

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.

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

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

Then you install deBInfer from github

```
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:

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

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

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

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

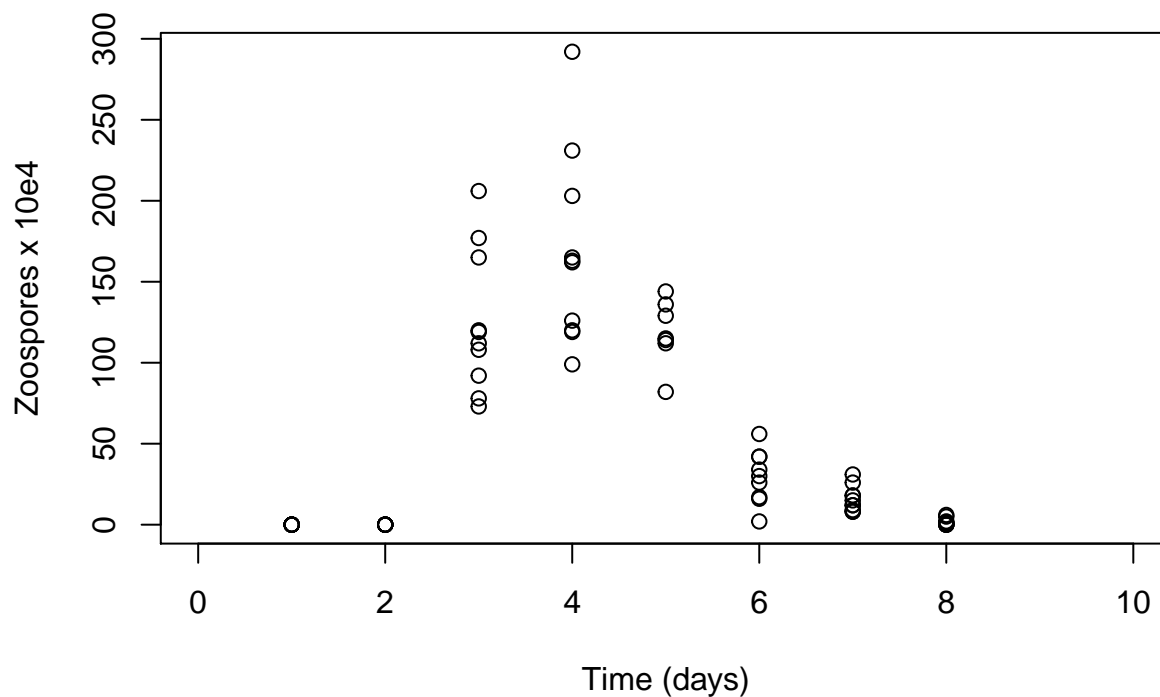
of Zoospores Z . Because the observations are counts (i.e. discrete numbers), we assume that observations of the system at a set of discrete times t' are independent Poisson random variables with a mean given by the solution of the DDE, at times t' .

The observations are provided with `deBInfer`. They can be loaded with the `data()` command.

```
#load chytrid data
data(chytrid)
#have a look at the variables
head(chytrid)
```

```
##   time count
## 1    1     0
## 2    1     0
## 3    1     0
## 4    1     0
## 5    1     0
## 6    1     0
```

```
#plot the data
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'

$$\ell(\mathbf{Z}|\theta) = \sum_t^n Z_t \log \lambda - n\lambda \quad (4)$$

Explain epsilon correction

This can be translated into an observation model function for `deBInfer`. The observation model function must have three named arguments `data`, `sim.data`, and `samp`, as these are used by the MCMC procedure

to pass in the data, the current state of the Markov chain, and the associated DE model solution. We can access these inputs to define the data likelihood. In this case we have repeat measurements for each time point, so we iterate over the unique timepoints in `data$time`, and then calculate the sum log-likelihood over all matching `data$count` observations using the current value of the state variable `Z` from the DE model solution at this point in the Markov chain.

