



hybridModels: An R Package for the Stochastic Simulation of Disease Spreading in Dynamic Networks

Fernando S. Marques
Universidade de São Paulo

José H. H. Grisi-Filho
Universidade de São Paulo

Marcos Amaku
Universidade de São Paulo

Jean C. R. Silva
Universidade Federal Rural
de Pernambuco

Erivânia C. Almeida
Agência de Defesa e
Fiscalização Agropecuária
de Pernambuco

José L. Silva Júnior
Agência de Defesa e
Fiscalização Agropecuária
de Pernambuco

Abstract

Disease spreading simulations are traditionally performed using coupled differential equations. However, in the setting of metapopulations, most of the solutions provided by this method do not account for the dynamic topography of subpopulations. Conversely, the alternative approach of individual-based modeling (IBM) may add computational cost and complexity.

Hybrid models allow for the study of disease spreading because they combine both aforementioned approaches by separating them across different scales: a local scale that addresses subpopulation dynamics using coupled differential equations and a global scale that addresses the contact between these subpopulations using IBM.

We present a simple way of simulating the spread of disease in dynamic networks using the high-level statistical computational language R and the **hybridModels** package. We built four examples using disease spread models at the local scale in several different networks: an animal movement network; a three-node network, whose model solution using a stochastic simulation algorithm is compared with the ordinary differential equations approach; the commuting of individuals between patches, which we compare with the permanent migration of individuals; and the commuting of individuals within the metropolitan area of São Paulo.

Keywords: modeling, dynamic networks, epidemic, stochastic simulation, R.

1. Introduction

Coupled differential equation models are traditionally used to study disease spreading. These models carry assumptions that apply to scenarios with large populations whose individuals are well mixed. Despite being effective even in complex scenarios (Keeling 2005), coupled differential equation models do not describe sufficiently well the behavior of all systems.

A population is often spatially divided into subpopulations (metapopulation), and the communication between these subpopulations occurs irregularly, such as in animal movement networks (Dubé, Ribble, Kelton, and McNab 2008; Bigras-Poulin, Thompson, Chriel, Mortensen, and Greiner 2006; Grisi-Filho *et al.* 2013; Negreiros 2010) in which animals migrate irregularly from one subpopulation to another in time and space, thus forming a dynamic movement network. The usual terminology for migration in veterinary epidemiology is “movement”. Thus, when this paper refers to the migration of animals from farm A to farm B, we will use the term “movement”. In addition, the term “metapopulation” denotes a population that lives across a fragmented landscape of potential habitat patches. Therefore, this paper considers the animal movement network as a rather special case of a metapopulation model.

A dynamic movement network (i.e., a spatial network in which individuals migrate along the edges at discrete time points) can violate the assumptions of coupled differential equation models, which restricts the use of the model. In cases where the model treats the metapopulation as a single population, the assumption of homogeneity of contacts among individuals may be violated because the connections between subpopulations may take a topology that is far from the homogeneous mixing of individuals (Bigras-Poulin *et al.* 2006; Kao, Green, Johnson, and Kiss 2007; Bigras-Poulin, Barfod, Mortensen, and Greiner 2007; Rautureau, Dufour, and Durand 2011; Büttner, Krieter, and Traulsen 2015). In cases where the model describes subpopulations individually, the equations may not capture the dynamic behavior of migrations of individuals from one subpopulation to another.

There is an alternative approach able to address these situations known as the individual-based model (IBM). This class of models uses a bottom-up strategy and starts by creating rules for the behavior of individuals, and based on the interaction among these individuals, it attempts to understand the properties that emerge from the system (Grimm and Railsback 2005; Grimm *et al.* 2005). An IBM has a more flexible construction, but depending on the degree of granularity desired for the model, its complexity increases and its computational performance can be compromised; the number of parameters, the difficulty of model construction and communication, and analysis of results all increase (Osgood 2007; Rahmandad and Sterman 2008; Vincenot, Giannino, Rietkerk, Moriya, and Mazzoleni 2011).

The response to the difficulties found in both approaches came in the form of a hybrid model, which combines an IBM with coupled differential equations. This combination balances the complexity of the former with the assumptions of the latter. In this approach, the subpopulations are treated as a homogeneous mixing of individuals, and the migrations between the subpopulations are treated by the IBM. In other words, using the terminology employed by Vincenot and Moriya (2011), there is a local scale in which the assumptions used in the formulation of coupled differential equations are employed, and there is a global scale that is managed by the IBM. The local scale is a submodel within the IBM structure, making the IBM a hybrid model.

Nevertheless, both scales can be treated stochastically. At the local scale, this stochasticity is a model requirement in many situations. Subpopulations may contain a few individuals that

violate the assumption that the situations generated by coupled differential equations are good approximations of the average behavior of large populations. In these cases, the stochasticity and the discrete aspect of the population may have an important role in the evolution of the system. In models described by coupled ordinary differential equations, each process (e.g., infection and recovery) has a rate. This approach assumes that the time evolution is both continuous and deterministic and that the number of individuals can change by fractions for each time interval. However, this is not the case, as population size can change only by discrete integers (Gillespie 1977). Alternatively, instead of focusing on a process rate, one can develop models centered on individual reaction events, such as migration, that do not necessarily occur at regular intervals. The timing of these events is stochastic rather than deterministic (Monk 2012).

For these reasons, the R (R Core Team 2020) package **hybridModels** (Marques, Grisi-Filho, and Amaku 2020) does not address the local scale using coupled differential equations, but rather, it uses the stochastic simulation algorithm or its approximations (Gillespie 1976, 1977, 2001; Chatterjee, Vlachos, and Katsoulakis 2005; Cao, Gillespie, and Petzold 2006; Gillespie 2007) with the help of the **GillespieSSA** package (Pineda-Krch 2008), which considers the stochasticity and the discrete aspect of the system. The Gillespie stochastic simulation algorithm (SSA) is a procedure for generating the time-evolution trajectories of finite populations over continuous time and has become the standard algorithm for these types of stochastic models (Pineda-Krch 2008). The discrete aspect of the stochastic simulation also agrees with the discrete aspect of the number of individuals that migrate from one subpopulation to another, and this does not require approximations that may influence the evolution of the epidemic.

The purpose of the **hybridModels** package is to allow the user to implement simulations of infectious disease spreading in dynamic networks while considering the local (within the subpopulation) and global (among subpopulations) scales. In the **hybridModels** package, it is possible to implement models that consider the permanent migration of individuals or the commuting of individuals between nodes. Package **hybridModels** is available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=hybridModels>. In Section 3, we show four different examples of applications of this package: a susceptible-infected-recovered (SIR) model applied to the dynamics of an animal movement network; a susceptible-infected (SI) model in a three-node network, whose solution using a stochastic simulation algorithm is compared with the ordinary differential equations approach; commuting of individuals between nodes using a susceptible-infected-susceptible (SIS) model, which we compare with a model of permanent individual migration; and disease spread accounting for commuting of individuals within São Paulo.

2. Methodology

We will describe how the models are implemented by closely following the overview, design concepts, and details (ODD) protocol (Grimm *et al.* 2006; Grimm, Berger, DeAngelis, Polhill, Giske, and Railsback 2010). One of the purposes of the ODD protocol is to clearly communicate the features of an IBM, thus making it reproducible. Here, the protocol will be used to describe the implementation of the hybrid models of the **hybridModels** package, which are customizable at the local scale; i.e., they allow the creation of propensity functions and other objects (Gillespie 2007) that describe the disease spread dynamics within subpopulations. A

propensity function is related to the probability of a certain event (e.g., infection or recovery) occurring in an infinitesimal time interval, and it can be given by rates of individuals transferred among the states of the system.

2.1. Purpose

From a given initial condition and under the structure of functions that describe disease spreading in subpopulations, the models aim to study the progression of an epidemic in a dynamic migration network of individuals, in which there is no individual identification or in which it is not of interest.

2.2. Entities, state variables, and scales

The subpopulation is the model entity. It is described by state variables that represent the number of individuals in each compartment of the subpopulation. For example, in a disease model, the state variable S refers to the number of susceptible individuals and the state variable I refers to the number of infected individuals.

These variables describe the subpopulation state at the local scale. With migration between the subpopulations, state variables influence the state of other entities, producing the global behavior. When individuals in a given state migrate, they are subtracted from the number of individuals in that state in the origin subpopulation and added to the destination subpopulation in the corresponding state. Each subpopulation corresponds to a node in the movement network and may have its own behavior with regards to the migration of individuals, including either permanent migration or commuting.

2.3. Process overview and routine

Both scales (local and global) implemented in this package are executed alternately (Figure 1). The input data (Section 2.6) contain the number of individuals that migrated from an origin to a destination subpopulation, and the time when the migration occurred. After setting the initial condition of the simulation, the state variables of each subpopulation are updated with the data on the migration of individuals (i.e., the migration events occur instantaneously at the points of time specified in the input data).

Subsequently, based on the stochastic simulation model chosen (Pineda-Krch 2008), the simulation is performed for the local scale submodels that address disease spreading within the subpopulations. The disease spread within a node occurs between migration events. Once the local scale simulation is concluded, the state variables of each subpopulation are updated again with migration data, following the chronological order in which they appear. The simulation of local dynamics and the migration process are performed alternately. The time interval used in each step of the local scale is the interval between the migrations of individuals. Local dynamics stop in all subpopulations when a migration occurs, no matter which subpopulation is undergoing migration.

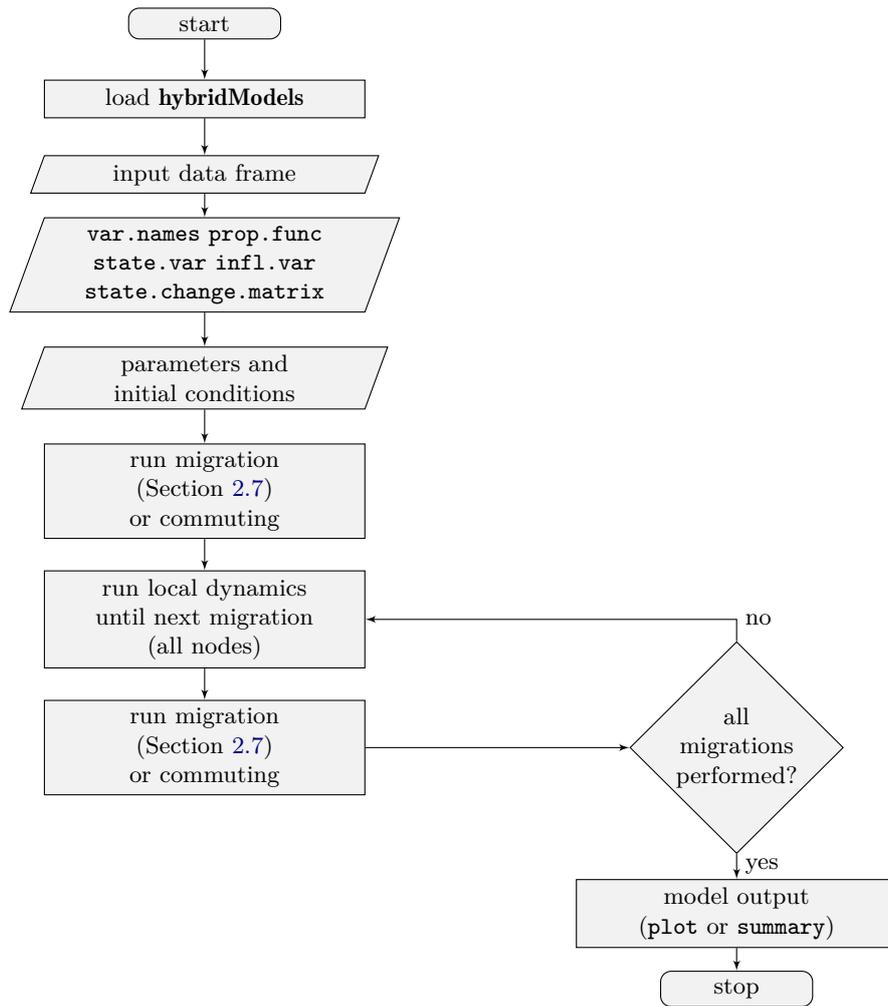


Figure 1: Flow chart of the sequential steps used to perform the models in the **hybridModels** package.

