Bayesian inference for a population growth model of the chytrid fungus

Philipp H Boersch-Supan and Leah R Johnson

June 04, 2016

1 Preliminaries

This examples assumes that `deBInfer` is installed and loaded. If this is not the case you need to install it from github, which requires the devtools package.

```r
install.packages("devtools")
```

```r
#Load the devtools package.
library(devtools)
```

Then you install deBInfer from github

```r
install_github("pboesu/debinfer")
library(deBInfer)
```

2 The chytrid population growth model

Our example demonstrates parameter inference for a DDE model of population growth in the environmentally sensitive fungal pathogen *Batrachochytrium dendrobatidis* (Bd), which causes the amphibian disease chytridiomycosis (Rosenblum et al. 2010; Voyles et al. 2012). This model has been used to further our understanding of pathogen responses to changing environmental conditions. Further details about the model development, and the experimental procedures yielding the data used for parameter inference can be found in (Voyles et al. 2012).

The model follows the dynamics of the concentration of an initial cohort of zoospores, C, the concentration of zoospore-producing sporangia, S, and the concentration of zoospores in the next generation Z. The initial cohort of zoospores, C, starts at a known concentration, and zoospores in this initial cohort settle and become sporangia at rate s_r, or die at rate μ_Z. f_s is the fraction of sporangia that survive to the zoospore-producing stage. We assume that it takes a minimum of T_{min} days before the sporangia produce zoospores, after which they produce zoospores at rate η. Zoospore-producing sporangia die at rate d_s. The concentration of zoospores, Z, is the only state variable measured in the experiments, and it is assumed that these zoospores settle (s_r) or die (μ_Z) at the same rates as the initial cohort of zoospores.

2.1 DDE model

The equations that describe the population dynamics are as follows:

$$\frac{dC}{dt} = -(s_r + \mu_Z)C(t) \tag{1}$$

$$\frac{dS}{dt} = s_r f_s C(t - T_{min}) - d_s S(t) \tag{2}$$

$$\frac{dZ}{dt} = \eta S(t) - (s_r + \mu_Z)Z(t) \tag{3}$$

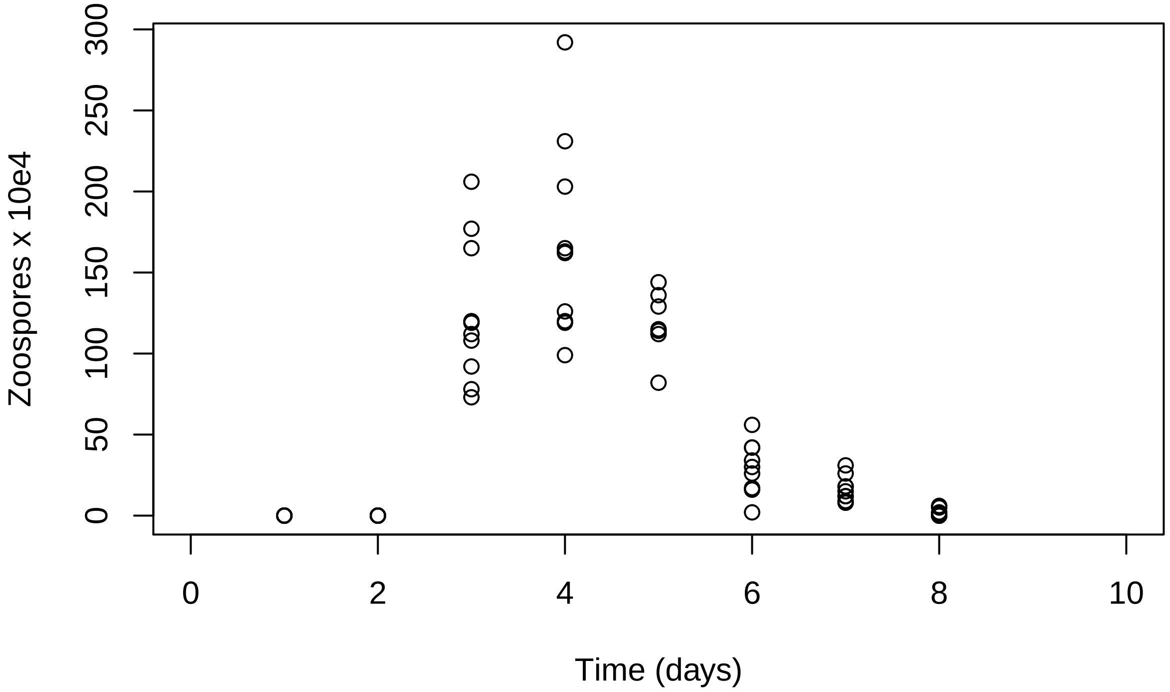
We can implement this system of differential equations for the `deSolve::dede` solver as follows. More details on how to specify differential equations for this solver can be found in the package documentation and vignettes (Soetaert, Petzoldt, and Setzer 2010).

```
#dede version
CSZ.dede<-function(t,y,p){

  sr    <-p["sr"]
  fs    <-p["fs"]
  ds    <-p["ds"]
  eta   <-p["eta"]
  Tmin  <-p["Tmin"]
  ##Tmax  <-p["Tmax"]
  muz <-p["muz"]

  Rs<-Ms<-0
  lag1<-lag2<-0

  if (t>Tmin){
    lag1<-lagvalue(t-Tmin)
    Rs <- sr*fs*lag1[1]
  }

  phiZ <- eta*y[2]
  dy1 <- -(muz+sr)*y[1]
  dy2 <- Rs - Ms - ds*y[2]
  dy3 <- phiZ - (muz+sr)*y[3]

  if(y[1]<0) dy1<-0
  if(y[2]<0){
    dy2 <- Rs - Ms
    dy3 <- -(muz+sr)*y[3]
  }
  if(y[3]<0){
    dy3 <- dy3+(muz+sr)*y[3]
  }

  list(c(dy1,dy2,dy3))
}
```

