Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Available online at www.sciencedirect.com
ScienceDirect
Computational Statistics & Data Analysis 52 (2007) 680–686
COMPUTATIONAL STATISTICS & DATA ANALYSIS
www.elsevier.com/locate/csda

RLadyBug—An R package for stochastic epidemic models

Michael Höhle*, Ulrike Feldmann

Department of Statistics, University of Munich, Ludwigstr. 33, 80539 Munich, Germany

Available online 4 December 2006

Abstract

RLadyBug is an S4 package for the simulation, visualization and estimation of stochastic epidemic models in R. Maximum likelihood and Bayesian inference can be performed to estimate the parameters in a susceptible-exposed-infectious-recovered (SEIR) model, which is a stochastic model for describing a single outbreak of an infectious disease. The package is thus one step towards statistical software supporting parameter estimation, calculation of confidence intervals and hypothesis testing for transmission models.

Keywords: SIR model; SEIR model; Stochastic modelling; MCMC; S4; R

1. Introduction

Understanding the dynamics and spread of infectious diseases is a key component in the design and analysis of defensive strategies. As a consequence, a multitude of epidemiological data on infectious diseases in animal, plant and human communities has been collected to gain insights into the underlying biological and epidemiological processes. The SIR (susceptible-infectious-recovered) model and its variants (S-Exposed-IR, SIS, etc.) are the mathematical tools most commonly used in such analyses.

This paper describes a software program for the statistical analysis of a single outbreak in a small population. Special focus is on the spatial spread between subpopulations arranged on a lattice. In veterinary or plant epidemiology such data arise from so-called transmission experiments, where one or more individuals in a controlled environment are inoculated with the infectious disease pathogen. Subsequently, the course of the epidemic is monitored through visual inspection, clinical testing and other methods. The aim ranges in veterinary epidemiology from quantifying disease transmission (Laevens et al., 1999; Stärk et al., 2000) to determining the effect of a vaccine (Dewulf et al., 2001; Meyns et al., 2004).

Because outbreaks induced by transmission experiments are planned and occur in a controlled environment, the produced outbreak data are especially rich. However, the mathematical setup of course also applies to the analysis of ordinary outbreaks. Interesting is also the application to entirely different areas such as the analysis of computer virus in a network (Wierman and Marchette, 2004), spread of the severe acute respiratory syndrome (SARS) (Donelly and Ghani, 2004) or outbreaks in the wards of a hospital (Grundmann and Hellriegel, 2006).

`RLadyBug` is a package implemented in R (R Development Core Team, 2006) providing functionality for the simulation, visualization and estimation in stochastic epidemic models. It enwraps the functionality of the Java program

* Corresponding author. Tel.: +49 89 2180 6404; fax: +49 89 2180 5040.
E-mail address: hoehle@stat.uni-muenchen.de (M. Höhle).

0167-9473/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2006.11.016

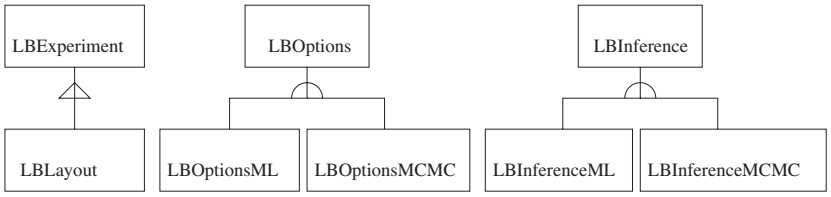

Fig. 1. The object hierarchy of the `RLadyBug` S4 classes.

used in Höhle et al. (2005) by S4 classes and adds a volume of methods for the visualization of outbreak data and their estimation results. The aim of this paper is to describe the features of the package in order to make it accessible to statisticians and statistically trained epidemiologists who are looking for software to analyse their infectious disease data. Less attention is thus given to the statistical particulars, which are explained in Höhle et al. (2005).

This paper is organized as follows: Section 2 gives a short introduction to stochastic epidemic models, Section 3 introduces the package and illustrates its use by providing the corresponding R code for analyzing a transmission experiment with classical swine fever virus (CSFV). Section 4 provides a discussion.