2.4. Design concepts

Basic principles

The principles guiding hybrid models are that the topology of the dynamic individual movement network influences disease spreading (Shirley and Rushton 2005; Vincenot and Moriya 2011) and, in addition, subpopulations allow the use of equations that assume a homogeneous mixing of contact between individuals.

Emergence

Spatiotemporal disease spread emerges from interacting local population and infection dynamics within subpopulations as well as the migration of (infected) individuals among subpopulations. Both the network topology and the interaction of the topology with the spread dynamics within subpopulations are important for disease spread behavior.

Interaction

The interaction between subpopulations occurs with the migration of individuals that change the state variables of each subpopulation, both in subpopulations that had immigrations and in those that had emigrations. We assume that the migration of individuals is an instantaneous process. The mortality of individuals during migration is not considered.

Stochasticity

Stochasticity is present at the global and local scales. At the local scale, stochasticity is defined by the numerical simulation method chosen (Pineda-Krch 2008). At the global scale, stochasticity is present in the random selection of individuals that are going to migrate. The migration process is described in Section 2.7.

Observation

It is possible to choose the number of simulations to be performed. For each simulation, the state variables of each subpopulation are stored in a data frame at the end of each time period in which individual migrations are recorded. The state variables are those of the epidemic model that describes the local dynamics. In a SIR model, for instance, the state variables represent the number of susceptible, infected, and recovered individuals. The data frame and other information relevant for the model (for example, state variables, parameters, and initial conditions) are stored in an S3 class object ‘HM’ by the **hybridModels** package.

2.5. Initialization

Initial conditions are specified in a named vector indicating the initial state of the subpopulations’ state variables. To demonstrate how to provide the initial conditions, let us suppose that we have only three subpopulations, each of which has 200 susceptible individuals (state variable S). Additionally, there is one infected individual (state variable I) in subpopulation number 2. To set the initial conditions, it is necessary to indicate the state variable followed by the number assigned to the subpopulation and, after an “equal” sign, the number of individuals in that state. In this particular case, the initial conditions would be given by:

```
R> init.cond <- c(S1 = 200, S2 = 200, S3 = 200, I2 = 1)
```

The model uses this information to start the simulations, according to the description given in Section 2.3.

2.6. Input data

The input data for models simulated by this package are the connections between subpopulations (migrations or commutings), which are described in a data frame in which each row contains the date of the movement, the origin, the destination, and the number of individuals moving between just these two subpopulations.

2.7. Submodels

There are three separate submodels: a submodel for local population dynamics, a submodel for migration between subpopulations, and a submodel for commuting between subpopulations.

Submodel for local subpopulation dynamics

The submodel for local subpopulation dynamics is determined by the following objects:

- `prop.func`, a vector with the propensity functions that describe changes of state variables due to local population and disease dynamics. The propensity functions are given by rates of individuals transferred from one state to another. For instance, in a SIR model without vital rates, the propensity functions are the transmission term (i.e., incidence), related to the transition between the compartments of susceptible and infected individuals, and the recovery rate, which is related to the transition between the compartments of infected and recovered individuals. Examples of propensity functions are given in Section 3.
- `state.change.matrix`, a state change matrix that describes how propensity functions influence the state variables. The state change matrix values indicate whether a given propensity function does (1 or -1) or does not (0) change the corresponding state variable.
- `model.parms`, model parameters such as contact, recovery, and immunity waning rates.
- `state.var`, a vector with state variables. For instance, in a SIR model, the state variables are the number of susceptible, infected, and recovered individuals.
- `infl.var`, a vector with state variables of commuting individuals. This should only be used in the commuting submodel (explained above).

Models for the infection dynamics at the local scale can be constructed using these objects. Examples of these objects can be found in Section 3. In addition to the objects described above, the submodel also comprises the implementation of the migration process described below.

Submodel for migration between subpopulations

Individuals that migrate are sampled randomly from a node using the `sample` function without replacement, which returns a vector composed of randomly chosen individuals. These individuals are distributed among the destination nodes following the order of the nodes appearance in the input data frame. Below, we provide an illustrative example of the migration process.

1. Let us suppose that, at a specific time, nine individuals migrate from node A to nodes X , Y and Z , according to data from the input data frame described in Section 2.6. Let us also suppose that four susceptible (S), three infected (I), and two recovered (R) individuals are randomly sampled to migrate using the `sample` function without replacement, resulting in the following vector $[S, S, R, I, I, R, I, S, S]$.
2. In addition, according to the input data frame, three individuals migrate to node X , four individuals to node Y , and two individuals to node Z . If the order of the destination nodes in the input data frame is, for instance, Y , X , and Z , then the four first individuals in the migration vector ($[S, S, R, I]$) migrate to node Y , the following three individuals ($[I, R, I]$) migrate to node X , and the remaining two individuals ($[S, S]$) migrate to node Z .

| Variable | Description |
|---------------|--------------------------------------|
| Day | Day when the animal movement occurs. |
| originID | Premises of origin. |
| destinationID | Destination premises. |
| num.animals | Number of animals moved. |

Table 1: Variables in the `networkSample` data frame.*Submodel for commuting between subpopulations*

In this submodel, moving individuals contribute to the spread of disease on the target node, but the demography on the origin and target nodes are unaffected. All subpopulations are considered to have the same individuals over time. This is a useful approach for scenarios in which there is contact between subpopulations, but individuals do not migrate in a definitive manner, such as people commuting to work and home on a daily basis. This may also be applied to scenarios in which it is hard to keep track of all individuals, but the subpopulations are considered stable over time. If the user provides a `infl.var` parameter, the package automatically uses the commuting model.

The commuting process, including sampling of individuals and vector's arrangement, is similar to the migration process explained above, with the only difference being that all commuting individuals will remain at the origin node, but they will contribute to the local dynamics at the destination nodes through the `infl.var` parameter.

3. Examples

3.1. The cattle trade network

To illustrate the configuration and execution of hybrid models using the high-level statistical computational language R and the `hybridModels` package, we will use a SIR model applied to a small fragment of the real dynamic cattle trade network of the state of Pernambuco, Brazil, extracted from the animal movement network provided by the local veterinary agency (Agência de Defesa e Fiscalização Agropecuária de Pernambuco, ADAGRO-PE).

To simulate the model with the animal trade network, the package and its database must be loaded using the codes below:

```
R> library("hybridModels")
R> data("networkSample", package = "hybridModels")
```

The `networkSample` data frame (Table 1) describes the movement of cattle between the premises. The cattle trade network fragment (Figure 2) has 43 premises. In 2012 and 2013, 680 animals were moved across these premises in 78 batches. The distribution of the number of animals per batch based on the variable `num.animals` in the `networkSample` data frame is shown in Figure 3.

For each of the premises, we calculate the following network parameters to characterize the network: ingoing and outgoing contact chain using the dynamic movement network.

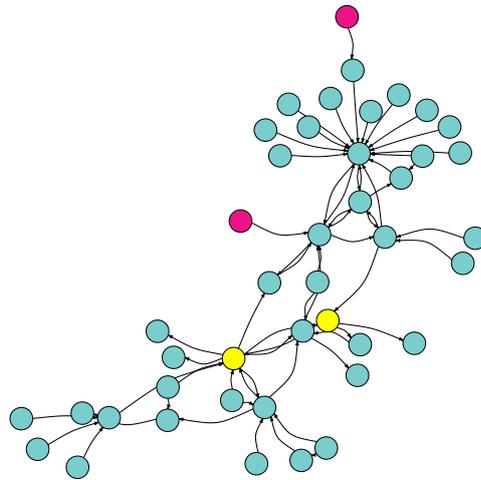


Figure 2: The circles represent the premises, and the arrows represent the direction of cattle movement. The network is dynamic, but the movements were aggregated to generate this visualization. In the examples below, premises in pink correspond to the initial infection cases of condition 1, and premises in yellow correspond to condition 2.

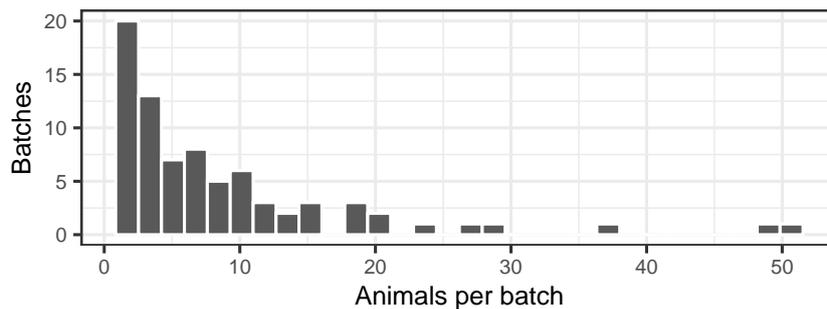


Figure 3: Histogram of the number of animals moved in each batch.

The outgoing contact chain is defined as the number of farms that can be reached through the movement of animals from a given farm, either directly or indirectly (Nöremark and Widgren 2014). The calculation of this measure considers that the network is dynamic, i.e., the temporal sequence of the movements is respected. Thus, the outgoing contact chain can be used for forward tracing because it is composed of all the farms that can be reached directly or indirectly from the source farm in a given time window.

The ingoing contact chain is defined as the number of farms that can reach, directly or indirectly, a given farm through the movement of animals, also respecting the temporal order of movements. Thus, the ingoing contact chain can be used for backward tracing. A schematic illustration of both the outgoing and ingoing contact chains can be found in Nöremark and Widgren (2014).

In general, these parameters contribute to the identification of farms that are most likely to contract or spread infectious diseases, and in the case of the contact chain, they also allow the estimation of the maximum size of the epidemic (Nöremark, Håkansson, Lewerin, Lindberg, and Jonsson 2011; Frössling, Ohlson, Björkman, Håkansson, and Nöremark 2012; Büttner

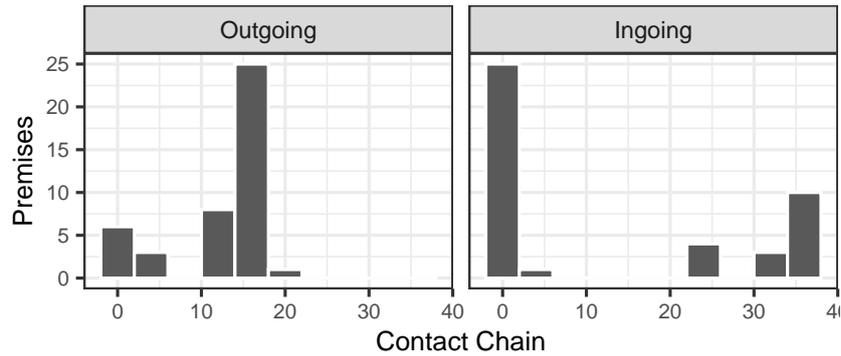


Figure 4: Histogram of the ingoing and outgoing contact chains.

et al. 2015). However, these parameters were used here to help understand the network topology and the behavior of premises regarding the trade of animals.

