uncontroversial. While we implement one particular class of models here – namely, the susceptible-exposed-infectious-removed (SEIR) class – the framework we present could easily be extended to other classes of compartmental model.

We require that the epidemic data include: the size of the initial susceptible population, the total number infected during the outbreak and, for each infected host, (some proxy of) the time of one of the transitions between compartments, for example, the time of exposure, infection or recovery. This assumption limits our approach to well-documented outbreaks in closed populations.

2.1. Network models

The model we use to represent the contact network within a population of N individuals is an exponential-family random graph model (ERGM; Frank and Strauss 1986; Handcock, Hunter, Butts, Goodreau, and Morris 2008). If \mathcal{G} is an ERGM with a fixed number of nodes, N , it can be represented by the discrete probability mass function

$$f(\mathcal{G}) = \frac{\exp\{\boldsymbol{\eta}^\top g(\mathcal{G})\}}{\kappa(\boldsymbol{\eta})}, \quad (1)$$

where $g(\mathcal{G})$ is a vector of length k of graph statistics defining the ERGM, which could include terms such as number of edges, number of triangles or cycles, and homophily terms, among others. The parameter vector $\boldsymbol{\eta} = (\eta_1, \dots, \eta_k)^\top$ is such that η_i corresponds to the i -th network statistic and $\kappa(\boldsymbol{\eta})$ is a normalizing function. ERGMs encompass a wide range of

models, including some important special cases such as the Erdős-Rényi model (Gilbert 1959; Erdős and Rényi 1959). See Hancock *et al.* (2008) for a discussion of the advantages of using ERGMs to model contact networks.

In package **epinet**, we utilize a specific class of ERGMs known as dyadic-independent models, meaning that the probability of an edge existing between any two nodes in the population is independent of the absence or presence of any other edges in the network. The specific subclass of dyadic-independent models we use has a probability mass function of the form

$$f(\mathcal{G}) = \frac{\exp \left\{ \sum_{t=1}^k \eta_t \sum_{\{i,j\}} \mathcal{G}_{\{i,j\}} \mathbf{X}_{\{i,j\},t} \right\}}{\kappa(\boldsymbol{\eta})}, \quad (2)$$

where \mathbf{X} is a $\binom{N}{2} \times k$ matrix of dyadic covariates, with one row corresponding to each of the $\binom{N}{2}$ dyads and one column for each of the covariates in the model; $\mathcal{G}_{\{i,j\}}$ is an indicator of the presence of an edge between individuals i and j . Under this class of models, we can calculate the probability of an edge between any two nodes i and j as $p_{\{i,j\}} = \mathbb{P}(\mathcal{G}_{i,j} = 1)$, where

$$\log \left(\frac{p_{\{i,j\}}}{1 - p_{\{i,j\}}} \right) = \sum_{t=1}^k \mathbf{X}_{\{i,j\},t} \boldsymbol{\eta}_t. \quad (3)$$

There are some notable advantages associated with the subclass of ERGMs given in Equation 2. The model is intuitively understandable and it is easy to interpret the model parameters: each coefficient can be interpreted as the incremental log odds of the probability of an edge associated with the corresponding effect. The dyadic-independence property of these models also makes them computationally manageable. As described in Groendyke, Welch, and Hunter (2012), all of the manipulations of the network in the MCMC algorithm can be done iteratively by cycling through the dyads. The normalizing function $\kappa(\boldsymbol{\eta})$ in Equation 1, which is in general intractable, becomes computationally tractable for this subclass. In addition, the models described by Equation 2 are a rich enough subclass of ERGMs to realistically model contact networks in many situations. Any statistic whose value is a function of the properties of a dyad (and the individuals comprising the dyad) can be included in this subclass of model. Some of these types of terms include uniform and differential homophily terms (sometimes called matching or assortative terms) (Cauchemez *et al.* 2011) and various types of distance metrics. We use both of these types of terms in the examples in Section 4.

The major disadvantage of a dyadic-independence ERGM is that it does not capture some important effects influencing edge formation in social networks and, by extension, contact networks. One key characteristic of social networks is transitivity – if (i, j) and (j, k) are edges then (i, k) is more likely an edge too, or, more prosaically, friends of friends are likely to be friends (Wasserman 1994). Including a dyadic-dependent term such as transitivity in an ERGM results in an intractable normalization constant, $\kappa(\boldsymbol{\eta})$, and makes inference for these models very difficult. Despite the importance of dyadic dependence terms, we believe there is sufficient richness in the dyadic-independent ERGMs to make them useful models for many situations.

2.2. Epidemic models and the transmission tree

Given a contact network, we assume that an epidemic spreads across it as an SEIR compartmental model (for a detailed description, see Keeling and Rohani 2007). The compartments

S, E, I, and R are based on an individual's current disease status. Individuals that have not yet been infected, but could be in the future are in the susceptible class. Upon transmission, an individual moves to the exposed class; in this class, the individual cannot yet infect others. After some period of time, an exposed individual becomes infectious, at which point they are able to infect individuals in the susceptible class. Lastly, infectious individuals are removed, at which time they can no longer infect others and they play no further role in the epidemic. Removal could be due to, for example, acquired immunity or death. For epidemics where re-infection plays a significant role, an SIS or SEIRS model should be used, but no such model is currently implemented in package **epinet**.

Thus, the progression of an SEIR infection in each host is described by:

$$\text{Susceptible} \rightarrow \text{Exposed} \rightarrow \text{Infectious} \rightarrow \text{Removed}.$$

The epidemic begins when a single individual becomes infected. The infection may be transmitted across any edge from an infectious to a susceptible node. The waiting time for a transmission across a particular edge is modeled by an exponential random variable with mean $1/\beta$. The times spent in the exposed and infectious states are modeled as gamma random variables with parameters (θ_E, k_E) and (θ_I, k_I) , respectively, where we use the parametrization of the gamma distribution which has mean $k\theta$ and variance $k\theta^2$.

The subgraph of the contact network that includes all nodes infected during the epidemic and the edges across which transmission occurred is called the transmission tree and denoted \mathcal{P} . It is a tree because it includes no cycles (since any node can only be infected once) and is a directed graph where the direction of the edges is given by the direction of transmission. The transmission tree can be visualized simply as a directed network or, coupled with times at which infection and recovery events occurred, as a timed tree which gives a lot of detail about the progress of the epidemic as seen in Figure 1 and described in Section 4.

2.3. Data

The primary data used in the inferential procedure are the times at which each individual in the population enters each of the exposed, infectious, and removed states. In addition, it is necessary to know how many members of the population were never infected (remained susceptible throughout the course of the epidemic). In the case that either the exposed or infectious (or both) times are not known, they can be inferred as part of the MCMC algorithm; however, knowing more data strengthens the ability of the procedure to infer the other parameters. In reality, the exact times of state transitions will usually not be known, and it will be necessary to use some sort of proxies for these times. In some cases, the onset of various symptoms in hosts can be mapped to the transitions between stages in the compartmental model. The difficulty of acquiring this type of data is typically the greatest difficulty in performing the analysis described here.