2. Stochastic epidemic models

With focus on the software dimension, only a short introduction to stochastic epidemic models in terms of the SEIR model is given. For a more thorough description see Andersson and Britton (2000).

A closed population P is hosted in k units. Each individual in P can be in one of the states susceptible, exposed, infectious or recovered. A spatial dimension is introduced by assuming that the k units are arranged in a $k = k_1 \times k_2$ lattice. At the beginning of the outbreak, $t = 0$, the number of susceptibles in each unit is $\boldsymbol{S}(0) = (n_1, \ldots, n_k)$, the number of exposed is $\boldsymbol{E}(0) = (m_1, \ldots, m_k)$ and the number of infectious is $\boldsymbol{I}(0) = (0, \ldots, 0)$. At time t, an individual j in unit u_j meets infectious at rate

$$\lambda_{u_j}(t) = \beta I_{u_j}(t) + \beta_n \sum_{u \in N(u_j)} I_u(t), \tag{1}$$

where $N(u_j)$ denotes the neighbors of u_j (e.g. in the four compass directions). Furthermore, β quantifies the within unit transmission rate, whilst β_n quantifies the transmission rate between neighboring units. If a susceptible meets an infectious it becomes exposed. After being exposed at time E_j an individual j has a gamma-distributed incubation time $T_E^j \sim \mathcal{G}a(\gamma_E, \delta_E)$ before becoming infectious, i.e. starting from time $I_j = E_j + T_E^j$, j can infect others. Similarly, the infectious period lasts for $T_I^j \sim \mathcal{G}a(\gamma_I, \delta_I)$ after which recovery occurs, hence $R_j = I_j + T_I^j$ labels the time of recovery.

Objective of analyzing outbreak data with SEIR models is the estimation of the parameters $\boldsymbol{\theta} = (\beta, \beta_n, \gamma_E, \delta_E, \gamma_I, \delta_I)$, which permit the effect of control measures to be studied and to generalize the results to different settings. `RLadyBug` is a tool to accomplish this ambition.

3. The `RLadyBug` package

Where applicable, `RLadyBug` uses the new S4 class system of R implemented in the `methods` package. This implies a more explicit class system requiring the programmer to define classes, slots and generics explicitly (Chambers, 1998). Fig. 1 shows the object oriented hierarchy of the package: The classes `LBExperiment` and `LBLayout` represent the data layer, `LBOptions` the algorithmic component and `LBInference` the results. With the help of the `rJava` package (Urbanek, 2006), this S4 framework is connected to the functionality of an underlying Java program performing the computationally intensive operations.

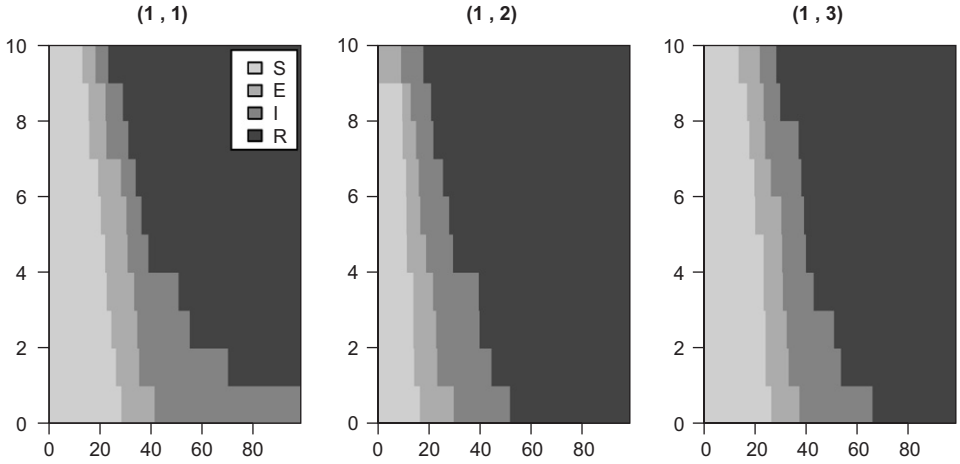

Fig. 2. $S_u(t)$, $E_u(t)$, $I_u(t)$ and $R_u(t)$ for the three units in a 1×3 lattice layout.