The **hybridModels** package provides a function to calculate the contact chain size (*ingoing* and *outgoing*), and its use is shown by the code below:

```
R> contactChain <- findContactChain(Data = networkSample,
+   from = "originID", to = "destinationID", Time = "Day",
+   selected.nodes = c(36966, 36798, 37226, 37345))
```

The histograms shown in Figures 3 and 4 indicate that the network of animal movements in Figure 2 has highly heterogeneous distributions for the ingoing and the outgoing contact chains.

Some farms trade batches with few animals, but some batches have many animals. Few farms are connected with many network nodes, and most have few connections. These characteristics are common in animal movement networks (Aznar, Stevenson, Zarich, and León 2011; Bigras-Poulin *et al.* 2006; Buhnerkempe, Grear, Portacci, Miller, Lombard, and Webb 2013; Büttner *et al.* 2015; Green, Gregory, and Munro 2009; Nöremark, Håkansson, Lindström, Wennergren, and Lewerin 2009; Rautureau *et al.* 2011; Rasamoelina-Andriamanivo *et al.* 2014; Thakur, Revie, Hurnik, Poljak, and Sanchez 2016), which makes this network fragment an interesting example.

The local dynamics

Let S_i , I_i , and R_i be the number of susceptible, infected, and recovered animals, respectively, in a given subpopulation i . We assume local dynamics described by the SIR model with a frequency-dependent transmission term given by $\beta S_i I_i / N_i$, where β is the contact rate, and N_i is the total number of animals in subpopulation i . Individuals recover from infection at a rate of γ . In this model, immunity is life-long, and subpopulations are finite and have the same size initially. It is worth noting that the size of subpopulations varies over time due to animal migrations. The model does not account for mortality or breeding. In addition, we assume that the disease neither causes mortality nor affects individual performance.

Although the process is stochastic, we will use the differential equation notation below to help

build the propensity functions and the state change matrix.

$$\frac{dS_i}{dt} = -\beta S_i \frac{I_i}{N_i} \quad (1)$$

$$\frac{dI_i}{dt} = \beta S_i \frac{I_i}{N_i} - \gamma I_i \quad (2)$$

$$\frac{dR_i}{dt} = \gamma I_i \quad (3)$$

$$N_i = S_i + I_i + R_i.$$

These equations define a compartmental model for each subpopulation $i = 1, \dots, 43$.

Model input

The names of the variables in the database that refer to the origin node, destination node, date of movement, and number of cattle moved must be entered in the `var.names` list, as shown in the next code chunk. Furthermore, the propensity functions and the state change matrix must be constructed, identifying which are the state variables. These two objects are built based on the differential Equations 1, 2 and 3, as shown in the following code chunk.

```
R> var.names <- list(from = "originID", to = "destinationID", Time = "Day",
+   arc = "num.animals")
R> prop.func <- c("beta * S * I / (S + I + R)", "gamma * I")
R> state.var <- c("S", "I", "R")
R> state.change.matrix <- matrix(c(-1, 0, 1, -1, 0, 1), nrow = 3, ncol = 2,
+   byrow = TRUE)
```

Each line of the state change matrix corresponds to a state in the order presented by the `state.var` vector, and each column corresponds to a propensity function in the order presented by the `prop.func` vector. The state change matrix values indicate whether a given propensity function does (1 or -1) or does not (0) change the corresponding population. For example, the first position of the state change matrix is -1, which means that the first propensity function ($\beta S \frac{I}{S+I+R}$) models the reduction of the population of susceptible individuals.

Model simulation

The model simulation also requires establishing the values of parameters and the initial conditions of the system in which two initial conditions are established:

```
R> model.parms <- c(beta = 1, gamma = 0.01)
R> init.cond <- rep(200, 43)
R> names(init.cond) <- paste0("S", unique(c(networkSample$originID,
+   networkSample$destinationID)))
R> init.cond1 <- c(init.cond, I36966 = 1, I36798 = 1)
R> init.cond2 <- c(init.cond, I37226 = 1, I37345 = 1)
```

After defining the `beta` and `gamma` parameters of the model, we assigned 200 susceptible animals to each subpopulation. For the first set of simulations, we choose two farm premises

| Initial condition | Premises | Contact chain | | Degree | |
|-------------------|----------|---------------|---------|--------|----|
| | | Outgoing | Ingoing | Out | In |
| 1 | 36966 | 19 | 1 | 1 | 0 |
| | 36798 | 18 | 1 | 1 | 0 |
| 2 | 37226 | 13 | 34 | 6 | 5 |
| | 37345 | 13 | 34 | 3 | 2 |

Table 2: Centrality measures for the premises chosen to contain initial infection cases.

with high outgoing contact chain values (premises 36966 and 36798) as the initial infection cases. For the second set, we choose two farm premises with low outgoing contact chain values (premises 37226 and 37345) as the initial infection cases. Some parameters describing the abovementioned premises are shown in Table 2.

Results and discussion

For each initial condition, 100 simulations were performed using the direct method of the package (Pineda-Krch 2008). The stochastic simulations may have a high computational cost; therefore, the **hybridModels** package allows running them in parallel using the **doParallel** and **foreach** packages (Kane, Emerson, and Weston 2013; Microsoft Corporation and Weston 2019, 2020). In this example, four cores were used (the default is "max", which is the maximum number of real cores in the computer processor).

```
R> results.Cond1 <- hybridModel(network = networkSample,
+   var.names = var.names, model.parms = model.parms,
+   state.var = state.var, prop.func = prop.func,
+   state.change.matrix = state.change.matrix, init.cond = init.cond1,
+   sim.number = 100, num.cores = 4)
R> results.Cond2 <- hybridModel(network = networkSample,
+   var.names = var.names, model.parms = model.parms,
+   state.var = state.var, prop.func = prop.func,
+   state.change.matrix = state.change.matrix, init.cond = init.cond2,
+   sim.number = 100, num.cores = 4)
```

Figures 5, 6, 7, and 8 were generated by the generic plot function (see code below). These plots are customizable via the commands available in **ggplot2** (Wickham 2009).

```
R> plot(results.Cond1, plot.type = "subpop.mean")
R> plot(results.Cond1, plot.type = "pop.mean")
R> plot(results.Cond2, plot.type = "subpop.mean")
R> plot(results.Cond2, plot.type = "pop.mean")
```

Using the same initial conditions, we changed the value of the transmission rate β from 1 to 0.1. The results are shown in Figures 7 and 8.

The function **summary()** summarizes the distribution of the simulation results (i.e., the number of susceptible and infected individuals) for specific nodes at a given time. The default is the last date in the input data frame. Let us suppose we are interested in the summary of susceptible and infected individuals in nodes 36812 and 36813 with initial conditions 1 and 2, and $\beta = 1$ and $\gamma = 0.01$. The outputs are shown below.

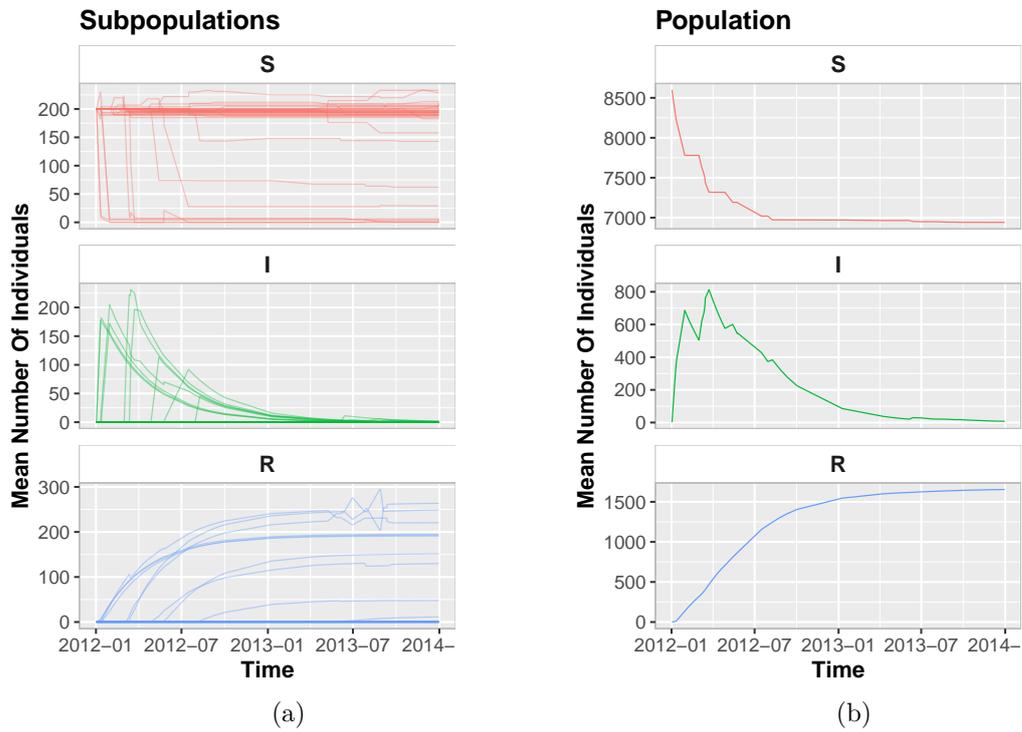


Figure 5: Initial condition 1 with $\beta = 1$: (a) mean number of individuals in each subpopulation; (b) mean number of individuals in the population.

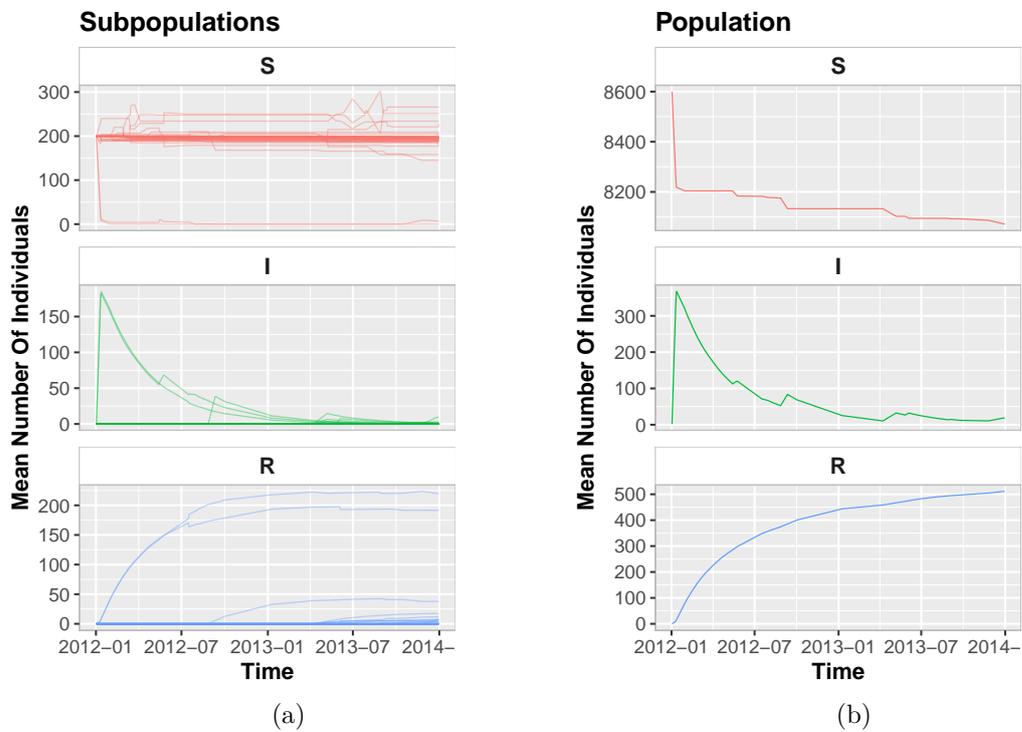


Figure 6: Initial condition 2 with $\beta = 1$: (a) mean number of individuals in each subpopulation; (b) mean number of individuals in the population.