Times may be given on any scale and arbitrarily shifted. We often shift the times so that the first recorded event corresponds to $t = 0$.

If a network model using dyadic covariates is being used, then the covariates for each individual in the population must be known and form part of the data. Finally, information we might have about the transmission tree, such as known infectious contacts, may be incorporated in an analysis; Section 2.4 describes this in more detail.

2.4. Priors

We complete the specification of the model by specifying independent prior distributions for the model parameters. The $\boldsymbol{\eta}$ parameters governing the probabilities of edge formation in the contact network are assigned normal prior distributions. Typically we use vague normal priors to indicate a lack of prior information for these parameters, though when we have prior knowledge regarding these parameters, the parameters of the priors can be specified accordingly.

The prior distributions for the epidemic parameters can be chosen to reflect any relevant biological information known about the pathogen in question. Specifically, the θ_E and θ_I parameters can be set to inverse gamma distributions, while prior information for the β , k_I , and k_E parameters can be specified using gamma distributions. These distributions are flexible enough to be useful in a variety of situations. If little or no prior information is known regarding these parameters, this uncertainty can be specified using uniform distributions on the range of biologically plausible values for each parameter.

By default, the prior distribution for the transmission tree is uniform on the set of all possible transmission trees. Equivalently, the default prior distribution for the parent of (i.e., the individual responsible for infecting) node j is uniformly distributed on the set of all individuals that were in the infectious state when j became exposed. If we have reason to believe that a particular individual is responsible for this infection, we can incorporate this information by multiplying the weight given to this individual in the prior distribution by a factor greater than 1. [Groendyke, Welch, and Hunter \(2011\)](#) studies the effect of various weights on the resulting posterior distribution of the transmission tree. In the cases where it is necessary to infer the exposure and/or infectious times of individuals, we use a flat (uninformative) prior distribution for these times.

2.5. Inference

All inference is done within a Bayesian framework. We use MCMC techniques to sample from the otherwise intractable joint posterior distribution. The parameters over which the posterior is defined are the network parameters $\boldsymbol{\eta}$, epidemic parameters (β , θ_E , θ_I , k_E , and k_I), the network, \mathcal{G} , and the transmission tree, \mathcal{P} . The marginal posterior distribution of any set of parameters can be studied by simply ignoring sampled values of parameters that are not of interest. The network and transmission tree are included so that the likelihood can be more easily calculated, but these parameters may not always be of interest. Full details of the implementation of the algorithm are given in [Groendyke *et al.* \(2011\)](#) and [Groendyke *et al.* \(2012\)](#).

3. The R package *epinet*

The R package *epinet* does simulation, plotting, and Bayesian inference for epidemics assumed to spread through a population across edges of a contact network. The underlying contact network is assumed to be a dyadic-independent ERGM of the type specified in Section 2.1; the epidemic model is an SEIR model as described in Section 2.2.

In the main routine implementing the MCMC algorithm, a binary tree structure is used internally to store the contact network; this structure has the advantage of efficiently han-

dling the storage and manipulation of the sparse networks that are often encountered in the situations considered here. The implementation of this structure and some of the internal methods operating on the network are very similar to those used in the **ergm** package for R. The representation of the transmission tree is straightforward: for each node, we need only store the node responsible for its infection. In the remainder of this section, we describe the primary functions available in package **epinet**.

3.1. Simulation

The **epinet** package can be used to simulate the spread of an epidemic through a population across a given undirected contact network. A network is represented in package **epinet** as the pair (N, M) where N is the number of nodes and M an edge list matrix which has a row for each edge in the network and two columns giving the identities of the nodes sharing the edge. A routine for simulating random edge lists according to the ERGM model given in Equation 2 is provided as `SimulateDyadicLinearERGM()`, which takes as inputs a matrix of dyadic covariates and a vector of network parameters, denoted in Equation 2 by \mathbf{X} and $\boldsymbol{\eta}$, respectively. Other network simulation functions are available in the **ergm** and **network** (Butts 2015) packages.

Given a network, the main function performing epidemic simulation is `SEIR.simulator()`:

```
SEIR.simulator(M, N, beta, ki, thetai, ke = ki, thetae = thetai,
  latencydist = "fixed", latencyperiod = 0)
```

The arguments `beta`, `ki`, `thetai`, `ke`, and `thetae` correspond to the parameters β , k_I , θ_I , k_E , and θ_E (as defined in Section 2.2), respectively. The latency period (time spent in the exposed state by each individual) can be modeled either as a fixed period of time (`latencydist = "fixed"`) or as a gamma random variable (`latencydist = "gamma"`). When the latent period is fixed, the argument `latencyperiod` specifies the length of the latency period, and the `thetae` and `ke` arguments are ignored. When the latent period is gamma-distributed, the `latencyperiod` argument is ignored.

The output of this function is an object of class ‘`epidemic`’, a simulated epidemic in the matrix form required by **epinet**, with one row for each individual in the population. The first two columns give the IDs of the individual and the individual responsible for their infection. The final three columns give the times that the individual entered the exposed, infectious, and removed states. For individuals not infected in the epidemic, the final four columns are assigned NA values. There are S3 methods `print`, `summary`, and `plot` for objects of class ‘`epidemic`’.

3.2. Inference

The MCMC algorithm used to generate posterior samples is implemented in C (Kernighan, Ritchie, and Ekelint 1988); it is interfaced in R through the `epinet()` function. This algorithm is based on that described in Britton and O’Neill (2002), Groendyke *et al.* (2011), and Groendyke *et al.* (2012):

```
epinet(formula, epidata, dyadiccovmat, mcmcinput = MCMCcontrol(),
  priors = priorcontrol(), verbose = TRUE)
```

The inputs to the function include: `formula`, a ‘`formula`’ object that specifies the covariates to be included in the model (an intercept term is included by default), `epidata`, a matrix in the format produced by `SEIR.simulator()`, `dyadiccovmat`, a matrix of dyadic covariates, `mcmcinput`, a list which sets the controls for the MCMC algorithm, `priors`, the list of prior distributions and hyperparameters for the model parameters, and `verbose`, which controls the level of output printed to the screen during the operation of the routine.

The `mcmcinput` parameter can either be set directly or via a call to `MCMCcontrol()`. The number of MCMC iterations and thinning interval are controlled by `nsamp` and `thinning`, respectively. Because the posterior samples for the inferred exposure times, inferred infectious times, and transmission tree parameters can be very memory intensive, the input argument `extrathinning` can be used to further thin the output for them; the possible values for this argument are `FALSE` (for no extra thinning) or a positive integer specifying the additional thinning interval. The desired number of burn-in iterations to be performed by the MCMC routine is specified by the `burnin` parameter, which defaults to 0. The `seed` parameter allows the user to manually set the random seed. Finally, the parameter `etapropsd` is an array of standard deviations for the proposal distributions of the η parameters; these are tuning parameters for the MCMC algorithm.