3.1. Simulation and visualization

To get an understanding of the SEIR model specified in Section 2, simulation based on the Sellke construction (Andersson and Britton, 2000) is utilized to generate data from the model. Code wise, this is accomplished by creating an object of class `LBExperiment` using the `simulate` function. Several visualizations of the experiment data can then be generated by `plot`, e.g. the `type = state ~ time | position` argument shows $S_u(t)$, $E_u(t)$ and $I_u(t)$ as a function of time for each unit u in the lattice. The below stated code creates a simulated epidemic in a 1×3 lattice, see Fig. 2. Initially $\boldsymbol{S}(0) = (10, 9, 10)$ and $\boldsymbol{E}(0) = (0, 1, 0)$.

```
> library("RLadyBug")
> layout <- new("LBLayout", S0 =matrix(c(10, 9, 10), 1, 3),
+                           E0 =matrix(c(0, 1, 0), 1, 3))
> options <- new("LBOptions", initBeta =list(init =0.125),
+  initBetaN =list(init =0.018), initIncu =list(g =6.697, d =0.84),
+  initInf =list(g =1.772, d =0.123))
> plot(simulate(options, layout =layout), type =state~time|position)
```

Further views of `LBExperiment` objects can be generated using `type = state~time` or `type = state~1 | position` as arguments in `plot`. The former shows $S(t)$, $E(t)$ and $I(t)$, with e.g. $S(t) = \sum_{u=1}^{k} S_u(t)$, the latter creates an animation illustrating the course of the epidemic by plotting the state given position for a set of fixed time points. As exemplification, the code below uses a 8×16 lattice with 15 individuals in each unit (a realistic setup for a pig farm) and creates a 20-picture animation of the epidemic. Fig. 3 shows the eighth picture of this sequence: Each stacked bar shows the current ($t = 58.9$ days) percentage of susceptible, exposed, infectious and recovered in the unit.

```
> E0 <- matrix(0, 8, 16)
> E0[4, 8] <- 1
> S0 <- matrix(15, 8, 16) - E0
> exp  <- simulate(options, layout =new("LBLayout", S0 =S0, E0 =E0))
> plot(exp, type =state~1|position, options =list(noOfPics =20))
```

3.2. Test data

Besides the ability to simulate data, the package contains several data sets from human and veterinary epidemiology. Amongst others are the Smallpox Epidemic in Abakaliki, Nigeria, analyzed e.g. in Andersson and Britton (2000) or O'Neill and Becker (2001), and the data from CSFV experiments by Laevens et al. (1999) and Dewulf et al. (2001).