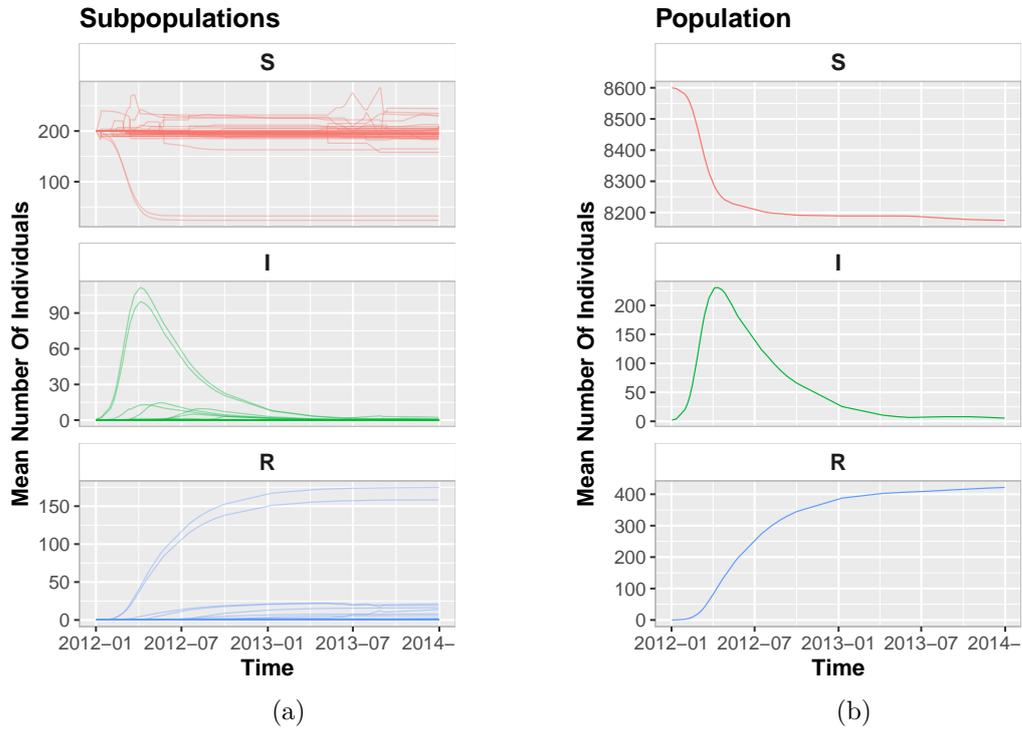


Figure 7: Initial condition 1 with $\beta = 0.1$: (a) mean number of individuals in each subpopulation; (b) mean number of individuals in the population.

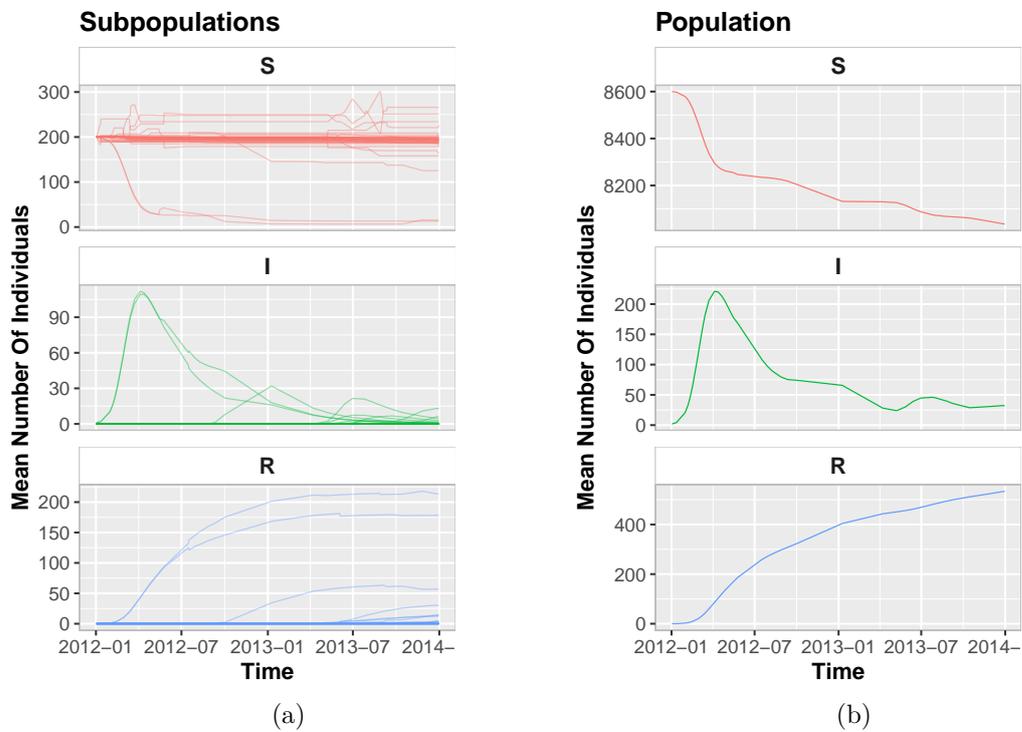


Figure 8: Initial condition 2 with $\beta = 0.1$: (a) mean number of individuals in each subpopulation; (b) mean number of individuals in the population.

```
R> summary(results.Cond1, stateVars = c("S", "I"), nodes = c(36812, 36813))
```

| S36812 | | I36812 | | S36813 | | I36813 | |
|----------|----|----------|-------|----------|--------|----------|--------|
| Min. | :7 | Min. | :0.00 | Min. | : 0.00 | Min. | : 0.00 |
| 1st Qu.: | :7 | 1st Qu.: | :0.00 | 1st Qu.: | : 0.00 | 1st Qu.: | : 0.00 |
| Median | :7 | Median | :0.00 | Median | : 0.00 | Median | : 0.00 |
| Mean | :7 | Mean | :0.32 | Mean | : 1.29 | Mean | : 1.12 |
| 3rd Qu.: | :7 | 3rd Qu.: | :1.00 | 3rd Qu.: | : 0.00 | 3rd Qu.: | : 1.00 |
| Max. | :7 | Max. | :5.00 | Max. | :10.00 | Max. | :16.00 |

```
R> summary(results.Cond2, stateVars = c("S", "I"), nodes = c(36812, 36813))
```

| S36812 | | I36812 | | S36813 | | I36813 | |
|----------|--------|----------|-------|----------|------|----------|----|
| Min. | :219.0 | Min. | :0.00 | Min. | :266 | Min. | :0 |
| 1st Qu.: | :228.0 | 1st Qu.: | :0.00 | 1st Qu.: | :266 | 1st Qu.: | :0 |
| Median | :228.0 | Median | :0.00 | Median | :266 | Median | :0 |
| Mean | :227.6 | Mean | :0.05 | Mean | :266 | Mean | :0 |
| 3rd Qu.: | :228.0 | 3rd Qu.: | :0.00 | 3rd Qu.: | :266 | 3rd Qu.: | :0 |
| Max. | :228.0 | Max. | :3.00 | Max. | :266 | Max. | :0 |

Using a fragment of a dynamic cattle trade network, we showed how to customize and simulate hybrid epidemic models. We also aimed to highlight the role of the dynamic network in the disease spread.

Hybrid epidemic models can provide information that contributes to identifying farms that represent greater risk for the rest of the trade network or generate the largest epidemics. To address this question and exemplify the complexity that the network brings to the evolution of the epidemic, initial cases of infected animals in premises that occupied different positions in the trade network were added.

In initial condition 1, these premises were more isolated in the network, although they had higher outgoing contact chain values. In initial condition 2, premises that occupied more central positions were chosen (Figure 2). For the higher transmission rate ($\beta = 1$), condition 1 produces a larger epidemic, on average, than condition 2 (Figures 5 and 6). For the lower transmission rate ($\beta = 0.1$), condition 2 (premises that had initial infection cases were more connected and had smaller outgoing contact chains than those in condition 1) generated a slightly larger epidemic, on average, than condition 1 (Figures 7 and 8).

In the simulations described above, the number of animals traded is independent of premises' state variables (S , I and R). Moreover, animals have equal probabilities of emigrating, as they are randomly sampled. However, one might want to simulate scenarios in which animals' chance of emigration is influenced by the state variables of their premises. For such scenarios, two arguments can be applied: `emigrRule` and `probWeights`. An example of how to use these arguments is presented below (results not shown).

```
R> networkSample[, "num.animals"] <- 1
R> emigrRule <- "if (I < 0.05 * (S + I + R)) {0.1 * (S + I + R)}
+   else {0.02 * (S + I + R)}"
R> probWeights <- c(S = "1",
+   I = "if (I < 0.05 * (S + I + R)) {1} else {100}", R = "1")
```

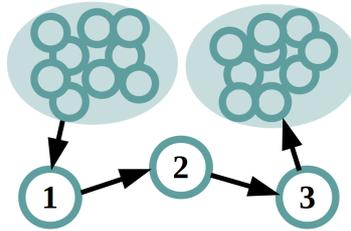


Figure 9: Schematic representation of the network: two sets of nodes (shaded regions) and three nodes connected by arcs that represent the movement direction of individuals.

```
R> results <- hybridModel(network = networkSample,
+   probWeights = probWeights, emigrRule = emigrRule,
+   var.names = var.names, model.parms = model.parms,
+   state.var = state.var, prop.func = prop.func, sim.number = 100,
+   state.change.matrix = state.change.matrix, init.cond = init.cond1)
R> plot(results, plot.type = "pop.mean")
```

In the code shown above, each premises trades animals at a constant rate, and trade restrictions are applied to premises in which the proportion of infected animals (prevalence) is greater than 5% of the population (`emigrRule`). In premises with a prevalence less than 5%, 10% of all animals are allowed to emigrate; otherwise, 2% are allowed to emigrate. Premises with restrictions adopt a strategy of trying to sell more infected animals than the number sold in the no-restriction scenario (`probWeights`). If the prevalence is less than 5%, the weight applied to the probability of migration for an infected animal is 1; otherwise, the weight is 100. For susceptible and recovered animals, the weight is 1. Thus, in this particular example, the migration of an infected animal is more likely to occur than the migration of a susceptible or a recovered animal if the prevalence of infection on the premises is greater than 5%.

3.2. Three-node model

Comparison with a deterministic model

This example compares a hybrid model with a coupled ordinary differential equations (ODE) model. We simulated disease spread using a SI model in the network shown in Figure 9. We focus on the three nodes that connect two set of nodes. One of these two sets of nodes (on the left in Figure 9) works as a source compartment from which individuals transition to node 1; the other set of nodes (on the right in Figure 9) is equivalent to a destination compartment for individuals departing from node 3. In the network of Figure 9, 100 individuals migrate from one node to another every 30 days, and the migrations occur on the same day. In addition, only healthy individuals reach node 1 ($i = 1$), and emigrants are randomly chosen from each node.