2.2 Observation model

Eventhough the data used in this example come from an experimental study, the system is only partially observed. We know the initial conditions for all states, but we only have observations for the second generation of Zoospores Z. Because the observations are counts (i.e. discrete numbers), we assume that observations of the system at a set of discrete times t' are independent Poisson random variables with a mean given by the solution of the DDE, at times t'.

```r
#load chytrid data
data(chytrid)
plot(chytrid, xlab='Time (days)', ylab='Zoospores x 10e4', xlim=c(0,10))
```



The log-likelihood of the data given the parameters, underlying model, and initial conditions is then a sum over the n observations at each time point in t'

$$\ell(\mathbf{Z}|\theta) = \sum_{t}^{n} Z_t \log \lambda - n\lambda \tag{4}$$

```r
## ----obs-model-----------------------------------------------------------
## observation model
chytrid_obs_model<-function(data, sim.data, samp){

  #z.temp<-sim.data$Z

  ##ec<-0.00001
  ec<-0.01
  llik.Z<-0
  for(i in unique(data$time)){
    try(llik.Z<-llik.Z + sum(dpois(data$count[data$time==i], lambda=(sim.data$Z[sim.data$time==i]+ec), 
  }
  llik<-llik.Z
  return(llik)
}
```

```r
## ----pars, results="hide", message=FALSE---------------------------------
sr <- debinfer_par(name = "sr", var.type = "de", fixed = FALSE,
                   value = 2, prior="gamma", hypers=list(shape = 5, rate = 1),
                   prop.var=0.5, samp.type="rw")

fs <- debinfer_par(name = "fs", var.type = "de", fixed = FALSE,
                   value = 0.5, prior="beta", hypers=list(shape1 = 1, shape2 = 1),
                   prop.var=0.05, samp.type="ind")

ds <- debinfer_par(name = "ds", var.type = "de", fixed = FALSE,
                   value = 2, prior="gamma", hypers=list(shape = 1, rate = 1),
                   prop.var=0.1, samp.type="rw")

muz <- debinfer_par(name = "muz", var.type = "de", fixed = FALSE,
                    value = 1, prior="gamma", hypers=list(shape = 5, rate = 1),
                    prop.var=0.1, samp.type="rw")

eta <- debinfer_par(name = "eta", var.type = "de", fixed = FALSE,
                    value = 10, prior="gamma", hypers=list(shape = 1, rate = 0.25),
                    prop.var=5, samp.type="rw")

Tmin <- debinfer_par(name = "Tmin", var.type = "de", fixed = FALSE,
                     value = 3, prior="unif", hypers=list(min = 2, max = 6),
                     prop.var=0.2, samp.type="rw")


## ----inits---------------------------------------------------------------
C <- debinfer_par(name = "C", var.type = "init", fixed = TRUE, value = 120)
S <- debinfer_par(name = "S", var.type = "init", fixed = TRUE, value = 0)
Z <- debinfer_par(name = "Z", var.type = "init", fixed = TRUE, value = 0)

## ----setup---------------------------------------------------------------
mcmc.pars <- setup_debinfer(sr, fs, ds, muz, eta, Tmin, C, S, Z)

## ----deBinfer, results="hide"--------------------------------------------
# do inference with deBInfer
# MCMC iterations
iter = 2000
# inference call
dde_plot <- microbenchmark::microbenchmark(

dde_rev = de_mcmc(N = iter, data=chytrid, de.model=CSZ.dde,
                      obs.model=chytrid_obs_model, all.params=mcmc.pars,
                      Tmax = max(chytrid$time), data.times=c(0,chytrid$time), cnt=50,
                      plot=TRUE, sizestep=0.1, which=2, verbose = TRUE),

dede_rev <- de_mcmc(N = iter, data=chytrid, de.model=CSZ.dede,