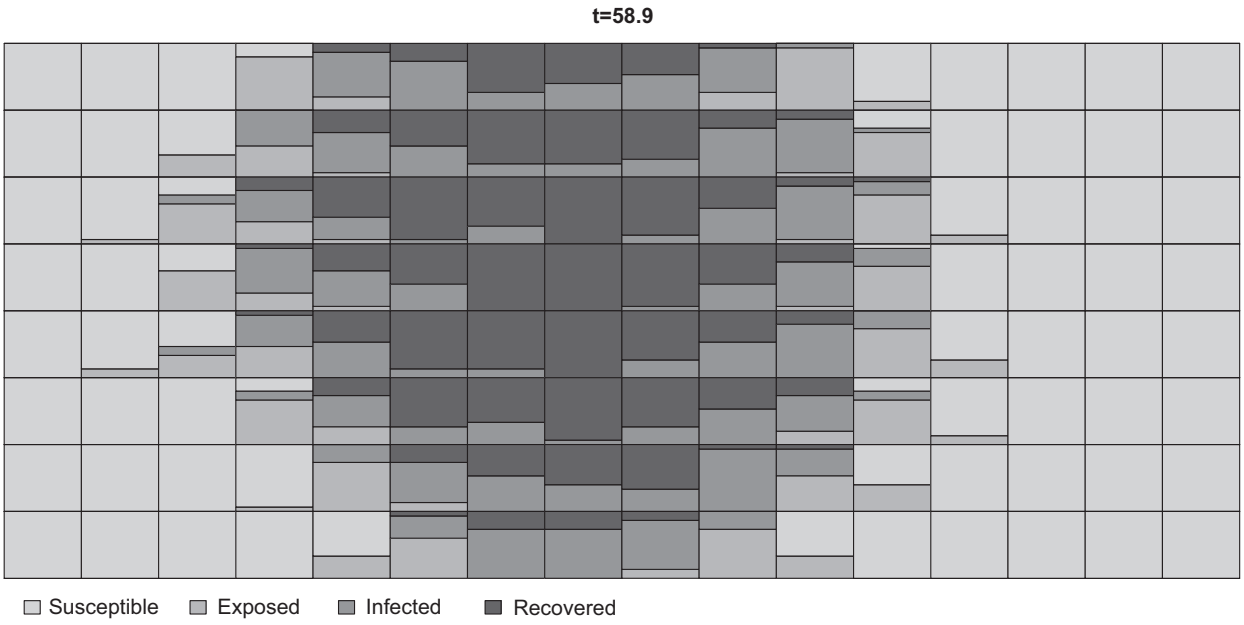

Fig. 3. Snapshot of the animation illustrating the outbreak in a 8×16 lattice.

The next section describes a statistical analysis of the experiment by Laevens et al. (1999) using `RLadyBug`. In this experiment the spread of CSFV was investigated in a 1×3 layout with $\boldsymbol{S}(0) = (5, 5, 6)$ and $\boldsymbol{E} = (0, 1, 0)$ slaughter pigs. Every second day all pigs still alive were investigated using a virus isolation test based on blood plasma.

3.3. Analysis of an CSFV transmission experiment

For full data, i.e. with known E_j, I_j, R_j event times for all individuals, `RLadyBug` provides likelihood or Bayesian inference for $\boldsymbol{\theta} = (\beta, \beta_n, \gamma_E, \delta_E, \gamma_I, \delta_I)$. Details about the underlying equations leading to the log-likelihood and posterior distribution can be found in Höhle et al. (2005).

In practice, however, E_j is unobservable. A typical assumption is thus to assume a fixed and known incubation time c and hence compute E_j as $E_j = I_j - c$. This was also done in the CSFV experiment by assuming $c = 6$. With this assumption all event times are known and can be illustrated as in Fig. 4:

```
> data("laevensML")
> plot(laevensML, type = individual~time|position)
```

The individual 12:01 was inoculated at time $t = 0$. By maximization of the log-likelihood the maximum likelihood estimators are readily determined using `RLadyBug`. Nonetheless, assuming a fixed and known incubation time is not very realistic. Höhle et al. (2005) therefore use a Bayesian framework to handle the missing but gamma-distributed exposure times. Unknown (or censored) waiting times $T_E^j \sim \mathcal{G}a(\gamma_E, \delta_E)$ are imputed and updated through a Gibbs-within-Metropolis–Hastings Markov Chain Monte Carlo (MCMC) algorithm.

A Bayesian-analysis of the CSFV data with unknown exposure times could be conducted by the following code:

```
> data("laevens")
> inf.mcmc <- seir(laevens, laevens.opts)
```

Instead of creating the necessary `LBOptionsMCMC` object by hand, the call to `data` also loads an appropriate object `laevens.opts` for MCMC estimation. For example:

```
> algo(laevens.opts)

samples          thin          burnin
   2500            25           50000
```