The `priors` parameter can either be set directly or via a call to `priorcontrol()`. The forms of the prior distributions for the epidemic parameters (β , k_I , θ_I , k_E , and θ_E) can be set individually, or can be specified collectively using the `priordists` argument. If “`gamma`” is selected, the θ_E and θ_I parameters are assigned inverse gamma prior distributions, while the other epidemic parameters use gamma distributions. If “`uniform`” is chosen, all of the epidemic parameters are assigned uniform distributions. The arguments `bprior`, `tiprior`, `teprior`, `etaprior`, `kiprior`, and `keprior` give the hyperparameters for the prior distributions of these parameters. The prior distributions of the network (η) parameters are specified to be normal whose means and standard deviations are specified in the `etaprior` argument. We can also specify the prior distribution for the parent of the individuals. The default is a uniform prior on the set of possible parents, i.e., the set of individuals who were infectious at the time of exposure. If we have reason to believe that a particular individual is more likely to be responsible for infecting another, we can put additional weight on the corresponding prior probability. This is done by specifying the suspected parent of the individual in the second column of the `epidata` argument, and by indicating the multiplier for the prior probability in the `parentprobmult` argument.

The primary output of this function are samples from the posterior distribution of the model parameters, β , θ_E , k_E , θ_I , k_I and η . Posterior samples for the exposure and infectious times can also be returned, along with those corresponding to the identity of the initially infected individual, initial exposure time and the transmission tree. Other output values include the log-likelihood for each recorded sample and the number of proposed and accepted moves of each move type within the MCMC algorithm. All of these values can be inspected to assess whether or not the chain has converged and, therefore, whether or not the samples obtained are actually from the desired posterior. The remaining output values include the function call, the model formula, and the MCMC control input. The fitted model is an object of class ‘`epinet`’; there are S3 methods `print`, `summary`, and `plot` for this class.

Function	Input(s)	Output
<code>BuildX</code>	Types of covariate(s) desired, e.g., matching, distance, etc.	A matrix of dyadic covariates of the form required by <code>SEIR.simulator()</code> and <code>epinet()</code> .
<code>SimulateDyadicLinearERGM</code>	Network (η) parameters for desired network.	A simulated ERGM network in edge list matrix format.
<code>plot.epidemic</code>	An epidemic in the form produced by <code>SEIR.simulator()</code> .	A graph visually depicting the epidemic.
<code>plot.epinet</code>	MCMC output in the form produced by <code>epinet()</code> .	A graph visually depicting the state of the epidemic at a particular point in the MCMC chain.
<code>epi2newick</code>	An epidemic in the form produced by <code>SEIR.simulator()</code> .	The transmission tree corresponding to the epidemic as a string in Newick format.
<code>epi2newickmcmc</code>	MCMC output in the form produced by <code>epinet()</code> .	The transmission tree corresponding to an inferred epidemic as a string in Newick format.
<code>write.epinet</code>	MCMC output in the form produced by <code>epinet()</code> .	Written output files consisting of the posterior samples of the epidemic parameters and transmission trees.
<code>ess</code>	MCMC output for a single variable, i.e., a sample from the posterior distribution of a variable.	An estimate of the ESS (effective sample size) of the sample.

Table 1: Description of some of the functions in package **epinet**.

3.3. Other functions

In addition to the epidemic simulation and inference routines described above, the **epinet** package also contains some auxiliary functions whose inputs and outputs are briefly described in Table 1.

4. Examples

In this section, we demonstrate the use of the **epinet** package through two examples, the first using simulated data, and the second involving data from an actual epidemic.

4.1. An example using simulated data

For this example, we will use a population of 50 individuals with node IDs of 1 through 50. We will consider only two covariates, the overall edge density (a constant covariate of 1) and the Euclidean distance between each pair of individuals. We randomly assign a spatial position on the two-dimensional unit square to each individual.

```
R> library("epinet")
R> set.seed(1)
R> N <- 50
R> mycov <- data.frame(id = 1:N, xpos = runif(N), ypos = runif(N))
R> dyadCov <- BuildX(mycov, binaryCol = list(c(2, 3)),
+   binaryFunc = "euclidean")
```

We then simulate a contact network on this population; we will take the baseline log-odds probability of an edge to be 0, and the incremental log-odds of an edge to decrease by 7 for each unit of distance between the nodes in a dyad. The `SimulateDyadicLinearERGM()` function is used to simulate a network from the model given in Equation 2, given any number of dyadic covariates.

```
R> eta <- c(0, -7)
R> net <- SimulateDyadicLinearERGM(N = N, dyadcovmat = dyadCov, eta = eta)
```

(We note that the `plot` method for ‘network’ objects in the **network** package (Butts 2008) can be used to produce a nice visual representation of a network, if desired.) Next we simulate an epidemic across this network, assuming that the length of time spent in the exposed state follows a gamma distribution. The parameters used for the simulation are $\beta = 1$, $k_I = 3$, $\theta_I = 7$, $k_E = 3$, and $\theta_E = 7$. The `SEIR.simulator()` function returns an ‘epidemic’ object consisting of a matrix with a row for each infected individual and five columns: node ID, node ID of parent (the individual infecting the node), and the times that the individual entered the exposed, infectious, and removed states. For convenience, the time $t = 0$ is chosen to match the time of the first observed recovery.

```
R> epi <- SEIR.simulator(M = net, N = N, beta = 1, ki = 3, thetai = 7,
+   ke = 3, latencydist = "gamma")
R> epi
```

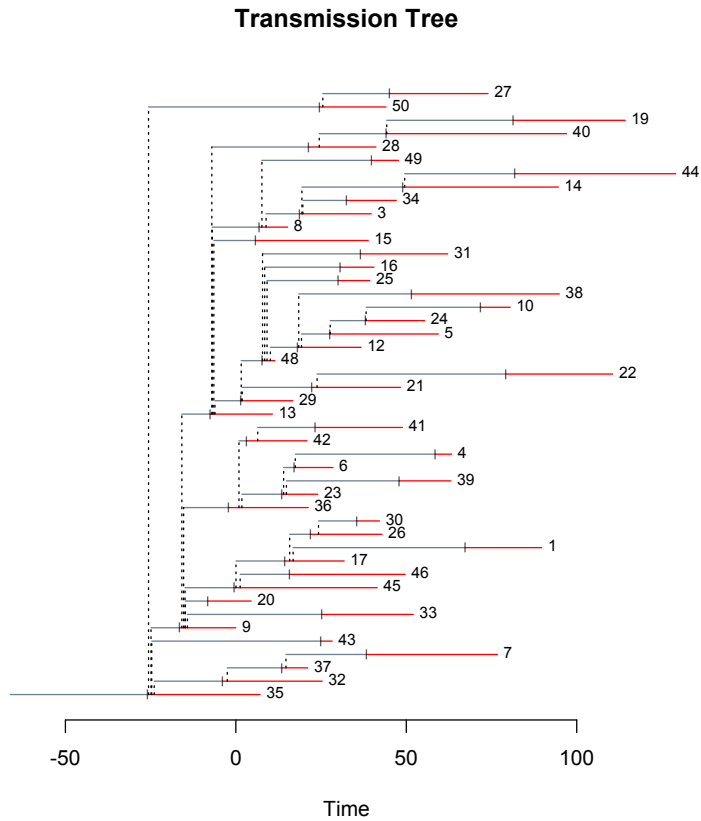


Figure 1: Output from the `plot` method of the `'epidemic'` class showing the spread of the sample epidemic through time. Each horizontal line segment represents the period of time that the individual (labeled to the right of the segment) was in the exposed (gray) and infectious (red) states. The small vertical ticks represent the times of transition from exposed to infectious. Dotted vertical lines indicate transmission events. The time $t = 0$ is chosen to correspond to the first observed recovery.