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

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

We continue by defining the parameters for inference

```
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)
```

And then do the actual inference

```

## ----deBInfer, results="hide"-----
# do inference with deBInfer
# MCMC iterations
iter = 200
# inference call
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, solver="dede", verbose = TRUE)

## [1] "Order of initial conditions is C" "Order of initial conditions is S"
## [3] "Order of initial conditions is Z"
## [1] "initial posterior probability = -13968.8205931737"
## [1] "proposal outside prior support for muz = -0.257994567889461"
## [1] "proposal outside prior support for Tmin = 1.97985281680766"
## [1] "proposal outside prior support for Tmin = 1.99802618244234"
## [1] "proposal outside prior support for muz = -0.029718332396355"
## [1] "proposal outside prior support for muz = -0.684930229934571"
## [1] "proposal outside prior support for muz = -0.107499717322304"
## [1] "proposal outside prior support for muz = -0.173230000444132"
## [1] "proposal outside prior support for muz = -0.239265983944054"
## [1] "proposal outside prior support for muz = -0.0653305891057693"
## [1] "proposal outside prior support for muz = -0.0101797805300531"
## [1] "proposal outside prior support for muz = -0.142995560826004"
## [1] "proposing muz: prob.old = -514.33; prob.new = -514.33; A = 0"
## [1] "proposing Tmin: prob.old = -514.33; prob.new = -761.64; A = 3.8996e-108"
## [1] "proposing eta: prob.old = -514.33; prob.new = -518.04; A = 0.024313"
## [1] "proposing ds: prob.old = -514.33; prob.new = -670.47; A = 1.5371e-68"
## [1] "proposing sr: prob.old = -514.33; prob.new = -2301.4; A = 0"
## [1] "proposing fs: prob.old = -514.33; prob.new = -5762; A = 0"
## [1] "sample number 50"
## [1] "proposal outside prior support for muz = -0.03573615146259"
## [1] "proposal outside prior support for Tmin = 1.57719803840932"
## [1] "proposal outside prior support for muz = -0.402098785880582"
## [1] "proposal outside prior support for Tmin = 1.65466711610711"
## [1] "proposal outside prior support for muz = -0.0604571884437843"
## [1] "proposal outside prior support for muz = -0.208946339965845"
## [1] "proposal outside prior support for Tmin = 1.60262033143469"
## [1] "proposal outside prior support for muz = -0.219768920250713"
## [1] "proposal outside prior support for muz = -0.207822990520223"
## [1] "proposal outside prior support for muz = -0.172648550488712"
## [1] "proposal outside prior support for muz = -0.0571156616565814"
## [1] "proposal outside prior support for muz = -0.310165230423611"
## [1] "proposal outside prior support for muz = -0.181474511384289"
## [1] "proposal outside prior support for muz = -0.0292713238165734"
## [1] "proposal outside prior support for muz = -0.218857937588714"
## [1] "proposing eta: prob.old = -456.72; prob.new = -481.96; A = 1.0927e-11"
## [1] "proposing Tmin: prob.old = -456.72; prob.new = -11456; A = 0"
## [1] "proposing fs: prob.old = -456.72; prob.new = -460.62; A = 0.020208"
## [1] "proposing ds: prob.old = -456.72; prob.new = -462.38; A = 0.0034881"
## [1] "proposing sr: prob.old = -456.72; prob.new = -1212.8; A = 0"
## [1] "proposing muz: prob.old = -456.72; prob.new = -518.49; A = 1.4884e-27"
## [1] "sample number 100"