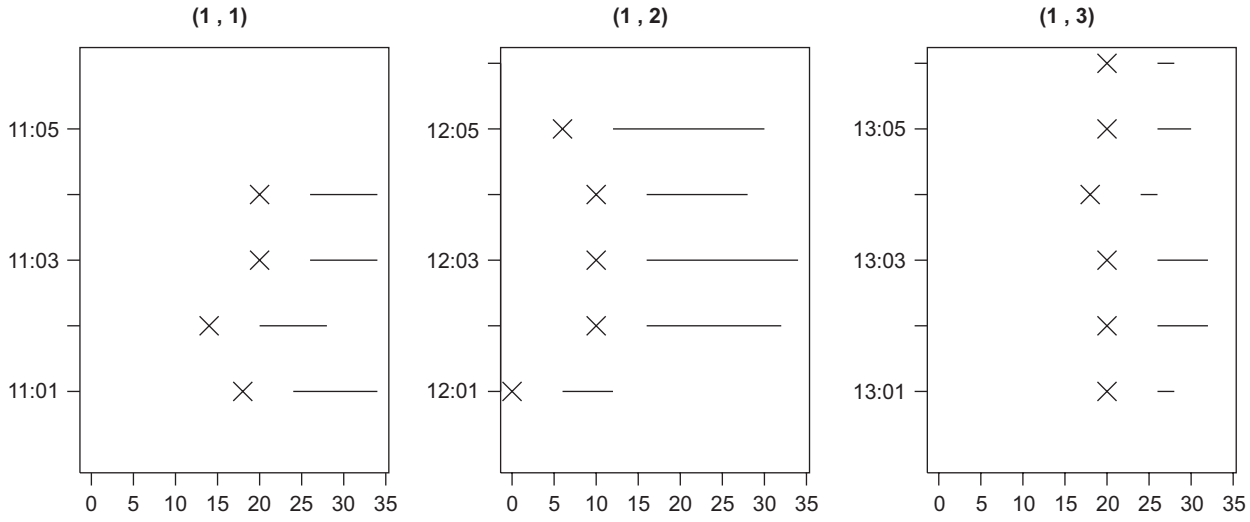

Fig. 4. Infectious period of each individual ($xy : id$) in the CSFV example. Crosses denote the time of exposure (in days), lines connect the $I_{xy:id}$ and $R_{xy:id}$ events.

shows the requested number of samples to draw from the posterior together with the burn-in and thinning rate; a total of `samples*thin+burnin` samples are generated. The results are as follows.

```
> inf.mcmc
An object of class LBInferenceMCMC

Parameter Estimations (posterior mean from 2500 samples):
  Parameter:
     beta     betaN     gammaE    deltaE    gammaI    deltaI
0.03706   0.02837   56.82000   9.37400   2.16200   0.25640

StandardErrors (posterior std.dev. from 2500 samples):
     beta     betaN     gammaE    deltaE    gammaI    deltaI
 0.018500  0.009481 45.510000 7.761000  0.738100  0.097760
```

The results of the MCMC inference are provided as realizations from the Markov chain having the posterior distribution of $\boldsymbol{\theta}$ as stationary distribution. Access to the samples is obtained through the `samplePaths` method—this makes allowance for further processing using e.g. the `coda` (convergence diagnostic and output analysis) or `boa` (Bayesian output analysis) packages from the Comprehensive R Archive Network (Plummer et al., 2006; Smith, 2005). To exemplify, the sampling paths and the marginal posterior of β are inspected in Fig. 5 by

```
> samples <- mcmc(samplePaths(inf.mcmc))
> plot(samples[, "beta"])
```

A matter of particular interest in the CSFV experiment is the relationship between β and β_n—especially whether there is a difference in spread within the pen and between neighboring pens. The following code uses the `plot` method for `LBInferenceMCMC` objects to generate Fig. 6 and to display the posterior mean together with lower and upper boundaries of a 95%-highest posterior density (HPD) interval for β/β_n.

```
> betabetaN <- plot(inf.mcmc, which ="betabetaN")
> c(mean =betabetaN$mean, betabetaN$hpd)

     mean    LB 95% HPD    UB 95% HPD
1.4740612     0.1245430     3.3158915
```

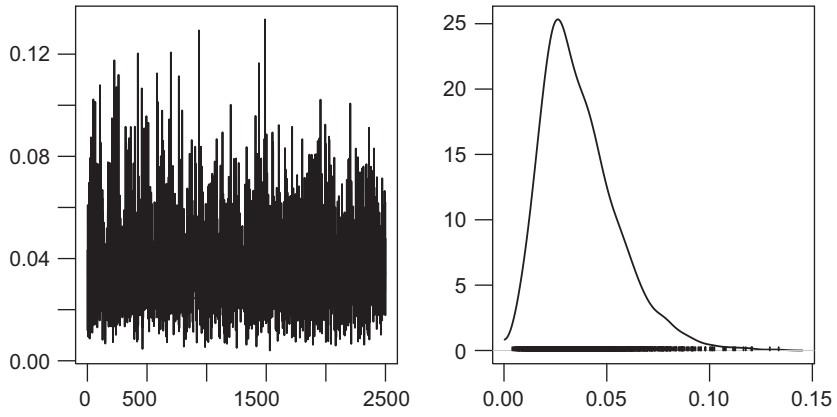

Fig. 5. Sample path and kernel density of the β-samples generated using `coda`.