	Node ID	Parent	Etime	Itime	Rtime
[1,]	35	NA	-66.14114560	-25.8757849	7.099042
[2,]	50	35	-25.61093119	24.5164631	44.023516
[3,]	9	35	-24.88898877	-16.5180717	0.000000
[4,]	43	35	-24.67232216	24.9623163	28.253567
[5,]	32	35	-23.94421137	-3.8943032	25.342128
[6,]	13	9	-15.84157623	-7.5727699	10.718028
.
.
.
[45,]	19	40	44.26119572	81.3977265	114.193368
[46,]	44	14	49.52465905	81.8946425	129.073446
[47,]	2	NA	NA	NA	NA
[48,]	11	NA	NA	NA	NA
[49,]	18	NA	NA	NA	NA
[50,]	47	NA	NA	NA	NA

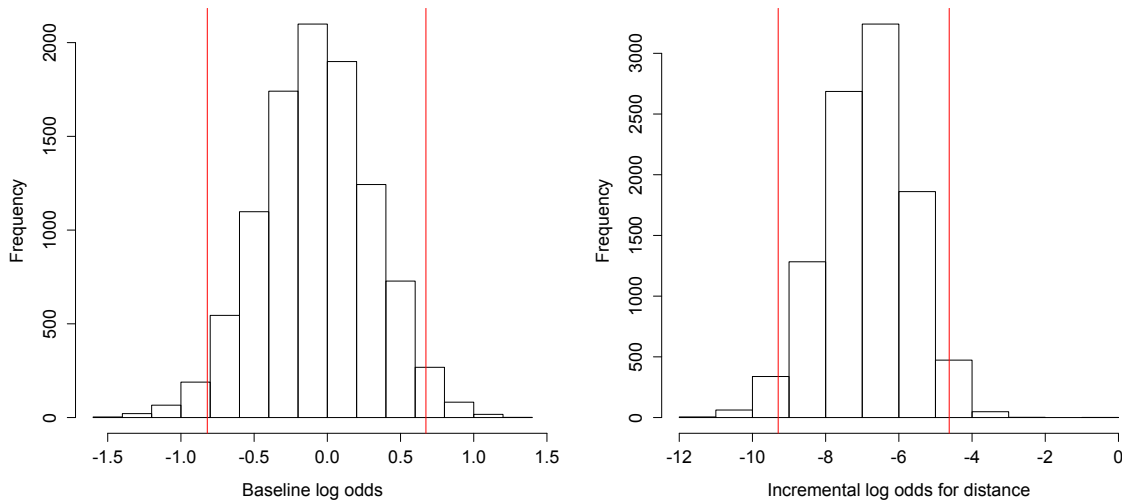


Figure 2: Histograms of the posterior distributions of the η parameters, with associated equitailed 95% posterior credible intervals marked.

From the output, we can see that 46 of the 50 members of the population were ultimately infected during the course of this simulated epidemic; the final four rows in the output represent the members of the population remaining susceptible throughout the epidemic. The `plot` method of the ‘`epidemic`’ class can be used to visualize this epidemic (see Figure 1):

```
R> plot(epi, e.col = "slategrey", i.col = "red")
```

Finally, we can proceed to using the `epinet()` routine to perform inference on the model parameters. We will use uninformative uniform priors for the epidemic parameters, running the algorithm for 1,000,000 iterations, and thinning every 100 to produce a posterior sample of 10,000 points.

```
R> mcmcinput <- MCMCcontrol(nsamp = 1000000, thinning = 100,
+   etapropsd = c(1, 1))
R> priors <- priorcontrol(bprior = c(0, 4), tiprior = c(1, 15),
+   teprior = c(1, 15), etaprior = c(0, 10, 0, 10), kiprior = c(1, 7),
+   keprior = c(1, 7), priordists = "uniform")
R> out <- epinet(~ xpos.ypos.L2Dist, epidata = epi, dyadiccovmat = dyadCov,
+   mcmcinput = mcmcinput, priors = priors)
```

In this example, we run the MCMC algorithm assuming that the exposure and infectious times were known; as a result, inference on the θ_I , k_I , θ_E , and k_E parameters is not interesting. Figure 2 shows histograms of the marginal posterior distributions of the two η parameters. We can see that the algorithm has been able to recover the original parameter values of 0 and -7 .

4.2. An example using the Hagelloch measles data

Next we demonstrate the use of package `epinet` for a situation involving an actual epidemic. Specifically, we consider a data set resulting from a measles epidemic that spread through

Covariate	Description
Household	Indicator variable whose value is 1 if the two individuals are in the same household and 0 otherwise.
Classroom 1	Indicator variable whose value is 1 if the two individuals are both in classroom 1 and 0 otherwise. <code>Classroom 1</code> consists primarily of individuals 6–10 years of age.
Classroom 2	Indicator variable whose value is 1 if the two individuals are both in classroom 2 and 0 otherwise. <code>Classroom 2</code> consists primarily of individuals over the age of 10.
House Distance	Spatial distance (measured in units of 2.5m) between the households of the two individuals.
Male Match	Indicator variable whose value is 1 if the two individuals are both male and 0 otherwise.
Female Match	Indicator variable whose value is 1 if the two individuals are both female and 0 otherwise.
Age Diff	Absolute value of the age difference (in years) of the two individuals.

Table 2: Description of dyadic covariates in `HagellochDyadCov`.

the town of Hagelloch in the winter of 1861. This data was originally collected by Pfeilsticker (1863) and later augmented by Oesterle (1992). This data set has been analyzed in various manners by many researchers, including Lawson and Leimich (2000), Neal and Roberts (2004), Britton, Kypraios, and O’Neill (2011), Groendyke *et al.* (2012) and Meyer *et al.* (2017), among others.