In terms of connections among the nodes, this network is simpler than that of the previous example. The three nodes we are interested in have well-behaved connections; as a consequence, we might consider it as a static network. In contrast with a temporal network, the arcs between these nodes behave in the same way over time.

ODE model

To describe the disease spread throughout the network, we set a deterministic SI model (coupled ODE) that follows the equations below. Let us not forget that we are only interested in nodes $i = 1, 2$ and 3 .

$$\begin{aligned}\frac{dS_i}{dt} &= -\beta S_i \frac{I_i}{N_i} + \alpha \frac{S_{i-1}}{N_{i-1}} - \alpha \frac{S_i}{N_i} \\ \frac{dI_i}{dt} &= +\beta S_i \frac{I_i}{N_i} + \alpha \frac{I_{i-1}}{N_{i-1}} - \alpha \frac{I_i}{N_i} \\ N_i &= S_i + I_i \\ N_0 &= S_0.\end{aligned}$$

S_i , I_i and N_i represent the number of susceptible, infected, and total number of individuals in node i , and α is the migration rate. We consider the sets of nodes as two single nodes, and their indices are $i = 0$ and $i = 4$.

Because the **hybridModels** package does not solve ODEs, we simulate this scenario using the **deSolve** package (Soetaert, Petzoldt, and Setzer 2010). The code used to conduct the simulation is presented in Appendix A.

Hybrid model

The hybrid model approach for the disease spread follows Equations 4 and 5 that have the same force of infection as that in the deterministic approach. However, the migration of individuals from one node to another is set by building a data frame that describes it.

$$\frac{dS_i}{dt} = -\beta S_i \frac{I_i}{N_i} \tag{4}$$

$$\frac{dI_i}{dt} = +\beta S_i \frac{I_i}{N_i} \tag{5}$$

$$N_i = S_i + I_i.$$

The code used to create the network is shown below, and the first six rows of the data frame are presented in Table 3.

```
R> network <- data.frame(originNode = rep(0:3, 3600/30),
+   destinationNode = rep(1:4, 3600/30),
+   Day = rep(seq.Date(from = as.Date("2006-01-31"),
+   by = 30, length.out = 120), each = 4), num.individuals = 100)
```

All of the parameters needed to simulate disease spread are presented in Table 4.

We first set the names of the variables, the propensity function, the state variables, and the state change matrix as follows.

```
R> var.names <- list(from = "originNode", to = "destinationNode",
+   Time = "Day", arc = "num.individuals")
R> prop.func <- "(beta * S * I) / (S + I)"
R> state.var <- c("S", "I")
R> state.change.matrix <- matrix(c(-1, 1))
```

| originNode | destinationNode | Day | num.individuals |
|------------|-----------------|------------|-----------------|
| 0 | 1 | 2006-01-31 | 100 |
| 1 | 2 | 2006-01-31 | 100 |
| 2 | 3 | 2006-01-31 | 100 |
| 3 | 4 | 2006-01-31 | 100 |
| 0 | 1 | 2006-03-02 | 100 |
| 1 | 2 | 2006-03-02 | 100 |

Table 3: First 6 rows of the data frame of individuals (`num.individuals`) that migrate from `originNode` to `destinationNode` on a given date (`Day`).

| Parameter | Value | Description |
|-----------|-------|---|
| α | 10/3 | Migration rate. |
| β | 0.006 | Transmission rate. |
| t_f | 3600 | Duration of the simulation in days. |
| N_i | 1000 | Number of individuals inside each node. |
| $I_1(0)$ | 1 | Initial condition. |

Table 4: Values and description of the parameters used.

We then define the values of the parameters and the initial conditions. Note that we choose arbitrary values for the susceptible population in each node and one infected individual in node 1.

```
R> model.parms <- c(beta = 0.006)
R> init.cond <- c(S0 = 120000, S1 = 999, S2 = 1000, S3 = 1000, S4 = 10)
R> init.cond <- c(init.cond, I1 = 1)
```

The parameters used in this example are summarized in Table 4.

Results

After setting the model parameters, we conducted the simulation using the **hybridModels** package.

```
R> results <- hybridModel(network = network, var.names = var.names,
+   model.parms = model.parms, state.var = state.var,
+   prop.func = prop.func, state.change.matrix = state.change.matrix,
+   init.cond = init.cond, sim.number = 100, num.cores = "max")
```

The number of infected individuals in each node under both simulation approaches is shown in Figure 10: black lines represent the results of the ODE model, and gray lines denote each run of the hybrid model.

This example emphasizes aspects concerning the **hybridModel** package, i.e., validation and differences from deterministic models. We expected that both approaches would provide qualitatively comparable results for simple models, such as that illustrated in Figure 10. First, let us highlight the main difference of both approaches.

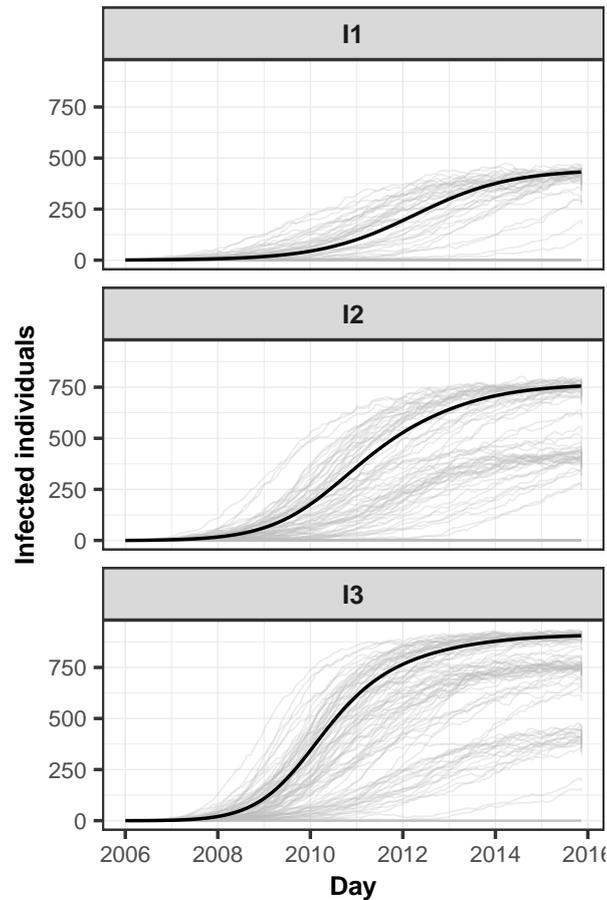


Figure 10: Results of the simulations for the number of infected individuals using the ODE approach (black solid line) and the curves obtained using the hybrid model (gray lines).

The hybrid model simulations show that there are two, three and four basins of attraction (plateaus with a concentration of gray lines in Figure 10) for nodes 1, 2, and 3, respectively, whereas the deterministic approach has only one. The hybrid model simulations show that each node adds basins of attractions from the previous node. This event occurs because all of the nodes are able to end up without infected individuals. The ODE approach does not generate this outcome. Even in this simple simulation, stochasticity contributes to a more complex scenario, which represents the main difference of both approaches.

In simulations in which all of the nodes have infected individuals, notwithstanding the fact that the stochastic model provides several alternative solutions, the curves obtained using the **hybridModels** package were qualitatively similar to the results of the ODE approach (Figure 10). This event can be understood as a numerical way to validate or check the accuracy of the **hybridModels** simulations.

3.3. Commuting

The **hybridModels** package can also be used to simulate the commuting of individuals between nodes. We used a modeling framework similar to that described by [Coelho, Cruz, and Codeço \(2008\)](#) based on the “forest fire” model ([Grenfell, Bjørnstad, and Kappey 2001](#)). In this

approach, individuals that commute between nodes contribute temporarily to the number of infected individuals in the destination node but not to its demography (Coelho *et al.* 2008). As an example, we used a SIS model; however, users can implement their own customized model.

Local dynamics in the commuting model

Let S_i and I_i be the number of susceptible and infected individuals in a subpopulation i . At the local scale, we consider SIS model dynamics with a frequency-dependent transmission term described by the following set of equations:

$$\begin{aligned}\frac{dS_i}{dt} &= -\beta S_i \frac{I_i + i_i^{\text{in}}}{N_i + n_i^{\text{in}}} + \gamma I_i \\ \frac{dI_i}{dt} &= \beta S_i \frac{I_i + i_i^{\text{in}}}{N_i + n_i^{\text{in}}} - \gamma I_i \\ N_i &= S_i + I_i,\end{aligned}$$

where β is the contact rate, $N_i = S_i + I_i$ is the total number of individuals in subpopulation i , γ is the recovery rate, i_i^{in} is the number of infected individuals that commute to node i at a given time, and $n_i^{\text{in}} = s_i^{\text{in}} + i_i^{\text{in}}$ is the total number of individuals that commute to node i .

Model input

In this example, we simulate the spread of an infectious disease among 30 patches, each of which has a population between 500 and 1,000 individuals, among which 10 to 50 commute on randomly sampled days. We assume that a given number of commuting connections (each of which has a given number of individuals commuting on a given date) occur across the 30 patches between April and June. The following code starts by setting randomly 1,000 commuting connections; then, we discard all of the loops, i.e., connections from a given patch to itself. We randomly select the date (`Day`), origin patch (`origin`), destination patch (`destination`), and number of individuals commuting from the origin to the destination patch (`num.individuals`). We then create a data frame (`commuting`) containing these data.

```
R> Day <- sample(seq(as.Date("2017/04/01"), as.Date("2017/06/30"),
+   by = "day"), size = 1000, replace = TRUE)
R> origin <- sample.int(30, size = 1000, replace = TRUE)
R> destination <- sample.int(30, size = 1000, replace = TRUE)
R> num.individuals <- as.integer(runif(1000, min = 10, max = 50))
R> commuting <- data.frame(Day = Day, origin = origin,
+   destination = destination, num.individuals = num.individuals)
R> commuting <- commuting[-which(origin == destination), ]
```

Next, we need to set the names of the variables, the propensity functions for the SIS model, the state variables, and the state change variables as in the example of Section 3.1. In the commuting model, an additional `infl.var` vector is related to the number of infected and susceptible individuals that influence the disease spread dynamics in the destination node.

```
R> var.names <- list(from = "origin", to = "destination", Time = "Day",
+   arc = "num.individuals")
```

```
R> prop.func <- c("beta * S * (I + i) / (S + I + s + i)", "gamma * I")
R> state.var <- c("S", "I")
R> infl.var <- c(S = "s", I = "i")
R> state.change.matrix <- matrix(c(-1, 1, 1, -1), nrow = 2, ncol = 2,
+   byrow = TRUE)
```

Model simulation and results

Before running the model, we need to provide the parameters related to the disease dynamics (`model.parms`) and the initial condition (`init.cond`). In the code below, we set $\beta = 0.1$ day⁻¹ and $\gamma = 0.05$ day⁻¹. We also assume that all nodes have between 500 and 1,000 individuals and that only nodes 10 and 12 have ten infected individuals each.

```
R> model.parms <- c(beta = 0.1, gamma = 0.05)
R> init.cond <- as.integer(runif(30, min = 500, max = 1000))
R> names(init.cond) <- paste0("S", seq(1, 30, by = 1))
R> init.cond <- c(init.cond, I10 = 10, I12 = 10)
```

We run the model using the `hybridModel` function.

```
R> sim.commuting <- hybridModel(network = commuting, var.names = var.names,
+   model.parms = model.parms, state.var = state.var, infl.var = infl.var,
+   prop.func = prop.func, init.cond = init.cond,
+   state.change.matrix = state.change.matrix, sim.number = 100,
+   num.cores = 4)
```

Comparison with the permanent migration model

To run a permanent migration model using the same data frame, parameters and initial conditions of the commuting model, we use the following code to set the propensity functions, the state variables, and the state change matrix.

```
R> prop.func <- c("beta * S * I / (S + I)", "gamma * I")
R> state.var <- c("S", "I")
R> state.change.matrix <- matrix(c(-1, 1, 1, -1), nrow = 2, ncol = 2,
+   byrow = TRUE)
```

We then conduct the simulations for the permanent migration model.

```
R> sim.migration <- hybridModel(network = commuting, var.names = var.names,
+   model.parms = model.parms, state.var = state.var,
+   prop.func = prop.func, init.cond = init.cond,
+   state.change.matrix = state.change.matrix, sim.number = 100,
+   num.cores = 4)
```

The results for the number of susceptible and infected individuals in the patches for both commuting and permanent migration models are shown in Figure 11.