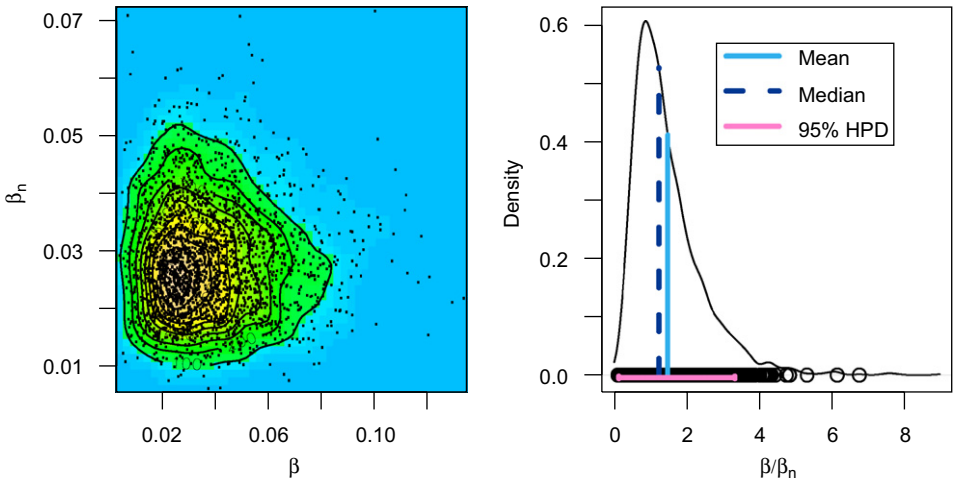

Fig. 6. Left panel: 2D-Kernel estimated posterior density surface of (β, β_n). Right panel: 1D-Kernel estimate of the posterior density of β/β_n together with posterior mean, posterior median and a 95%-HPD interval.

An important epidemiological quantity of an infectious disease is the basic reproduction ratio R_0, i.e. the expected number of new infections generated by a single infectious individual in a large susceptible population. In a multitype setup with k groups this can be computed as the largest eigenvalue of the $k \times k$ matrix containing the mean number of infectious contacts between all units (Andersson and Britton, 2000, Section 6.2). Samples from the posterior distribution of R_0 are generated by performing this computation for each posterior sample of $\boldsymbol{\theta}$. The code stated below uses the method `R0` to retrieve the samples for R_0 and thus to compute the posterior median together with a symmetric 95% credibility region.

```
> quantile(R0(inf.mcmc, laevens), c(0.025, 0.5, 0.975))

     2.5%       50%     97.5%
0.3245193 0.6370434 1.2426485
```

4. Discussion

In this paper we have illustrated the use of `RLadyBug` for the simulation, visualization and estimation of infectious disease outbreak data. To our knowledge, the package is the first publicly available software for the estimation in

stochastic epidemic models. By providing such specialist functionality within a standard software package as R we hope to make a thorough statistical analysis of infectious disease data a bit more routine.

Many extensions towards more complex and realistic models than provided by the package are imaginable. For example, the general multitype SEIR model (Andersson and Britton, 2000, Chapter 6) allows for more complicated neighbor dependencies: Letting $\boldsymbol{\Lambda}(\boldsymbol{\alpha})$ be a $k \times k$ matrix given as a function of the parameter vector $\boldsymbol{\alpha}$, the transmission rates are given by $\boldsymbol{\lambda}(t) = \boldsymbol{\Lambda}(\boldsymbol{\beta})\boldsymbol{I}(\beta_{ij})'$. Nearest neighborhood transmission as in (1) thus corresponds to $\Lambda_{ij} = \beta I(u_i = u_j) + \beta_n I(u_j \in N(u_j))$ and hence $\boldsymbol{\alpha} = (\beta, \beta_n)$. An alternative would be to let transmission from neighbors be a function of distance: $\Lambda_{ij} = \alpha_1 \exp(-\alpha_2 \operatorname{dist}(u_i, u_j))$. Modifications of the Java code to handle the simulation and maximum likelihood estimation in such models should be feasible. However, obtaining MCMC estimates in case of missing data would require a substantial amount of work. The same comment applies if one wants to extend beyond the currently implemented waiting time distributions (gamma-distributed, exponential-distributed and constant).