```

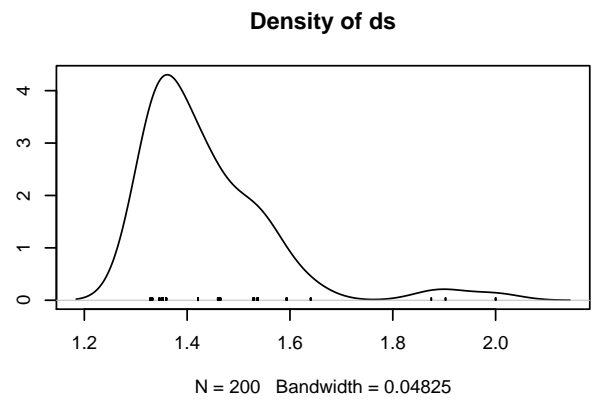
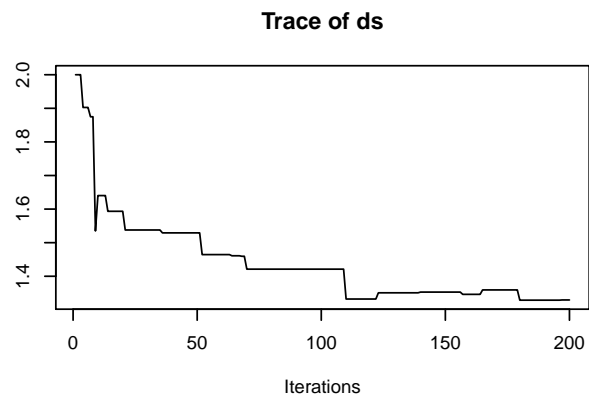
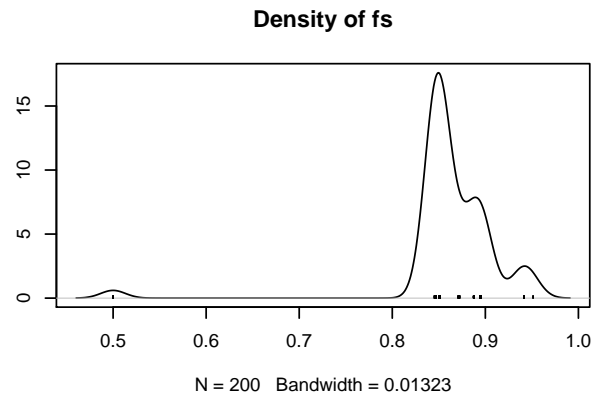
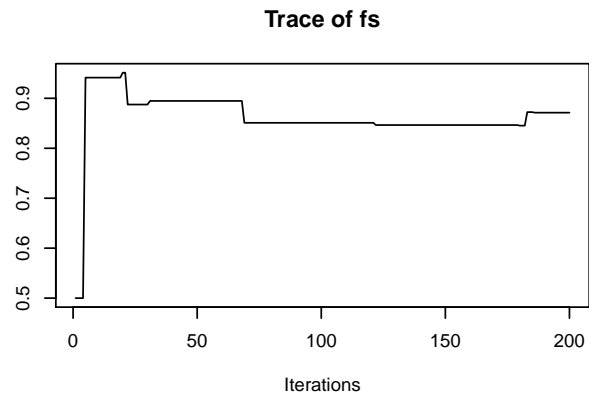
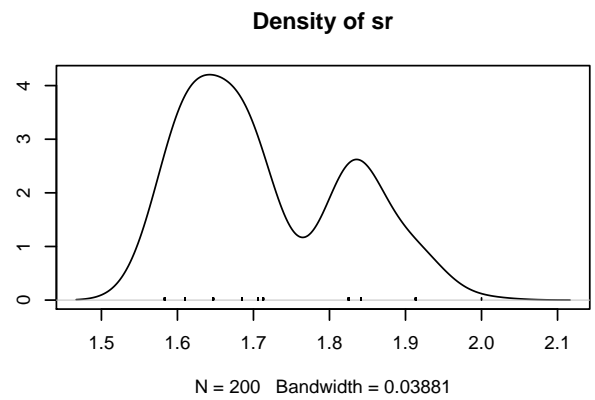
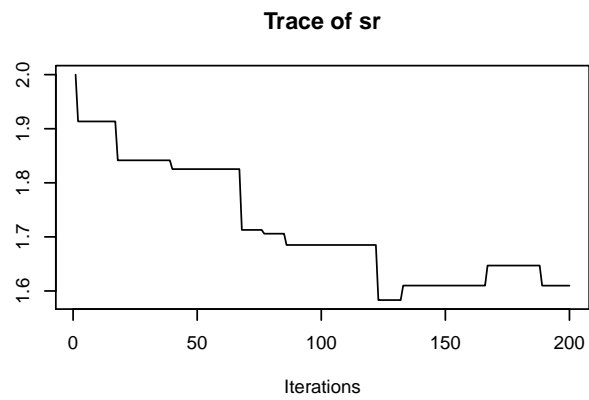
```

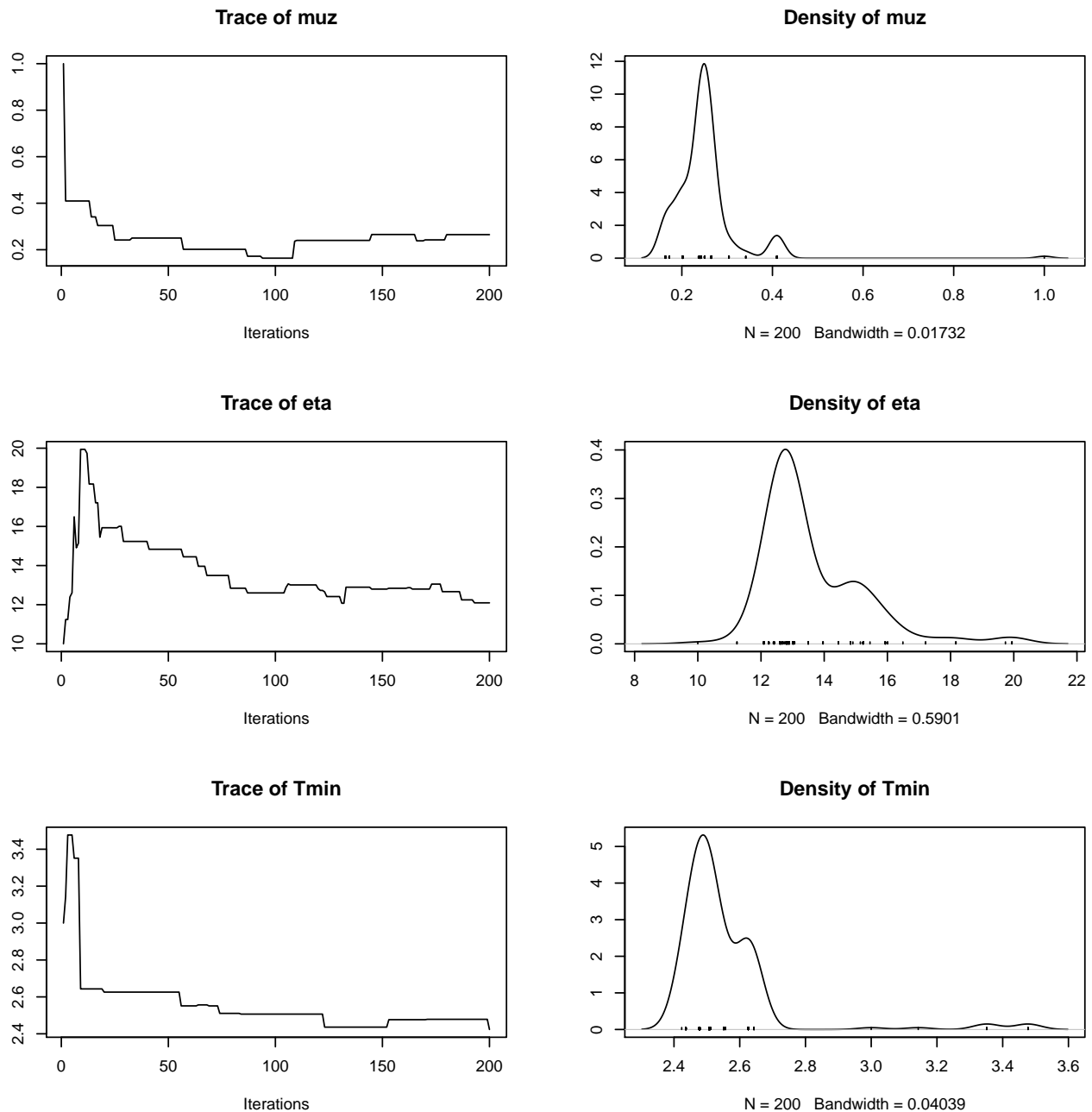
## [1] "proposal outside prior support for Tmin = 1.93828622157667"
## [1] "proposal outside prior support for Tmin = 1.82714990023146"
## [1] "proposal outside prior support for muz = -0.526971474051079"
## [1] "proposal outside prior support for Tmin = 1.93761686163815"
## [1] "proposal outside prior support for muz = -0.0397699116114231"
## [1] "proposal outside prior support for muz = -0.418699249522975"
## [1] "proposal outside prior support for Tmin = 1.92608025013557"
## [1] "proposal outside prior support for Tmin = 1.99615386358036"
## [1] "proposal outside prior support for Tmin = 1.86260465582754"
## [1] "proposal outside prior support for Tmin = 1.95415189313596"
## [1] "proposal outside prior support for muz = -0.0506227776003035"
## [1] "proposal outside prior support for muz = -0.0929332744134846"
## [1] "proposal outside prior support for Tmin = 