Much of the information contained in the original Hagelloch data set is included in the `epinet` package; it can be accessed using `data("Hagelloch", package = "epinet")` and consists of two components:

- `HagellochTimes` is an array with 187 rows (one for each individual in the population) and 5 columns. The format is similar to that produced by function `SEIR.simulator()`. The columns denote for each individual, respectively, their node ID, putative parent (the individual most likely responsible to have infected the node, as determined by Oesterle 1992), and times entering the exposed, infectious, and removed states. Proxies were used to construct the infectious and removal times as described in Lawson and Leimich (2000). No information regarding the times at which the individuals were exposed was available, so that this column consists entirely of `NA` values. As a result, the exposure times must be inferred as part of the inference process. One outlier was removed from the original data set; this individual did not appear to have been a part of the same epidemic.
- `HagellochDyadCov` is a matrix of dyadic covariates in a format similar to the one used in the previous example, with one row for each dyad. The first two columns consist of the dyad node IDs, while the remainder are described in Table 2.

Figure 3 shows the spread of the epidemic over time, in terms of the number of individuals entering the infectious state each day. From this figure, it appears that the classrooms play a role in the spread of this outbreak. Indeed, Neal and Roberts (2004), Britton *et al.* (2011), and Groendyke *et al.* (2012) all found significant classroom effects in their analyses of this data.

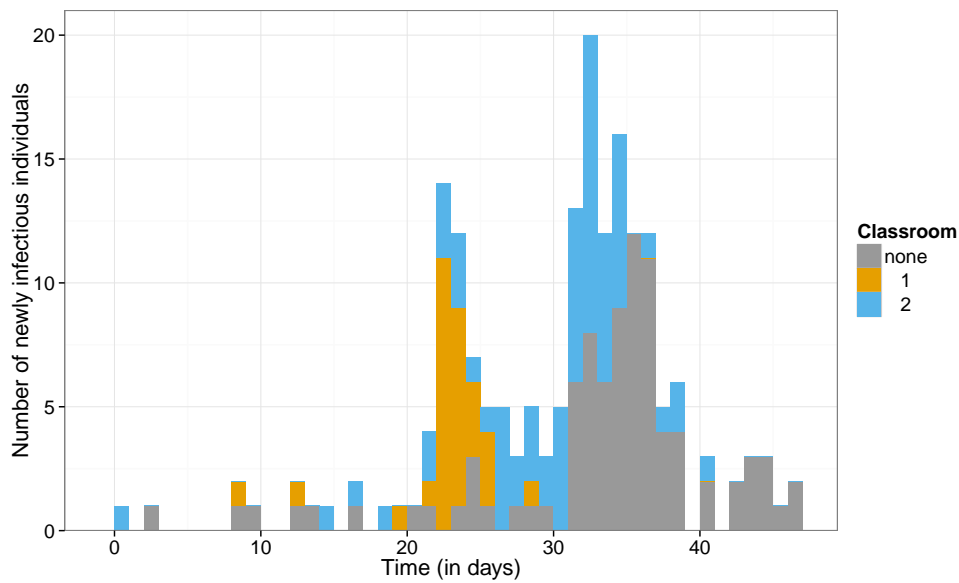


Figure 3: Times (in days) at which individuals entered the infectious state during the Hagelloch measles outbreak, with the individual’s classroom indicated by color. Children aged 6–10 belong to classroom 1, whereas the older children belong to classroom 2. Children under the age of 6 belong to neither classroom. Time 0 represents the time at which the first individual entered the infectious state.

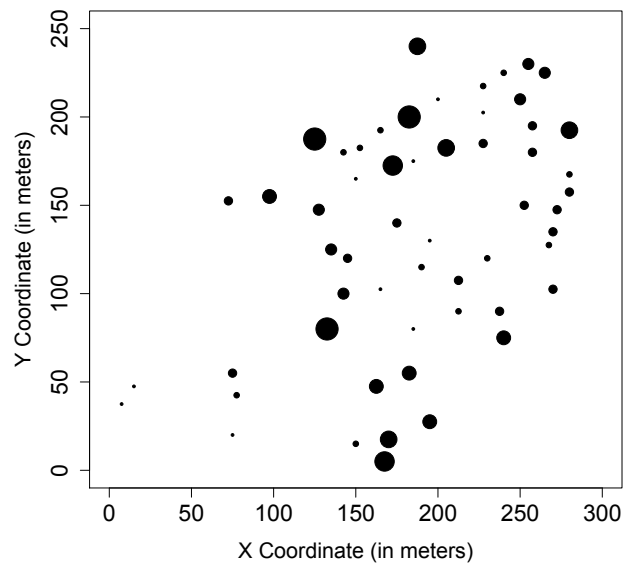


Figure 4: Spatial locations in meters of the households containing individuals infected in the Hagelloch measles outbreak. The size of each circle is proportional to the number of infected individuals in the household.

Figure 4 shows the physical location of the households containing individuals infected during this epidemic. It seems plausible that physical distance between the households of individuals, belonging to the same household, or both, could effect the likelihood of an infectious contact

between these individuals. Work by Neal and Roberts (2004), Britton *et al.* (2011), and Groendyke *et al.* (2012) all indicates that one or both of these factors is likely to be significant. Thus, for the purposes of this example, we utilize the two homophily variables indicating classroom membership (`Classroom 1` and `Classroom 2`), the continuous variable describing the spatial distance between the households of individuals (`House Distance`), and an intercept term. We have found that the MCMC algorithm implemented here can exhibit slow mixing and high levels of autocorrelation. As a result, it is often necessary to run the algorithm for a large number of iterations and to thin the output; this is especially true when the data set contains large numbers of individuals or covariates. Here we run the algorithm for 20,000,000 iterations (after a burn-in of 1,000,000 iterations) and thin every 1,000 samples.

For each of the β , k_E , k_I , θ_E , and θ_I parameters, we use a uniform distribution encompassing the range of biologically plausible values. We assign vague normal prior distributions centered at 0 to the η parameters. In this example, we must infer the exposure times as part of the MCMC algorithm. Following previous analyses, we assume that the entirety of the susceptible population was infected over the course of this epidemic, so that there are no additional individuals in the population beyond those infected individuals for whom we have data.

```
R> mcmcinput <- MCMCcontrol(nsamp = 20000000, thinning = 1000,
+   extrathinning = 10, burnin = 1000000, seed = 1,
+   etapropsd = c(rep(0.05, times = 3), 0.005))
R> priors <- priorcontrol(etaprior = c(0, 3, 0, 3, 0, 3, 0, 3 / 40),
+   bprior = c(0, 4), tiprior = c(0.25, 0.75), teprior = c(0.25, 1),
+   keprior = c(8, 20), kiprior = c(15, 25), priordists = "uniform")
R> out <- epinet(~ `Classroom 1` + `Classroom 2` + `House Distance`,
+   epidata = HaggellochTimes, dyadiccovmat = HaggellochDyadCov,
+   mcmcinput = mcmcinput, priors = priors)
```