The multitype model could also be applied to the handling of heterogeneous units exemplified by veterinary experiments quantifying the effect of a vaccine. Here, all individuals in specific units are vaccinated, thus having different parameters than individuals in non-vaccinated units. A Monte-Carlo approach could be employed to calculate sample sizes necessary to detect a certain difference in parameters with a given accuracy.

In addition, we are currently working on an implementation of the logistic-regression approach in Klinkenberg et al. (2002). This extension illustrates the benefits of providing a flexible and extensible package: the code is purely R-based exploiting the class structure of the package, while using the optimization routines of R for inference.

Sources, binaries and documentation of `RLadyBug` are available for download from the Comprehensive R Archive Network http://cran.r-project.org/ under the GNU Public License. Once installed, the analyses of this article can be reproduced using `demo("article-csda")`.

Acknowledgments

We thank Jeroen Dewulf, University of Ghent, Belgium, for providing us with the transmission experiment data of Laevens et al. (1999) and Dewulf et al. (2001). The research was conducted with financial support from the Collaborative Research Centre SFB 386 funded by the German research foundation (DFG).

References

Andersson, H., Britton, T., 2000. Stochastic epidemic models and their statistical analysis. Springer Lectures Notes in Statistics, vol. 151, Springer, Berlin.

Chambers, J.M., 1998. Programming with Data—A Guide to the S Language. Springer.

Dewulf, J., Laevens, H., Koenen, F., Vanderhallen, H., Mintiens, K., 2001. An experimental infection with classical swine fever in E2 sub-unit marker-vaccine vaccinated and in non-vaccinated pigs. Vaccine 19, 475–482.

Donelly, C., Ghani, A., 2004. Real-time epidemiology—understanding the spread of SARS. Significance 1, 176–179.

Grundmann, H., Hellriegel, B., 2006. Mathematical modelling: a tool for hospital infection control. Lancet Infect. Dis. 6, 39–45.

Höhle, M., Jørgensen, E., O'Neill, P., 2005. Inference in disease transmission experiments by using stochastic epidemic models. J. Roy. Statist. Soc. Ser. C 54 (2), 349–366.

Klinkenberg, D., de Bree, J., Laevens, H., de Jong, M., 2002. Within- and between-pen transmission of classical swine fever virus: a new method to estimate the basic reproduction ratio from transmission experiments. Epidemiol. Infect. 128 (2), 293–299.

Laevens, H., Koenen, F., Deluyker, H., de Kruif, A., 1999. Experimental infection of slaughter pigs with classical swine fever virus: transmission of the virus, course of the disease and antibody response. Vet. Rec. 145, 243–248.

Meyns, T., Maes, D., Dewulf, J., Vicca, J., Haesebrouck, F., de Kruif, A., 2004. Quantification of the spread of Mycoplasma hyopneumonia in nursery pigs using transmission experiments. Prev. Vet. Med. 66, 265–275.

O'Neill, P.D., Becker, N.G., 2001. Inference for an epidemic when susceptibility varies. Biostatistics 2 (1), 99–108.

Plummer, M., Best, N., Cowles, K., Vines, K., 2006. coda: output analysis and diagnostics for MCMC. R Package Version 0.10-5.

R Development Core Team, 2006. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Smith, B.J., 2005. boa: Bayesian Output Analysis Program (BOA) for MCMC. R Package Version 1.1.5-2.

Stärk, K., Pfeiffer, D., Morris, R., 2000. Within-farm spread of classical swine fever virus—a blueprint for a stochastic simulation model. Vet. Quarterly 22 (1), 36–43.

Urbanek, S., 2006. rJava: Low-level R to Java interface. R Package Version 0.4-3.

Wierman, J., Marchette, D., 2004. Modeling computer virus prevalence with susceptible-infected-susceptible model with reintroduction. Comput. Statist. Data Anal. 45, 3–23.