1.75839472675466"
## [1] "proposal outside prior support for Tmin = 1.88974013400067"
## [1] "proposal outside prior support for Tmin = 1.63931838910736"
## [1] "proposing ds: prob.old = -450.3; prob.new = -494.08; A = 9.6616e-20"
## [1] "proposing fs: prob.old = -450.3; prob.new = -483.01; A = 6.1961e-15"
## [1] "proposing sr: prob.old = -450.3; prob.new = -460.05; A = 5.7901e-05"
## [1] "proposing eta: prob.old = -450.3; prob.new = -670.25; A = 2.9927e-96"
## [1] "proposal outside prior support for muz = -0.110690924283921"
## [1] "proposing muz: prob.old = -450.3; prob.new = -450.3; A = 0"
## [1] "proposal outside prior support for Tmin = 1.67409064792076"
## [1] "proposing Tmin: prob.old = -450.3; prob.new = -450.3; A = 0"
## [1] "sample number 150"
## [1] "proposal outside prior support for muz = -0.321116371275169"
## [1] "proposal outside prior support for muz = -0.406879081822555"
## [1] "proposal outside prior support for Tmin = 1.6494582561565"
## [1] "proposal outside prior support for Tmin = 1.97952355795685"
## [1] "proposal outside prior support for muz = -0.236604634970936"
## [1] "proposal outside prior support for muz = -0.764715394720414"