Again, we are mainly concerned here with the factors contributing to the likelihood of individuals making infectious contacts; hence we will concentrate on the inference for the network parameters. Trace plots for the posterior samples of these parameters are shown in Figure 5. A first step in checking that the samples are obtained from the actual target posterior distribution is to check that there are no obvious trends in the trace plots of the parameters of interest or log-likelihood values. The trace plots shown in Figure 5 demonstrate good mixing. If there are no obvious trends, an estimate of the number of independent posterior samples in the output can be obtained from the `ess()` function. A long enough run should be performed so that large effective sample sizes (at least 200) for all parameters of interest are obtained. Calculating the effective sample sizes of the network parameter samples yields the following:

```
R> apply(out$eta, 2, ess)
```

(Intercept)	`Classroom 1`	`Classroom 2`	`House Distance`
315.7480	407.3607	2617.4410	347.2327

Likewise, the effective sample sizes for the epidemic parameters are found to be at least 1300. Further convergence diagnostics are available in the R package `coda` (Plummer, Best, Cowles, and Vines 2006) or in tools such as `Tracer` (Rambaut, Suchard, and Drummond 2013); the

Effect (Parameter)	Mean	Median	95% CI
(Intercept) (η_0)	-0.44	-0.36	(-1.44, 0.51)
Classroom 1 (η_1)	7.54	7.44	(4.89, 10.60)
Classroom 2 (η_2)	0.77	0.80	(-0.26, 1.70)
House Distance (η_3)	-0.11	-0.12	(-0.17, -0.06)

Table 3: Summaries of marginal posterior distributions of network parameters for the Hagel-loch measles example, including mean, median, and equitailed 95% posterior credible intervals.

`write.epinet()` function can be used to produce output in tab delimited form. For example, we can use the `geweke.diag()` function in the `coda` package to test for convergence by calculating the Geweke diagnostic statistics for the various model parameters (Geweke 1992).

```
R> library("coda")
R> geweke.diag(as.mcmc(out$eta))
```

```
Fraction in 1st window = 0.1
Fraction in 2nd window = 0.5
```

```
(Intercept)      `Classroom 1`      `Classroom 2`      `House Distance`
      0.8858              0.1908              0.2872              -0.5660
```

Under this test, if the MCMC chain has converged, the value of the test statistic for each parameter has a standard normal distribution. We can similarly compute the test statistics corresponding to the epidemic parameters; they range from -1.306 to 1.225 . None of these test statistics is large enough in absolute value to cause us to question the convergence of the chain.

Being satisfied with the convergence of the Markov chain, we then proceed to analyzing the output. Table 3 gives summaries of the estimated marginal posterior distributions of the network parameters.

The signs of the parameters are all consistent with our intuition about the corresponding effects. The two classroom parameters both have positive means, indicating that sharing a classroom increases the likelihood that two individuals will have an infectious contact. The magnitude of the effect for classroom 1 is much larger than that of classroom 2; this is consistent with results found by Neal and Roberts (2004) and Groendyke *et al.* (2012). The posterior distribution of the household distance parameter is almost entirely negative, indicating that the greater the distance between the households of two individuals, the less likely they are to make an infectious contact.

We can use the values in Table 3 to estimate the posterior probability of a contact between two individuals. For example, consider a pair of individuals i and j who are both in classroom 1 and whose households are 75m apart. For such a dyad, we would have $\mathbf{X}_{\{i,j\},0} = 1$, $\mathbf{X}_{\{i,j\},1} = 1$, $\mathbf{X}_{\{i,j\},2} = 0$, and $\mathbf{X}_{\{i,j\},3} = 30$. Then we would estimate the log-odds of this dyad having a contact as $-0.44 + 1(7.54) + 0(0.77) + 30(-0.11) = 3.80$ so that the probability of a contact is estimated to be $e^{3.80} / (e^{3.80} + 1) = 0.98$.

Due to the size of this epidemic, using the `plot` method of the `'epinet'` class to display the inferred transmission trees on the screen is not ideal, as the output becomes very difficult

Trace plots of network parameters

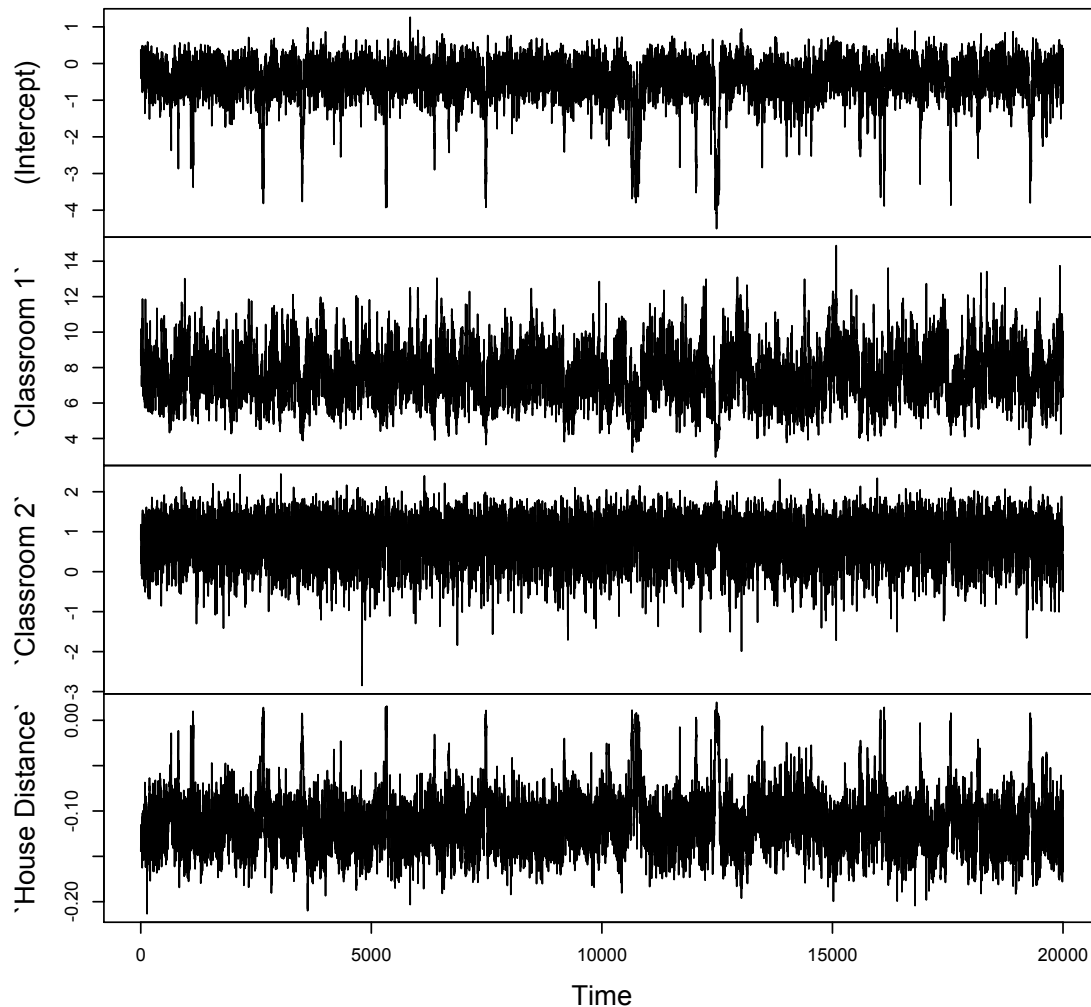


Figure 5: Trace plot for network (η) parameters in the Hagelloch measles example. The Markov chain had a length of 20,000,000 and was thinned every 1,000 iterations to produce a final sample size of 20,000 for each parameter.