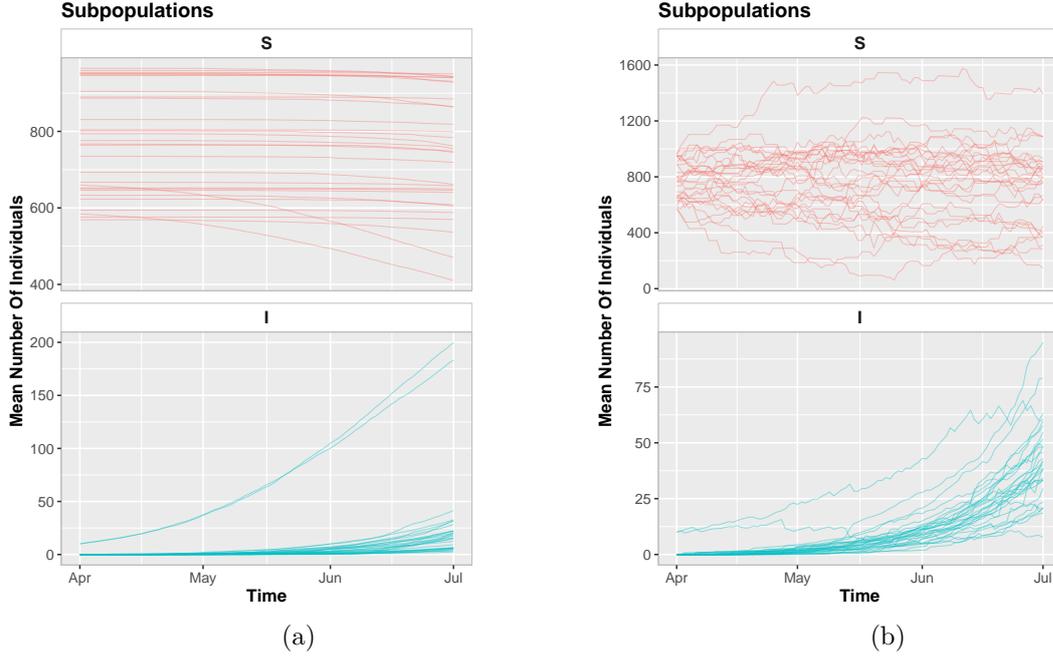


Figure 11: Results of the simulations for the mean number of susceptible and infected individuals in the patches, using the commuting model (a) and the permanent migration model (b).

```
R> plot(sim.commuting, plot.type = "subpop.mean")
R> plot(sim.migration, plot.type = "subpop.mean")
```

In Figure 11, we note that more intense oscillations occurred in the number of susceptible and infected individuals in the permanent migration model than in the commuting model.

3.4. Commuting within São Paulo

As another example of the commuting model, we implemented a SLIR (susceptible-latent-infected-recovered) model with the influenza parameters and real network data on the daily commuting of people within the metropolitan area of São Paulo. Also known as “Greater São Paulo”, this is the biggest metropolitan area in Brazil, with approximately 20 million people distributed among 39 municipalities, including the city of São Paulo.

Local dynamics in the commuting model

Let S_i , L_i , I_i and R_i be the number of susceptible, latent, infected and recovered individuals in a subpopulation i . At the local scale, we consider a SLIR model with a frequency-dependent transmission term described by the following set of equations:

$$\begin{aligned}\frac{dS_i}{dt} &= -\beta S_i \frac{I_i + i_i^{\text{in}}}{N_i + n_i^{\text{in}}} \\ \frac{dL_i}{dt} &= \beta S_i \frac{I_i + i_i^{\text{in}}}{N_i + n_i^{\text{in}}} - \kappa L_i\end{aligned}$$

$$\begin{aligned}\frac{dI_i}{dt} &= \kappa L_i - \gamma I_i \\ \frac{dR_i}{dt} &= \gamma I_i \\ N_i &= S_i + L_i + I_i + R_i,\end{aligned}$$

where β is the contact rate, $N_i = S_i + L_i + I_i + R_i$ is the total number of individuals in subpopulation i , $1/\kappa$ is the latent period, $1/\gamma$ is the infectious period, i_i^{in} is the number of infected individuals who commute to node i at a given time, and $n_i^{\text{in}} = s_i^{\text{in}} + l_i^{\text{in}} + i_i^{\text{in}} + r_i^{\text{in}}$ is the total number of individuals who commute to node i .

Data

We used data from the “2007 Origin and Destination Survey” conducted by the “Companhia do Metropolitano de São Paulo – Metro”, an organization within the São Paulo state government that is responsible for the implementation and operation of the subway system. This survey was conducted to estimate the number of daily trips within the metropolitan area of São Paulo. The area was divided into 460 districts/zones, and the number of daily trips between each zone was estimated based on a household survey. The data on daily trips and the population were downloaded and are accessible through the Metro Transparency Portal (Portal de transparência do Metro), at <https://transparencia.metrosp.com.br/dataset/pesquisa-origem-e-destino>. For this example, we estimate the number of daily commutes based on the first commute of each person leaving their home to another district/zone, which results in almost 10 million daily commutes.

We first load the commute and population databases:

```
R> daily_commuting <- read.csv2("daily_commuting_from_residence.csv",
+   fileEncoding = "UTF-8")
R> zones <- read.csv2("zones.csv", fileEncoding = "UTF-8")
```

We assume that these commutes occur on every day of the simulation period and replicate them accordingly. We choose the duration and initial date of the simulation and create a database with the same daily commutes for each day of the time period.

```
R> duration <- 360
R> initial.date <- as.Date("2017/01/01")
R> days <- seq(initial.date, initial.date + duration, by = "day")
R> commuting <- expand.grid(1:nrow(daily_commuting), days)
R> names(commuting) <- c("commute", "Day")
R> commuting[, "origin"] <- daily_commuting[commuting[, "commute"],
+   "origin"]
R> commuting[, "destination"] <- daily_commuting[commuting[, "commute"],
+   "destination"]
R> commuting[, "num.individuals"] <- daily_commuting[commuting[, "commute"],
+   "num.individuals"]
```

Model input

As previously described in Section 3.3, we need to set the names of the variables, the propensity functions, the state change matrix, and the number of individuals who influence the disease spread dynamics in the destination node.

```
R> var.names <- list(from = "origin", to = "destination", Time = "Day",
+   arc = "num.individuals")
R> prop.func <- c("beta * S * (I + i) / (S + L + I + R + s + l + i + r)",
+   "kappa * L", "gamma * I")
R> state.var <- c("S", "L", "I", "R")
R> infl.var <- c(S = "s", L = "l", I = "i", R = "r")
R> state.change.matrix <- matrix(c(-1, 0, 0, 1, -1, 0, 0, 1, -1, 0, 0, 1),
+   nrow = 4, ncol = 3, byrow = TRUE)
```

Model simulation and results

To provide the `model.parms`, we set $\kappa = \frac{1}{1.9} \text{ day}^{-1} \approx 0.53 \text{ day}^{-1}$ and $\gamma = \frac{1}{4.1} \text{ day}^{-1} \approx 0.24 \text{ day}^{-1}$, assuming the rates κ and γ , respectively, as the inverse of a latent period of 1.9 days and of an infectious period of 4.1 days for pandemic influenza (Longini, Jr., Halloran, Nizam, and Yang 2004; Mills, Robins, and Lipsitch 2004). We estimate β assuming that $R_0 = \beta/\gamma$ and that $R_0 = 1.7$ (Longini, Jr. *et al.* 2004) such that $\beta = 1.7 * 4.1^{-1} = 0.41 \text{ day}^{-1}$. The purpose of this section is to provide an example of a simulation running on a larger dataset, based on real-world data. For more information on the model parameters please refer to Longini, Jr. *et al.* (2004); Mills *et al.* (2004); Chowell, Miller, and Viboud (2008); Chowell, Viboud, Simonsen, Miller, and Alonso (2010).

```
R> model.parms <- c(beta = 1.7/4.1, kappa = 1/1.9, gamma = 1/4.1)
```

We set the initial condition `init.cond` with the entire population of each zone being susceptible, and insert 100 infected individuals in zone 217, which corresponds to Cidade Tiradentes, a peripheral district in the city of São Paulo.

```
R> nodes <- sort(unique(c(commuting[, "origin"],
+   commuting[, "destination"])))
R> init.cond <- as.integer(zones[nodes, "population"])
R> names(init.cond) <- paste0("S", nodes)
R> init.cond <- c(init.cond, I217 = 100)
```

To save running time, we use the explicit tau-leap method (ETL) instead of the traditional direct method. It is also possible to use other methods implemented by the **GillespieSSA** package (Pineda-Krch 2008), like the binomial tau-leap or the optimized tau-leap. The direct method is an exact method in the sense presented by Gillespie (Gillespie 2001), but it is computationally demanding. To do this, we set the parameter `ssa.method` to "ETL", in the `hybridModel` function. This run takes approximately 60 minutes in a computer with a i7 processor 7th generation (we used just one core) and 16GB RAM (note that the process consumed only 2 GB RAM).