## [1] "proposal outside prior support for Tmin = 1.34297669449661"
## [1] "proposal outside prior support for Tmin = 1.8438864266491"
## [1] "proposal outside prior support for muz = -0.267765011621944"
## [1] "proposal outside prior support for Tmin = 1.83689330651173"
## [1] "proposal outside prior support for muz = -0.205358763159512"
## [1] "proposal outside prior support for sr = -0.000340106243836846"
## [1] "proposal outside prior support for Tmin = 1.95668851249464"
## [1] "proposal outside prior support for muz = -0.552537480881553"
## [1] "proposal outside prior support for Tmin = 1.90504629019482"
## [1] "proposing Tmin: prob.old = -450.04; prob.new = -450.06; A = 0.97694"
## [1] "proposal outside prior support for sr = -0.0233943289635132"
## [1] "proposing sr: prob.old = -450.06; prob.new = -450.06; A = 0"
## [1] "proposing fs: prob.old = -450.06; prob.new = -1661.6; A = 0"
## [1] "proposing eta: prob.old = -450.06; prob.new = -469.76; A = 2.7881e-09"
## [1] "proposing muz: prob.old = -450.06; prob.new = -1005.5; A = 5.6995e-242"
## [1] "proposing ds: prob.old = -450.06; prob.new = -460.38; A = 3.2839e-05"
## [1] "sample number 200"