                               obs.model=chytrid_obs_model, all.params=mcmc.pars,
                               Tmax = max(chytrid$time), data.times=c(0,chytrid$time), cnt=50,
                               plot=FALSE, sizestep=0.1, which="dede", verbose = TRUE),
times = 10)

dde_noplot <- microbenchmark::microbenchmark(
  dde_old = de_mcmc(N = iter, data=chytrid, de.model=CSZ.dde,
                    obs.model=chytrid_obs_model, all.params=mcmc.pars,
                    Tmax = max(chytrid$time), data.times=c(0,chytrid$time), cnt=50,
                    burnin=0.1, plot=FALSE, sizestep=0.1, which=2, verbose = TRUE),


  dde_rev = de_mcmc_rev(N = iter, data=chytrid, de.model=CSZ.dde,
                        obs.model=chytrid_obs_model, all.params=mcmc.pars,
                        Tmax = max(chytrid$time), data.times=c(0,chytrid$time), cnt=50,
                        burnin=0.1, plot=FALSE, sizestep=0.1, which=2, verbose = TRUE),

  dede_rev <- de_mcmc_rev(N = iter, data=chytrid, de.model=CSZ.dede,
                          obs.model=chytrid_obs_model, all.params=mcmc.pars,
                          Tmax = max(chytrid$time), data.times=c(0,chytrid$time), cnt=50,
                          burnin=0.1, plot=FALSE, sizestep=0.1, which="dede", verbose = TRUE),
  times = 10)
par(mfrow=c(1,2))
plot_benchmark(dde_noplot, expr.levels = c("dde", "dde-rev", "dede"))
plot_benchmark(dde_plot, expr.levels = c("dde", "dde-rev", "dede"))
saveRDS(list(plot = dde_plot, noplot = dde_noplot, session = sessionInfo()), file="sandbox/benchmarks-de

## ----solve-dde------------------------------------------------------------
pars <- colMeans(mcmc_samples$samps)
names(pars) <- names(mcmc_samples$samps)

voylesfit <- dde(y=c(C=120,S=0,Z=0), times = seq(0,10,by=0.02), func=CSZ.dde, parms=pars )


#Sim CIs
library(coda)
library(plyr)
parsamps <- mcmc_samples$samps[sample(nrow(mcmc_samples$samps), size=500),]



siml <- alply(parsamps, 1, function(x){spars <- as.numeric(x)
names(spars) <- names(parsamps)
dde(y=c(120,0,0), times = seq(0,10,by=0.02), func=CSZ.dde, parms=spars)}, .progress='text')


plot(chytrid, xlim=c(0,9))
l_ply(siml, .fun=function(x){lines(x$time, x$y1, col='red3')})

l_ply(siml, .fun=function(x){lines(x$time, x$y2, col='blue3')})

l_ply(siml, .fun=function(x){lines(x$time, x$y3, col='green3')})

#figure
library(viridis)

#set colors
colours <- viridis(9)[c(1,4,7)]
```

```
with(voylesfit,{
  lwd=3
  plot(time,C, type='l', ylim=c(0,300),col=colours[1],lwd=lwd,ylab="Counts x 10^4", xlab="Time (days)")
  lines(time,Z,col=colours[2],lwd=lwd)
  lines(time,S,col=colours[3],lwd=lwd, lty=2)
  legend("topright", legend=c("C","Z","S", "Z_obs"), lwd=3, col=c(colours[c(1,2,3)],"black"), lty=c(1,1
  points(data_obs)
}
)
```

References

Rosenblum, Erica Bree, Jamie Voyles, Thomas J Poorten, and Jason E Stajich. 2010. "The Deadly Chytrid Fungus: A Story of an Emerging Pathogen." *PLoS Pathogens* 6 (1): 1–3. doi:10.1371/journal.ppat.1000550.

Soetaert, KER, Thomas Petzoldt, and R Woodrow Setzer. 2010. "Solving Differential Equations in R: Package DeSolve." *Journal of Statistical Software* 33 (9): 1–25.

Voyles, Jamie, Leah R Johnson, Cheryl J Briggs, Scott D Cashins, Ross A Alford, Lee Berger, Lee F Skerratt, Rick Speare, and Erica Bree Rosenblum. 2012. "Temperature Alters Reproductive Life History Patterns in Batrachochytrium Dendrobatidis, a Lethal Pathogen Associated with the Global Loss of Amphibians." *Ecology and Evolution* 2 (9). Wiley Online Library: 2241–49.