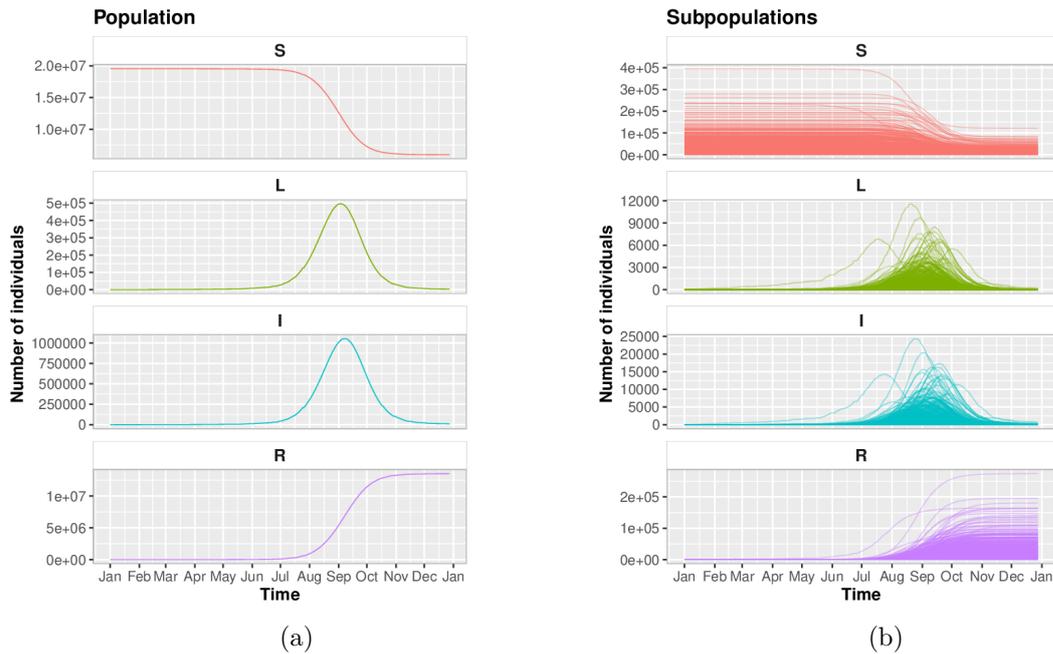


Figure 12: Number of susceptible, latent, infected and recovered individuals in the total population (a) and in each zone of São Paulo metropolitan area (b).

```
R> sim.results <- hybridModel(network = commuting, var.names = var.names,
+   model.parms = model.parms, state.var = state.var, infl.var = infl.var,
+   prop.func = prop.func, init.cond = init.cond,
+   state.change.matrix = state.change.matrix, sim.number = 1,
+   num.cores = 1, ssa.method = list(method = "ETL", tau = 0.3))
```

The `plot` function shows the number of susceptible, latent, infected and recovered individuals. It is possible to adapt the plots using `ggplot2` functions, as shown in the following code, which generates Figure 12.

```
R> plot(sim.results, plot.type = "subpop.mean") +
+   ylab("Number of individuals") +
+   scale_x_date(date_breaks = "1 month", date_labels = "%b")
R> plot(sim.results, plot.type = "pop.mean") +
+   ylab("Number of individuals") +
+   scale_x_date(date_breaks = "1 month", date_labels = "%b")
```

4. Discussion

The results of the simulation using a fragment of a dynamic cattle trade network (Section 3.1) showed that classifying the importance of the farms in the epidemic progression depends on a combination of, at least, its position in the network and the submodel applied at the local scale.

In the schematic three-node network example, we compared the stochastic simulations obtained using the **GillespieSSA** package and the numerical solution of a system of ODEs (solved using the **deSolve** package). We tested the functions of the **hybridModels** package through a series of visual and spot checks. However, validation is a difficult task, and we decided to accomplish this goal by comparing the **hybridModels** package with a deterministic compartmental model (Grimm and Railsback 2005). We found that, in the numerical simulation performed using the **hybridModels** package, the stochastic model provides several alternative solutions compared with the results of the ODE approach, allowing for the consideration of more complex scenarios.

The **hybridModels** package aims to help create hybrid models that can contribute to the investigation of epidemics among subpopulations and to other studies that might benefit from the hybrid modeling paradigm. The **hybridModels** package could be applied to simulate disease spread in both human and animal populations, considering the interplay between local dynamics and migration processes, either by permanent migration or commuting.

As examples, we showed simulations of disease spread between farms through animal movement and between different localities through human commuting; however, the **hybridModels** package can be used for a wide range of applications. For example, it can be used to simulate the transfer of patients between hospitals and the spread of nosocomial pathogens in hospital units or the spread of infectious diseases across countries via travelers, considering the flow of air traffic.

The simulation software **Epigrass** (Coelho *et al.* 2008), which is written entirely in the Python language (van Rossum *et al.* 2011), is a tool to simulate network-epidemic models using an epidemiological modeling approach based on the assumption that individuals commute between localities and contribute temporarily to the number of infected individuals at the destination locality but not to its demography. The user can conduct the simulation routines of the **hybridModels** package using a commuting assumption similar to that used in **Epigrass** or considering the effective migration of individuals between localities, which might be more realistic depending on the scenario analyzed.

Simulation models exist for the spread of infectious diseases in animal populations such as **AusSpread** (Garner and Beckett 2005), **NAADSM** (Harvey *et al.* 2007) and **InterSpread Plus** (Stevenson *et al.* 2013). These simulation models have an IBM structure in which clusters of animals are the units of interest. They are spatially explicit, stochastic, and state-transition models (Harvey *et al.* 2007). The **InterSpread Plus** model (Stevenson *et al.* 2013) was written in C++, and the user is able to specify model parameters via a graphical user interface. The **NAADSM** (Harvey *et al.* 2007) model also provides a graphical user interface and output visualization features, and **AusSpread** (Garner and Beckett 2005) produces tabulated statistics and outbreak maps. The **hybridModels** package is an interesting alternative for users (particularly those of R) interested in customizing the interaction between local and global scales as well as the internal transmission process by defining the epidemic model at the local scale.

Implementing the stochastic simulation algorithm is a bottleneck in the performance of the simulations, which is currently implemented in the R language (Pineda-Krch 2008). Thus, a direction for future development of the package is the implementation of functions that execute the local dynamics with greater efficiency.

Currently, the package is implemented to address customizable equations at the local scale.

However, migration is the only type of interaction allowed between nodes. A possible next step in the development of the package is the possibility of allowing other forms of interaction between the network nodes.

Acknowledgments

We thank the Agricultural Protection and Supervision Agency of Pernambuco (Agência de Defesa e Fiscalização Agropecuária de Pernambuco – ADAGRO-PE), the Coordination for the Improvement of Higher Education Personnel (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Capes) and the National Council for Technological and Scientific Development (Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq). The graph presented in Figure 2 was generated using the **igraph** package (Csardi and Nepusz 2006). We used the **reshape2** and **ggplot2** packages (Wickham 2007, 2009) to generate the plots. We are grateful to the anonymous referees for their constructive feedback on the **hybridModels** package and critical comments on the manuscript.

References

- Aznar MN, Stevenson MA, Zarich L, León EA (2011). “Analysis of Cattle Movements in Argentina, 2005.” *Preventive Veterinary Medicine*, **98**(2–3), 119–127. doi:10.1016/j.prevetmed.2010.11.004.
- Bigras-Poulin M, Barfod K, Mortensen S, Greiner M (2007). “Relationship of Trade Patterns of the Danish Swine Industry Animal Movements Network to Potential Disease Spread.” *Preventive Veterinary Medicine*, **80**(2–3), 143–165. doi:10.1016/j.prevetmed.2007.02.004.
- Bigras-Poulin M, Thompson RA, Chriel M, Mortensen S, Greiner M (2006). “Network Analysis of Danish Cattle Industry Trade Patterns as an Evaluation of Risk Potential for Disease Spread.” *Preventive Veterinary Medicine*, **76**(1–2), 11–39. doi:10.1016/j.prevetmed.2006.04.004.
- Buhnerkempe MG, Gear DA, Portacci K, Miller RS, Lombard JE, Webb CT (2013). “A National-Scale Picture of U.S. Cattle Movements Obtained from Interstate Certificate of Veterinary Inspection Data.” *Preventive Veterinary Medicine*, **112**(3–4), 318–329. doi:10.1016/j.prevetmed.2013.08.002.
- Büttner K, Krieter J, Traulsen I (2015). “Characterization of Contact Structures for the Spread of Infectious Diseases in a Pork Supply Chain in Northern Germany by Dynamic Network Analysis of Yearly and Monthly Networks.” *Transboundary and Emerging Diseases*, **62**(2), 188–199. doi:10.1111/tbed.12106.
- Cao Y, Gillespie DT, Petzold LR (2006). “Efficient Step Size Selection for the Tau-Leaping Simulation Method.” *The Journal of Chemical Physics*, **124**(4), 044109. doi:10.1063/1.2159468.

- Chatterjee A, Vlachos DG, Katsoulakis MA (2005). “Binomial Distribution Based Tau-Leap Accelerated Stochastic Simulation.” *The Journal of Chemical Physics*, **122**(2), 024112. doi:10.1063/1.1833357.
- Chowell G, Miller MA, Viboud C (2008). “Seasonal Influenza in the United States, France, and Australia: Transmission and Prospects for Control.” *Epidemiology and Infection*, **136**(6), 852–864. doi:10.1017/s0950268807009144.
- Chowell G, Viboud C, Simonsen L, Miller M, Alonso WJ (2010). “The Reproduction Number of Seasonal Influenza Epidemics in Brazil, 1996–2006.” *Proceedings of the Royal Society B: Biological Sciences*, **277**(1689), 1857–1866. doi:10.1098/rspb.2009.1897.
- Coelho FC, Cruz OG, Codeço CT (2008). “**Epigrass**: A Tool to Study Disease Spread in Complex Networks.” *Source Code for Biology and Medicine*, **3**, 3. doi:10.1186/1751-0473-3-3.
- Csardi G, Nepusz T (2006). “The **igraph** Software Package for Complex Network Research.” *InterJournal, Complex Systems*, 1695.
- Dubé C, Ribble C, Kelton D, McNab B (2008). “Comparing Network Analysis Measures to Determine Potential Epidemic Size of Highly Contagious Exotic Diseases in Fragmented Monthly Networks of Dairy Cattle Movements in Ontario, Canada.” *Transboundary and Emerging Diseases*, **55**(9–10), 382–392. doi:10.1111/j.1865-1682.2008.01053.x.
- Frössling J, Ohlson A, Björkman C, Håkansson N, Nöremark M (2012). “Application of Network Analysis Parameters in Risk-Based Surveillance – Examples Based on Cattle Trade Data and Bovine Infections in Sweden.” *Preventive Veterinary Medicine*, **105**(3), 202–208. doi:10.1016/j.prevetmed.2011.12.011.
- Garner MG, Beckett SD (2005). “Modelling the Spread of Foot-and-Mouth Disease in Australia.” *Australian Veterinary Journal*, **83**(12), 758–766. doi:10.1111/j.1751-0813.2005.tb11589.x.
- Gillespie DT (1976). “A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions.” *Journal of Computational Physics*, **22**(4), 403–434. doi:10.1016/0021-9991(76)90041-3.
- Gillespie DT (1977). “Exact Stochastic Simulation of Coupled Chemical Reactions.” *Journal of Physical Chemistry*, **81**(25), 2340–2361. doi:10.1021/j100540a008.
- Gillespie DT (2001). “Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems.” *The Journal of Chemical Physics*, **115**(4), 1716. doi:10.1063/1.1378322.
- Gillespie DT (2007). “Stochastic Simulation of Chemical Kinetics.” *Annual Review of Physical Chemistry*, **58**(1), 35–55. doi:10.1146/annurev.physchem.58.032806.104637.
- Green DM, Gregory A, Munro LA (2009). “Small- And Large-Scale Network Structure of Live Fish Movements in Scotland.” *Preventive Veterinary Medicine*, **91**(2–4), 261–269. doi:10.1016/j.prevetmed.2009.05.031.
- Grenfell BT, Bjørnstad ON, Kappey J (2001). “Travelling Waves and Spatial Hierarchies in Measles Epidemics.” *Nature*, **414**, 716–723. doi:10.1038/414716a.