```

We plot and summarize the MCMC chains

```
plot(dede_rev, ask=FALSE)
```





From the traceplot we can see that the burnin period is **about 200 samples**. We can remove the burnin and have a look at parameter correlations, and the overlap between the posterior and prior densities.

```
burnin = 1
pairs(dede_rev, burnin = burnin, scatter=TRUE, trend=TRUE)
```

```
## Warning in simpleLoess(y, x, w, span, degree, parametric, drop.square,
## normalize, : pseudoinverse used at 0.87124
```

```
## Warning in simpleLoess(y, x, w, span, degree, parametric, drop.square,
## normalize, : neighborhood radius 0.024766
```

```
## Warning in simpleLoess(y, x, w, span, degree, parametric, drop.square,
## normalize, : reciprocal condition number 0
```

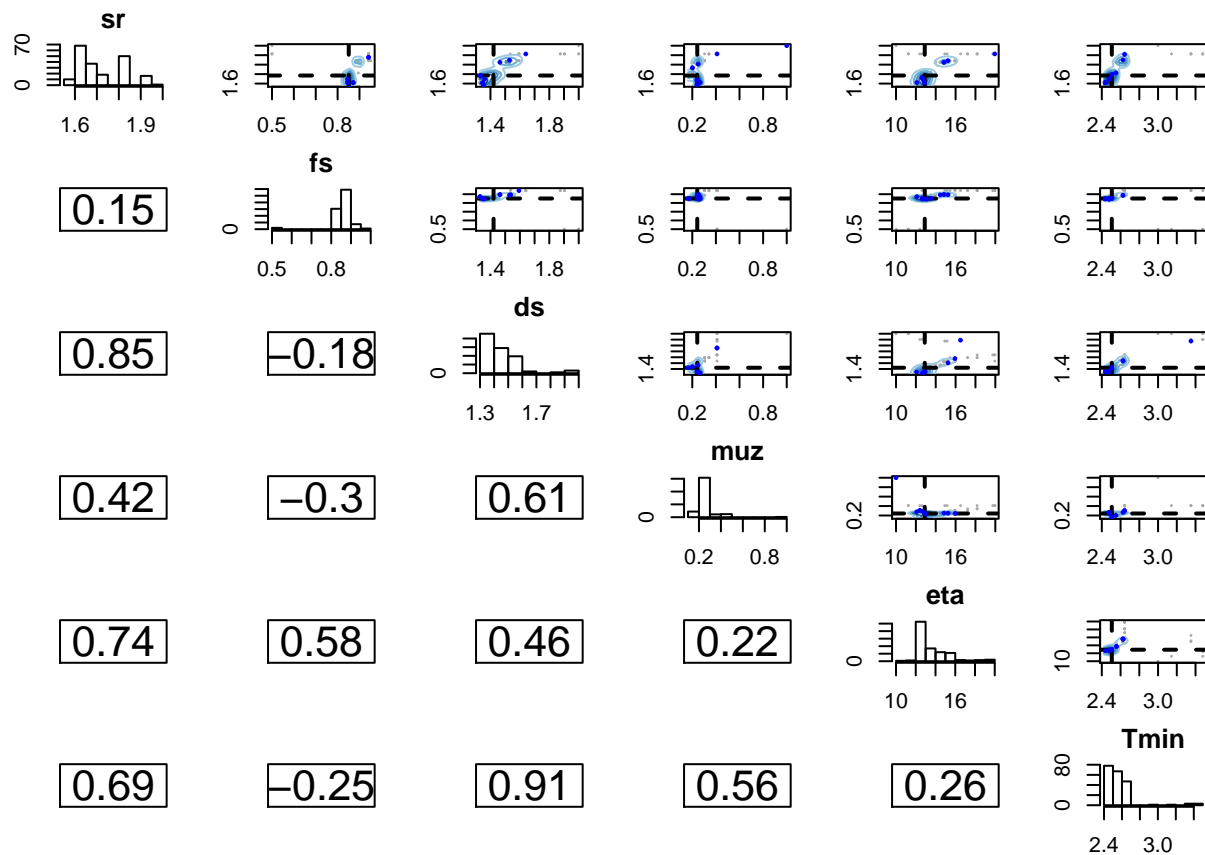


```
## Warning in simpleLoess(y, x, w, span, degree, parametric, drop.square,
## normalize, : pseudoinverse used at 2.6268
```