to read. However, the package `epinet` provides a couple of options for viewing the posterior samples of the transmission trees. First, we can use `pdf()` with this plot method in order to write the trees to a PDF file:

```
R> pdf(file = "SampleHagellochTree.pdf", height = 20, width = 7)
R> plot(out, index = 100)
R> dev.off()
```

A second option is to use the `epi2newick()` or `write.epinet()` function to write the transmission trees to the text-based Newick format (Felsenstein 2004), which is the standard tree format used in phylogenetics, that can then be plotted using any of a number of packages such as `ape` (Paradis, Claude, and Strimmer 2004) in R, `FigTree` (Rambaut 2014), or `IcyTree`

(Vaughan 2017). The `epi2newickmcmc()` function returns a single inferred transmission tree as a Newick formatted string. The `write.epinet()` function writes inference output into two files. The first file is a `.log` file containing the posterior samples of the epidemic parameters as a tab delimited file which can be read into, for example, **Tracer** (Rambaut *et al.* 2013) to compute and visualize various diagnostic information. The second file (which is created only if the transmission tree was inferred as part of the inference procedure) is a `.trees` file which contains all of the inferred transmission trees in Newick format. To write the posterior samples for the Hagelloch example, we can use the following code:

```
R> write.epinet(out, "HagellochOutput")
```

5. Conclusion and extensions

We have presented here the R package **epinet**, a collection of tools that helps the researcher in the analysis of epidemic outbreaks. After describing the inference methodology, models and assumptions used, we have discussed the simulation, plotting, and inference capabilities of this software package, and demonstrated the use of these functions through the analysis of both simulated and actual data. We believe that this software package can be of significant value to statisticians, epidemiologists, biologists, and other researchers wishing to study the spread of diseases through populations.

This software package has a couple of important limitations that may inhibit the use of this methodology in some cases. First, as mentioned above, obtaining complete and robust data of the type used here (i.e., the times that the hosts entered the various disease states) is often very challenging to obtain and frequently requires the researcher to make assumptions or use proxies for these times. The second caveat regards the size of the outbreaks that can be analyzed with this software. The time required by the inference routine `epinet()` grows with the square of the size of the number of infected individuals in the population. In addition, we have found that the MCMC algorithm used in this routine tends to exhibit poor mixing characteristics, thus requiring the algorithm to be run for a large number of iterations. These two factors necessarily limit the outbreaks that can be effectively analyzed to those with perhaps several hundred to a few thousand infected individuals, though this limit will greatly depend on the specific data and computing resources available.

There are a number of ways in which this software package and modeling approach can be developed that will enable them to be applied to a broader class of data sets. We suggest four here. First, we would like to be able to deal with incomplete sampling of the epidemic. This can be achieved via imputation within the MCMC algorithm when the number of unknown infections is small but we expect this approach would fail to mix when a large fraction of the infections were unsampled; in this case another, perhaps simpler, model needs to be applied to the missing data. Second, the current model considers each vertex as an individual. An obvious extension is to allow the nodes to be larger units such as households, farms, cities, or districts in which we model the separate outbreaks and connect them using the network, which becomes a meta-population model. Given an appropriate within-population model, we see no reason this approach would not succeed. A third challenge is to drop the assumption that the ERGM is dyadic-independent. We know that dyadic-dependent models are a better fit for many social networks, but are currently prevented from fitting them due to the difficulty in

calculating the likelihood – a routine that is repeated millions of times during a typical run of the MCMC algorithm in package **epinet**. As research develops for ERGM estimation, it could be integrated into the **epinet** framework. Finally, we reiterate the desire to apply these models to genetic data, a challenge mentioned in Groendyke *et al.* (2011). Given the complexity of standard phylogenetic models, it may make more sense to port the **epinet** framework to an existing phylogenetic framework rather than vice-versa to tackle this challenge.

Acknowledgments

The authors thank David Hunter who gave generous guidance and support during the construction of the **epinet** package. The authors also thank two anonymous reviewers for their helpful suggestions and comments, which greatly improved the quality of both the software package and this manuscript. We would also like to thank Peter Neal for providing the Hagelloch measles data discussed in Section 4.2. This work was supported by NIH grant R01-GM083603-01.

References

- Badham J, Stocker R (2010). “The Impact of Network Clustering and Assortativity on Epidemic Behaviour.” *Theoretical Population Biology*, **77**(1), 71–75. doi:10.1016/j.tpb.2009.11.003.
- Bailey NTJ (1957). *The Mathematical Theory of Epidemics*. Griffin, London.
- Boelle PY, Obadia T (2015). **R0**: *Estimation of R0 and Real-Time Reproduction Number from Epidemics*. R package version 1.2-6, URL <https://CRAN.R-project.org/package=R0>.
- Britton T, Kypraios T, O’Neill PD (2011). “Inference for Epidemics with Three Levels of Mixing: Methodology and Application to a Measles Outbreak.” *Scandinavian Journal of Statistics*, **38**(3), 578–599. doi:10.1111/j.1467-9469.2010.00726.x.
- Britton T, O’Neill P (2002). “Bayesian Inference for Stochastic Epidemics in Populations with Random Social Structure.” *Scandinavian Journal of Statistics*, **29**(3), 375–390. doi:10.1111/1467-9469.00296.
- Butts CT (2008). “**network**: A Package for Managing Relational Data in R.” *Journal of Statistical Software*, **24**(2), 1–36. doi:10.18637/jss.v024.i02.
- Butts CT (2015). **network**: *Classes for Relational Data*. R package version 1.13.0, URL <https://CRAN.R-project.org/package=network>.
- Caimo A, Friel N (2014). “**Bergm**: Bayesian Exponential Random Graphs in R.” *Journal of Statistical Software*, **61**(2), 1–25. doi:10.18637/jss.v061.i02.
- Cauchemez S, Bhattarai A, Marchbanks TL, Fagan RP, Ostroff S, Ferguson NM, Swerdlow D, Sodha SV, Moll ME, Angulo FJ, others (2011). “Role of Social Networks in Shaping Disease Transmission During a Community Outbreak of 2009 H1N1 Pandemic Influenza.” *Proceedings of the National Academy of Sciences of the United States of America*, **108**(7), 2825–2830. doi:10.1073/pnas.1008895108.