- Grimm V, Berger U, Bastiansen F, Eliassen S, Ginot V, Giske J, Goss-Custard J, Grand T, Heinz SK, Huse G, Huth A, Jepsen JU, Jørgensen C, Mooij WM, Müller B, Pe'er G, Piou C, Railsback SF, Robbins AM, Robbins MM, Rossmanith E, Rüger N, Strand E, Souissi S, Stillman RA, Vabø R, Visser U, DeAngelis DL (2006). “A Standard Protocol for Describing Individual-Based and Agent-Based Models.” *Ecological Modelling*, **198**(1–2), 115–126. doi:[10.1016/j.ecolmodel.2006.04.023](https://doi.org/10.1016/j.ecolmodel.2006.04.023).
- Grimm V, Berger U, DeAngelis DL, Polhill JG, Giske J, Railsback SF (2010). “The ODD Protocol: A Review and First Update.” *Ecological Modelling*, **221**(23), 2760–2768. doi:[10.1016/j.ecolmodel.2010.08.019](https://doi.org/10.1016/j.ecolmodel.2010.08.019).
- Grimm V, Railsback SF (2005). *Individual-Based Modeling and Ecology*. Princeton University Press.
- Grimm V, Revilla E, Berger U, Jeltsch F, Mooij WM, Reilback SF, Thulke HH, Weiner J, Wiegand T, DeAngelis DL (2005). “Pattern-Oriented Modeling of Agent-Based Complex Systems: Lessons from Ecology.” *Science*, **310**(5750), 987–991. doi:[10.1126/science.1116681](https://doi.org/10.1126/science.1116681).
- Grisi-Filho JHH, Amaku M, Ferreira F, Dias RA, Ferreira-Neto JS, Negreiros RL, Ossada R (2013). “Detecting Livestock Production Zones.” *Preventive Veterinary Medicine*, **110**(3–4), 304–311. doi:[10.1016/j.prevetmed.2012.12.013](https://doi.org/10.1016/j.prevetmed.2012.12.013).
- Harvey N, Reeves A, Schoenbaum MA, Zagmutt-Vergara FJ, Dubé C, Hill AE, Corso BA, McNab WB, Cartwright CI, Salman MD (2007). “The *North American Animal Disease Spread Model*: A Simulation Model to Assist Decision Making in Evaluating Animal Disease Incursions.” *Preventive Veterinary Medicine*, **82**(3–4), 176–197. doi:[10.1016/j.prevetmed.2007.05.019](https://doi.org/10.1016/j.prevetmed.2007.05.019).
- Kane M, Emerson J, Weston S (2013). “Scalable Strategies for Computing with Massive Data.” *Journal of Statistical Software*, **55**(14), 1–19. doi:[10.18637/jss.v055.i14](https://doi.org/10.18637/jss.v055.i14).
- Kao RR, Green DM, Johnson J, Kiss IZ (2007). “Disease Dynamics over Very Different Time-Scales: Foot-and-Mouth Disease and Scrapie on the Network of Livestock Movements in the UK.” *Journal of the Royal Society Interface*, **4**(16), 907–916. doi:[10.1098/rsif.2007.1129](https://doi.org/10.1098/rsif.2007.1129).
- Keeling M (2005). “The Implications of Network Structure for Epidemic Dynamics.” *Theoretical Population Biology*, **67**(1), 1–8. doi:[10.1016/j.tpb.2004.08.002](https://doi.org/10.1016/j.tpb.2004.08.002).
- Longini, Jr IM, Halloran ME, Nizam A, Yang Y (2004). “Containing Pandemic Influenza with Antiviral Agents.” *American Journal of Epidemiology*, **159**(7), 623–633. doi:[10.1093/aje/kwh092](https://doi.org/10.1093/aje/kwh092).
- Marques FS, Grisi-Filho JHH, Amaku M (2020). *hybridModels: Stochastic Hybrid Models in Dynamic Networks*. R package version 0.3.7, URL <https://CRAN.R-project.org/package=hybridModels>.
- Microsoft Corporation, Weston S (2019). *doParallel: Foreach Parallel Adaptor for the parallel Package*. R package version 1.0.15, URL <https://CRAN.R-project.org/package=doParallel>.

- Microsoft Corporation, Weston S (2020). **foreach**: Provides Foreach Looping Construct for R. R package version 1.4.8, URL <https://CRAN.R-project.org/package=foreach>.
- Mills CE, Robins JM, Lipsitch M (2004). “Transmissibility of 1918 Pandemic Influenza.” *Nature*, **432**(7019), 904–906. doi:10.1038/nature03063.
- Monk N (2012). “Stochastic Models.” CPIB Summer School 2012: Introduction to Biological Modelling.
- Negreiros RL (2010). *Caracterização e Análise Da Rede De Movimento De Bovinos No Estado De Mato Grosso*. Ph.D. thesis, Universidade de São Paulo.
- Nöremark M, Håkansson N, Lewerin SS, Lindberg A, Jonsson A (2011). “Network Analysis of Cattle and Pig Movements in Sweden: Measures Relevant for Disease Control and Risk Based Surveillance.” *Preventive Veterinary Medicine*, **99**(2–4), 78–90. doi:10.1016/j.prevetmed.2010.12.009.
- Nöremark M, Håkansson N, Lindström T, Wennergren U, Lewerin SS (2009). “Spatial and Temporal Investigations of Reported Movements, Births and Deaths of Cattle and Pigs in Sweden.” *Acta Veterinaria Scandinavica*, **51**(1), 37. doi:10.1186/1751-0147-51-37.
- Nöremark M, Widgren S (2014). “**EpiContactTrace**: An R-Package for Contact Tracing during Livestock Disease Outbreaks and for Risk-Based Surveillance.” *BMC Veterinary Research*, **10**(1), 71. doi:10.1186/1746-6148-10-71.
- Osgood N (2007). “Using Traditional and Agent Based Toolsets for System Dynamics: Present Tradeoffs and Future Evolution.” In *The 2007 International Conference of the System Dynamics Society*.
- Pineda-Krch M (2008). “**GillespieSSA**: Implementing the Stochastic Simulation Algorithm in R.” *Journal of Statistical Software*, **25**(12), 1–18. doi:10.18637/jss.v025.i12.
- Rahmandad H, Sterman J (2008). “Heterogeneity and Network Structure in the Dynamics of Diffusion: Comparing Agent-Based and Differential Equation Models.” *Management Science*, **54**(5), 998–1014. doi:10.1287/mnsc.1070.0787.
- Rasamoelina-Andriamanivo H, Duboz R, Lancelot R, Maminiaina OF, Jourdan M, Rakoton-dramaro TMC, Rakotonjanahary SN, de Almeida RS, Rakotondravao, Durand B, Chevalier V (2014). “Description and Analysis of the Poultry Trading Network in the Lake Alaotra Region, Madagascar: Implications for the Surveillance and Control of Newcastle Disease.” *Acta Tropica*, **135**, 10–18. doi:10.1016/j.actatropica.2014.03.008.
- Rautureau S, Dufour B, Durand B (2011). “Vulnerability of Animal Trade Networks to The Spread of Infectious Diseases: A Methodological Approach Applied to Evaluation and Emergency Control Strategies in Cattle, France, 2005.” *Transboundary and Emerging Diseases*, **58**(2), 110–120. doi:10.1111/j.1865-1682.2010.01187.x.
- R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Shirley MDFF, Rushton SP (2005). “The Impacts of Network Topology on Disease Spread.” *Ecological Complexity*, **2**(3), 287–299. doi:10.1016/j.ecocom.2005.04.005.

- Soetaert K, Petzoldt T, Setzer RW (2010). “Solving Differential Equations in R: Package **deSolve**.” *Journal of Statistical Software*, **33**(9), 1–25. doi:[10.18637/jss.v033.i09](https://doi.org/10.18637/jss.v033.i09).
- Stevenson MA, Sanson RL, Stern MW, O’Leary BD, Sujau M, Moles-Benfell N, Morris RS (2013). “**InterSpread Plus**: A Spatial and Stochastic Simulation Model of Disease in Animal Populations.” *Preventive Veterinary Medicine*, **109**(1–2), 10–24. doi:[10.1016/j.prevetmed.2012.08.015](https://doi.org/10.1016/j.prevetmed.2012.08.015).
- Thakur KK, Revie CW, Hurnik D, Poljak Z, Sanchez J (2016). “Analysis of Swine Movement in Four Canadian Regions: Network Structure and Implications for Disease Spread.” *Transboundary and Emerging Diseases*, **63**(1), e14–e26. doi:[10.1111/tbed.12225](https://doi.org/10.1111/tbed.12225).
- van Rossum G, *et al.* (2011). *Python Programming Language*. URL <http://www.python.org>.
- Vincenot CE, Giannino F, Rietkerk M, Moriya K, Mazzoleni S (2011). “Theoretical Considerations on the Combined Use of System Dynamics and Individual-Based Modeling in Ecology.” *Ecological Modelling*, **222**(1), 210–218. doi:[10.1016/j.ecolmodel.2010.09.029](https://doi.org/10.1016/j.ecolmodel.2010.09.029).
- Vincenot CE, Moriya K (2011). “Impact of the Topology of Metapopulations on the Resurgence of Epidemics Rendered by a New Multiscale Hybrid Modeling Approach.” *Ecological Informatics*, **6**(3–4), 177–186. doi:[10.1016/j.ecoinf.2011.04.002](https://doi.org/10.1016/j.ecoinf.2011.04.002).
- Wickham H (2007). “Reshaping Data with the **reshape** Package.” *Journal of Statistical Software*, **21**(12), 1–20. doi:[10.18637/jss.v021.i12](https://doi.org/10.18637/jss.v021.i12).
- Wickham H (2009). **ggplot2: Elegant Graphics for Data Analysis**. Springer-Verlag. doi:[10.1007/978-0-387-98141-3](https://doi.org/10.1007/978-0-387-98141-3). URL <https://ggplot2.tidyverse.org/>.

A. Procedure to solve the ODE model

The procedure to solve the ODE model is presented below.

```
R> library("deSolve")
```

We then define the parameters, initial condition, and time sequence.

```
R> alpha <- 10/3
R> n <- 3
R> beta <- rep(0.006, n)
R> alpha.S1 <- c(alpha, 0, 0)
R> A <- matrix(c(-alpha, alpha, 0, 0, -alpha, alpha, 0, 0, -alpha),
+   nrow = n, ncol = n, byrow = TRUE)
R> init.si.md <- c(S = c(999, 1000, 1000), I = c(1, 0, 0))
R> time.seq <- seq(1, 10 * 12 * 30, 1)
```

The set of differential equations that describe the migration and disease dynamics is provided as a function.

```
R> sirmodel <- function(t, Y, p) {
+   S <- Y[1:n]
+   I <- Y[(n+1):(2*n)]
+   N <- S + I
+   dS <- - beta * S * I / N + (S / N) %% A + alpha.S1
+   dI <- + beta * S * I / N + (I / N) %% A
+   list(c(dS, dI))
+ }
```

To run the model using the **deSolve** package, we use the following code:

```
R> resultsODE <- as.data.frame(ode(times = time.seq, func = sirmodel,
+   y = init.si.md, parms = NULL))
```

Affiliation:

Fernando Silveira Marques
 Núcleo de Perícias Criminalísticas de Informática
 Instituto de Criminalística
 Superintendência da Polícia Técnico-Científica
 Rua Moncorvo Filho, 110, São Paulo, Brasil
 E-mail: fernando.fsm@policiacientifica.sp.gov.br

Journal of Statistical Software

published by the Foundation for Open Access Statistics

June 2020, Volume 94, Issue 6

[doi:10.18637/jss.v094.i06](https://doi.org/10.18637/jss.v094.i06)

<http://www.jstatsoft.org/>

<http://www.foastat.org/>

Submitted: 2016-10-07

Accepted: 2018-12-30