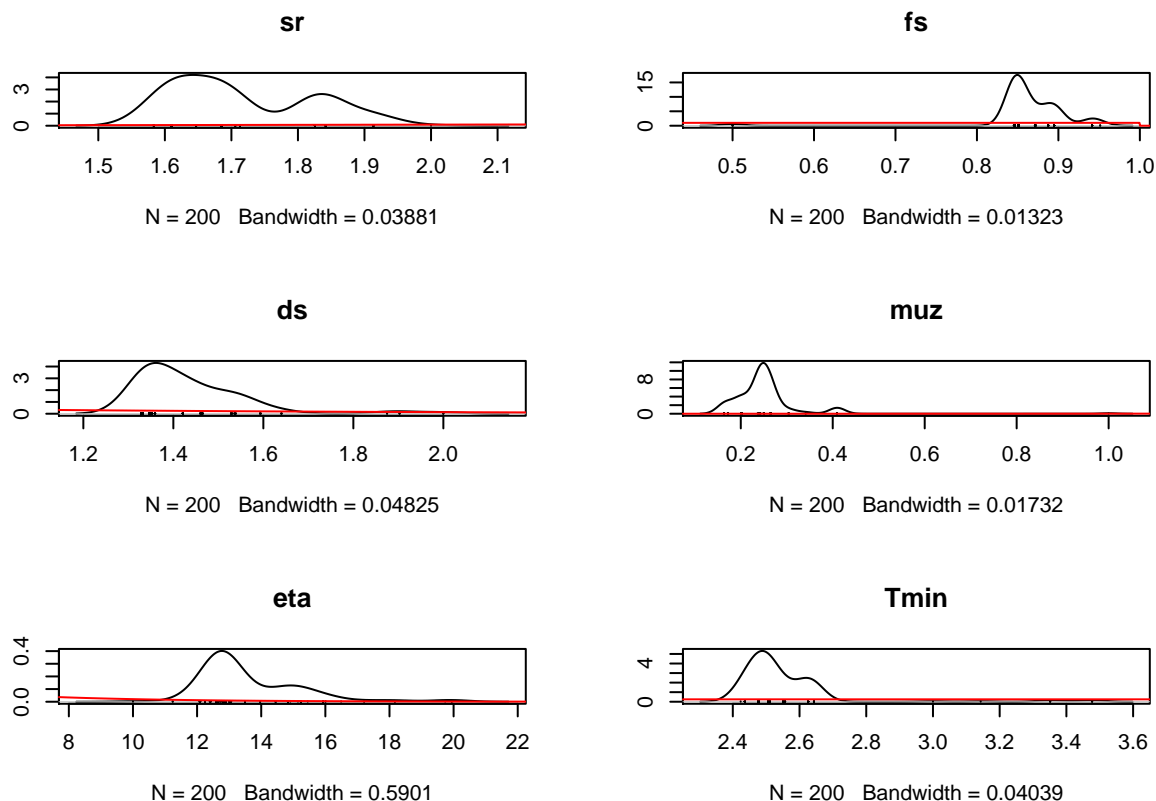
```
## Warning in simpleLoess(y, x, w, span, degree, parametric, drop.square,
## normalize, : neighborhood radius 0.14827
```

```
## Warning in simpleLoess(y, x, w, span, degree, parametric, drop.square,
## normalize, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree, parametric, drop.square,
## normalize, : There are other near singularities as well. 0.021705
```



```
post_prior_densplot(dede_rev, burnin = burnin)
```



```
summary(dede_rev)
```

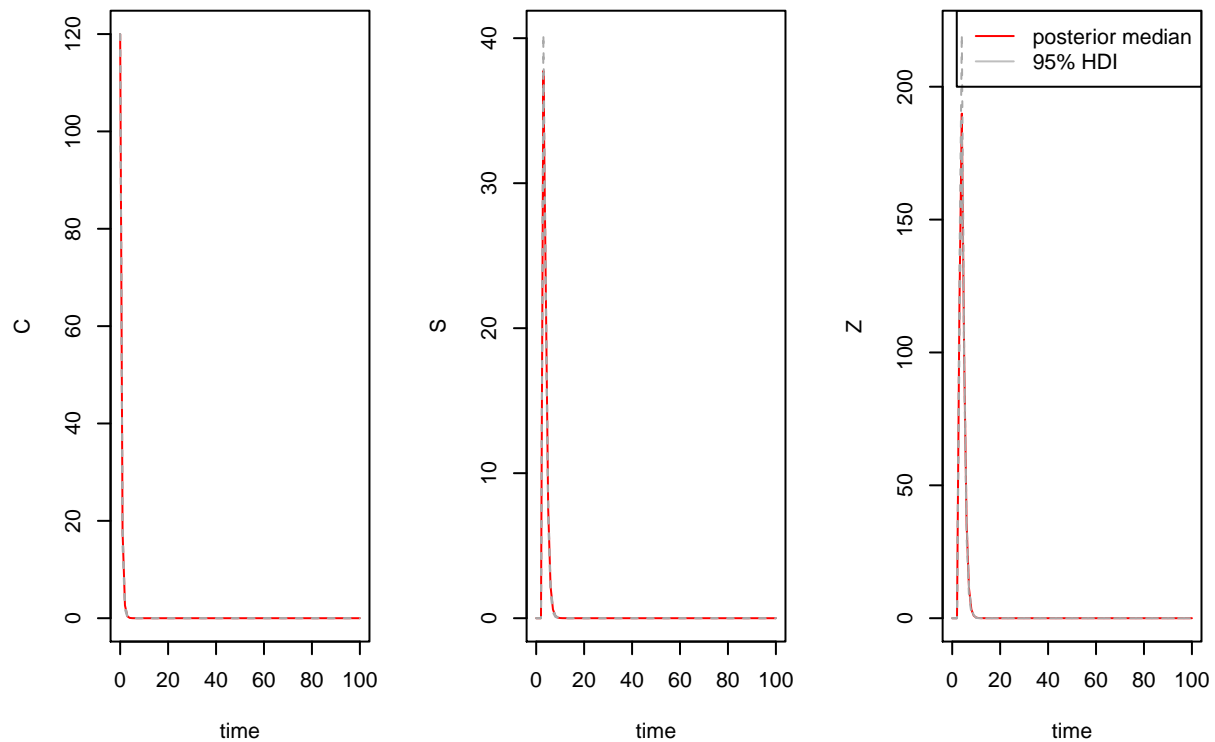
```
##
## Iterations = 1:200
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 200
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##      Mean      SD Naive SE Time-series SE
## sr    1.7174 0.10563 0.007469      0.07306
## fs    0.8621 0.05949 0.004206      0.01018
## ds    1.4389 0.13133 0.009286      0.06761
## muz   0.2509 0.07619 0.005388      0.02247
## eta  13.6135 1.60628 0.113581      0.74655
## Tmin  2.5551 0.17501 0.012375      0.04992
##
## 2. Quantiles for each variable:
##
##      2.5%    25%    50%    75%    97.5%
## sr    1.5832 1.6100 1.6850 1.8253 1.9135
## fs    0.8453 0.8465 0.8509 0.8947 0.9414
## ds    1.3288 1.3507 1.4211 1.5292 1.9026
## muz   0.1638 0.2017 0.2421 0.2649 0.4096
## eta  12.0943 12.6664 12.8939 14.8325 18.1680
## Tmin  2.4362 2.4785 2.5067 2.6258 3.3513
```