- Chongsuvivatwong V (2012). **epicalc**: *Epidemiological Calculator*. R package version 2.15.1.0, URL <https://CRAN.R-project.org/package=epicalc>.
- Cori A (2013). **EpiEstim**: *A Package to Estimate Time Varying Reproduction Numbers from Epidemic Curves*. R package version 1.1-2, URL <https://CRAN.R-project.org/package=EpiEstim>.
- Danon L, Ford AP, House T, Jewell CP, Keeling MJ, Roberts GO, Ross JV, Vernon MC (2011). “Networks and the Epidemiology of Infectious Disease.” *Interdisciplinary Perspectives on Infectious Diseases*, **2011**.
- Erdős P, Rényi A (1959). “On Random Graphs.” *Publicationes Mathematicae*, **6**, 290–297.
- Felsenstein J (2004). *Inferring Phylogenies*. Sinauer Associates Sunderland.
- Frank O, Strauss D (1986). “Markov Graphs.” *Journal of the American Statistical Association*, **81**(395), 832–842. doi:10.1080/01621459.1986.10478342.
- Geweke J (1992). “Evaluating the Accuracy of Sampling-Based Approaches to the Calculation of Posterior Moments.” In *Bayesian Statistics 4*, pp. 169–193. University Press.
- Gilbert EN (1959). “Random Graphs.” *The Annals of Mathematical Statistics*, **30**(4), 1141–1144. doi:10.1214/aoms/1177706098.
- Groendyke C, Welch D (2018). **epinet**: *Epidemic/Network-Related Tools*. R package version 2.1.8, URL <https://CRAN.R-project.org/package=epinet>.
- Groendyke C, Welch D, Hunter DR (2011). “Bayesian Inference for Contact Networks Given Epidemic Data.” *Scandinavian Journal of Statistics*, **38**(3), 600–616. doi:10.1111/j.1467-9469.2010.00721.x.
- Groendyke C, Welch D, Hunter DR (2012). “A Network-Based Analysis of the 1861 Hagelloch Measles Data.” *Biometrics*, **68**(3), 755–765. doi:10.1111/j.1541-0420.2012.01748.x.
- Handcock MS, Hunter DR, Butts CT, Goodreau SM, Krivitsky PN, Morris M (2017). **ergm**: *Fit, Simulate and Diagnose Exponential-Family Models for Networks*. The Statnet Project. R package version 3.8.0, URL <https://CRAN.R-project.org/package=ergm>.
- Handcock MS, Hunter DR, Butts CT, Goodreau SM, Morris M (2008). “statnet: Software Tools for the Representation, Visualization, Analysis and Simulation of Network Data.” *Journal of Statistical Software*, **24**(1), 1–11. doi:10.18637/jss.v024.i01.
- Hunter DR, Handcock MS, Butts CT, Goodreau SM, Morris M (2008). “ergm: A Package to Fit, Simulate and Diagnose Exponential-Family Models for Networks.” *Journal of Statistical Software*, **24**(3), 1–29. doi:10.18637/jss.v024.i03.
- Keeling MJ, Rohani P (2007). *Modeling Infectious Diseases in Humans and Animals*. Princeton University Press.
- Kenah E, Robins JM (2007). “Second Look at the Spread of Epidemics on Networks.” *Physical Review E*, **76**(3), 036113. doi:10.1103/physreve.76.036113.

- Kernighan BW, Ritchie DM, Ejkliint P (1988). *The C Programming Language*, volume 2. Prentice-Hall Englewood Cliffs.
- Lawson AB, Leimich P (2000). “Approaches to the Space-Time Modelling of Infectious Disease Behaviour.” *Mathematical Medicine and Biology*, **17**(1), 1–13. doi:10.1093/imammb/17.1.1.
- Melnik S, Hackett A, Porter MA, Mucha PJ, Gleeson JP (2011). “The Unreasonable Effectiveness of Tree-Based Theory for Networks with Clustering.” *Physical Review E*, **83**, 036112. doi:10.1103/physreve.83.036112.
- Meyer S, Held L, Höhle M (2017). “Spatio-Temporal Analysis of Epidemic Phenomena Using the R Package **surveillance**.” *Journal of Statistical Software*, **77**(11), 1–55. doi:10.18637/jss.v077.i11.
- Moslonka-Lefebvre M, Pautasso M, Jeger MJ (2009). “Disease Spread in Small-Size Directed Networks: Epidemic Threshold, Correlation Between Links to and from Nodes, and Clustering.” *Journal of Theoretical Biology*, **260**(3), 402–411. doi:10.1016/j.jtbi.2009.06.015.
- Neal PJ, Roberts GO (2004). “Statistical Inference and Model Selection for the 1861 Hagelloch Measles Epidemic.” *Biostatistics*, **5**(2), 249–261. doi:10.1093/biostatistics/5.2.249.
- Oesterle H (1992). *Statistische Reanalyse einer Masernepidemie 1861 in Hagelloch*. M.D. Thesis, Eberhard-Karls Universität, Tübingen.
- Paradis E, Claude J, Strimmer K (2004). “**ape**: Analyses of Phylogenetics and Evolution in R Language.” *Bioinformatics*, **20**(2), 289–290. doi:10.1093/bioinformatics/btg412.
- Pfeilsticker A (1863). *Beiträge zur Pathologie der Masern mit besonderer Berücksichtigung der statistischen Verhältnisse*. M.D. Thesis, Eberhard-Karls Universität, Tübingen.
- Plummer M, Best N, Cowles K, Vines K (2006). “**coda**: Convergence Diagnosis and Output Analysis for MCMC.” *R News*, **6**(1), 7–11.
- Rambaut A (2014). “**FigTree** V1.4.” URL <http://tree.bio.ed.ac.uk/software/figtree/>.
- Rambaut A, Suchard M, Drummond A (2013). “**Tracer** V1.6.” URL <http://tree.bio.ed.ac.uk/software/tracer/>.
- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Schweinberger M, Handcock M, Luna P (2017). **hergm**: *Hierarchical Exponential-Family Random Graph Models with Local Dependence*. R package version 3.1-0, URL <https://CRAN.R-project.org/package=hergm>.
- Vaughan TG (2017). “**IcyTree**: Rapid Browser-Based Visualization for Phylogenetic Trees and Networks.” *Bioinformatics*, **33**(15), 2392–2394. doi:10.1093/bioinformatics/btx155.
- Wasserman S (1994). *Social Network Analysis: Methods and Applications*, volume 8. Cambridge University Press.

Welch D, Bansal S, Hunter DR (2011). “Statistical Inference to Advance Network Models in Epidemiology.” *Epidemics*, **3**(1), 38–45. doi:10.1016/j.epidem.2011.01.002.

Affiliation:

Chris Groendyke
Department of Mathematics
Robert Morris University
6001 University Blvd.
Moon Township, PA 15108, United States of America
E-mail: groendyke@rmu.edu

David Welch
Department of Computer Science
University of Auckland
Private Bag 92019
Auckland 1142, New Zealand
E-mail: david.welch@auckland.ac.nz