We simulate DE model trajectories from the posterior and calculate the HPD interval for the deterministic part of the model.

```
post_traj <- post_sim(dede_rev, n=20, times=0:100, burnin=burnin, output = 'all')
```

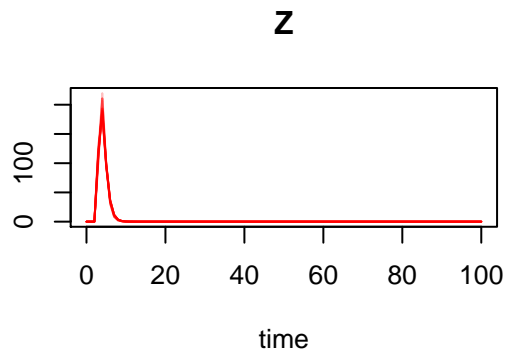
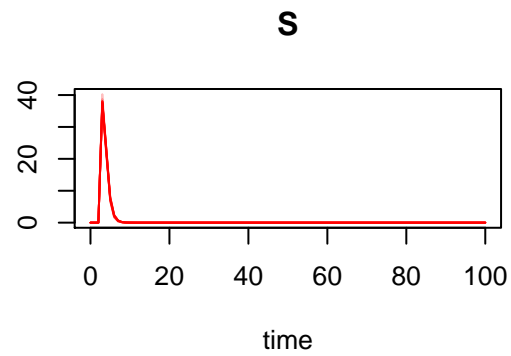
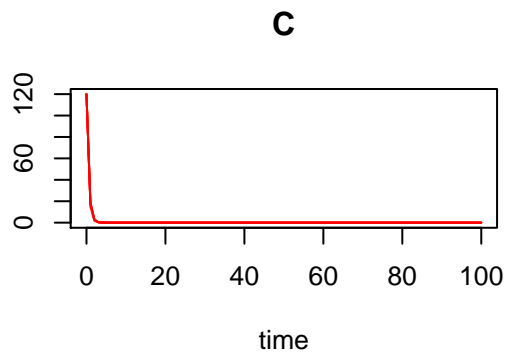
We can visualise the median posterior trajectory and the associated highest posterior density interval using

```
#median and HDI
par(mfrow=c(1,3))
plot(post_traj, plot.type = "medianHDI", auto.layout = FALSE)
legend("topright", legend=c("posterior median", "95% HDI"), lty=1, col=c("red","grey"))
```



Alternatively we can plot an ensemble of posterior trajectories

```
plot(post_traj, plot.type = "ensemble", col = "#FF000040")
```



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](https://doi.org/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.