Fitting multistate single-season occupancy models in Jags

O. Gimenez

2019-04-13 15:50:46

Motivation

We would like to implement a multistate single-season occupancy model (Nichols et al. 2007) in Jags. To do so, we adopt a hidden Markov modeling formulation of the model (Gimenez et al. 2014 and associated Wiki). To illustrate the analysis, we use simulated data provided by Donovan et al. (2007).

Data

The states are 1 for site unoccupied, 2 for occupied with no production of young ('uncertain' non-breeding state) and 3 for occupied with successful reproduction ('certain' breeding state). The observations are 0 for species not observed, 1 for species observed and 2 for species observed with young.

These data were simulated with the following parameter values: * probability that the site is occupied by non-breeders psi1 = 0.3; * probability that the site is occupied by breeders psi2 = 0.5; * detection probability of non-breeders p1 = 0.5; * detection probability of breeders p2 = 0.7; * probability of detecting evidence of reproduction, given the site is occupied with young delta = 0.8; * number of sites R = 250.

```
dat <- readr::read_csv2('https://raw.githubusercontent.com/oliviergimenez/multistate_occupancy/master/mu
#head(dat)
#tail(dat)
#sum(dat==1)
#sum(dat==2)
#sum(dat==0)
```

Model fitting in JAGS

Let's write the model:

```
model <- function() {

  # Define all parameters
  # Probabilities for initial states
  px0[1] <- 1 / (1 + prop[1] + prop[2])
  px0[2] <- prop[1] / (1 + prop[1] + prop[2]) # prob. of occupancy state 1
  px0[3] <- prop[2] / (1 + prop[1] + prop[2]) # prob. of occupancy state 2

  # Observation process
  # step 1: detection
  po1[1,1] <- 1
  po1[1,2] <- 0
  po1[1,3] <- 0
  po1[2,1] <- 1 - p1
  po1[2,2] <- p1 # detection state 1
  po1[2,3] <- 0
```

```
po1[3,1] <- 1 - p2
po1[3,2] <- 0
po1[3,3] <- p2 # detection state 2

# step 2: assignement
po2[1,1] <- 1
po2[1,2] <- 0
po2[1,3] <- 0
po2[2,1] <- 0
po2[2,2] <- 1
po2[2,3] <- 0
po2[3,1] <- 0
po2[3,2] <- 1 - delta
po2[3,3] <- delta # assignment conditional on detection
# form the matrix product
po <- po1 %*% po2

# State process
px[1,1] <- 1
px[1,2] <- 0
px[1,3] <- 0
px[2,1] <- 0
px[2,2] <- 1
px[2,3] <- 0
px[3,1] <- 0
px[3,2] <- 0
px[3,3] <- 1

for (i in 1:N){ # loop over site

  # state eq.
  z[i] ~ dcat(px0[1:3])

  # obs eq.
  for (j in 1:K){  # loop over occasion
    y[i,j] ~ dcat(po[z[i],1:3])
  }
}

# Prior
for (j in 1:2){ # use generalized logit for initial states
log(prop[j]) <- theta[j]
theta[j] ~ dnorm(0,1)
}
p1 ~ dunif(0, 1)
p2 ~ dunif(0, 1)
delta ~ dunif(0, 1)

psi1 <- prop[1] / (1 + prop[1] + prop[2]) # prob. of occupancy state 1
psi2 <- prop[2] / (1 + prop[1] + prop[2]) # prob. of occupancy state 2

}
```

Form the list of data:

```r
N <- nrow(dat)
K <- ncol(dat)
y <- as.matrix(dat + 1)
mydatax <- list(N = N, K = K, y = y)
```

Form the list of initial values:

```r
zinit <- apply(y,1,max)
init1 <- list(p1 = 0.3, theta = rnorm(2,0,1), z = zinit)
init2 <- list(p1 = 0.7, theta = rnorm(2,0,1), z = zinit)
inits <- list(init1, init2)
```

Specify the parameters to be monitored:

```r
parameters <- c("psi1","psi2","p1","p2","delta")
```

Tadaaaaaaan, fit the model:

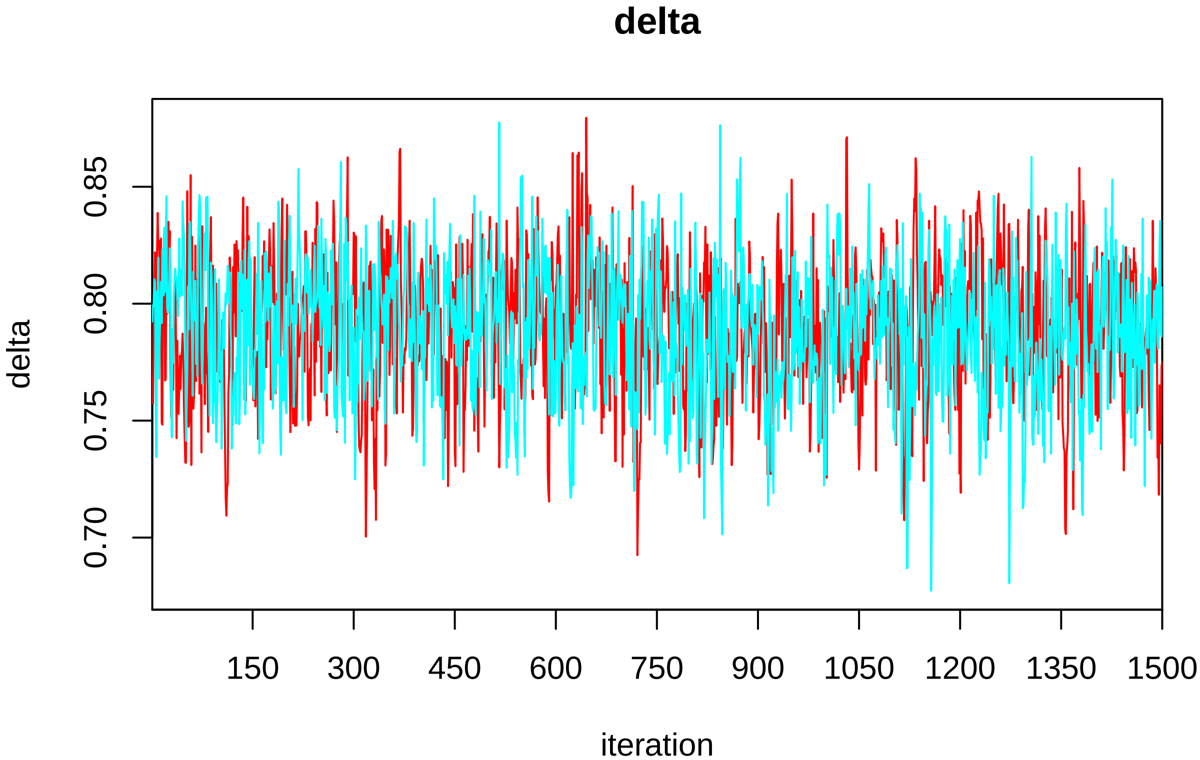
```r
library(R2jags)
out <- jags(mydatax, inits, parameters, model, n.chains=2, n.iter=2000, n.burnin=500)
```

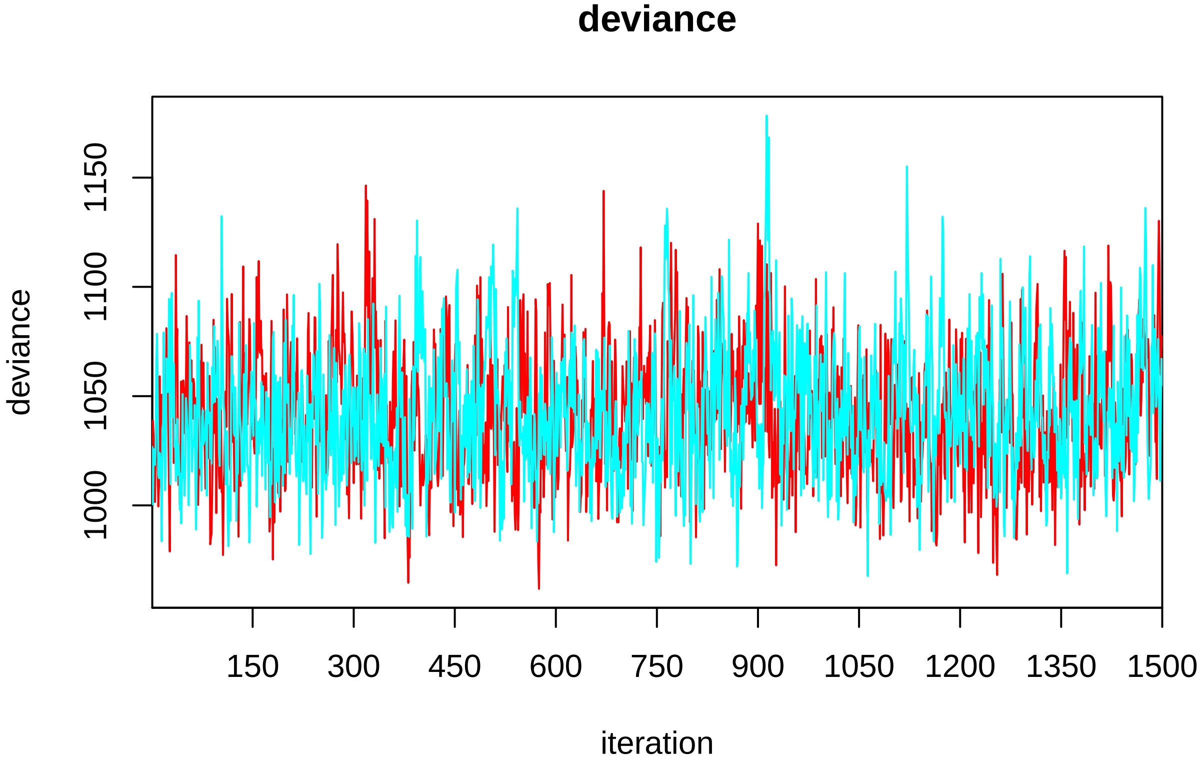
```
## Compiling model graph
##    Resolving undeclared variables
##    Allocating nodes
## Graph information:
##    Observed stochastic nodes: 747
##    Unobserved stochastic nodes: 254
##    Total graph size: 1270
##
## Initializing model
```

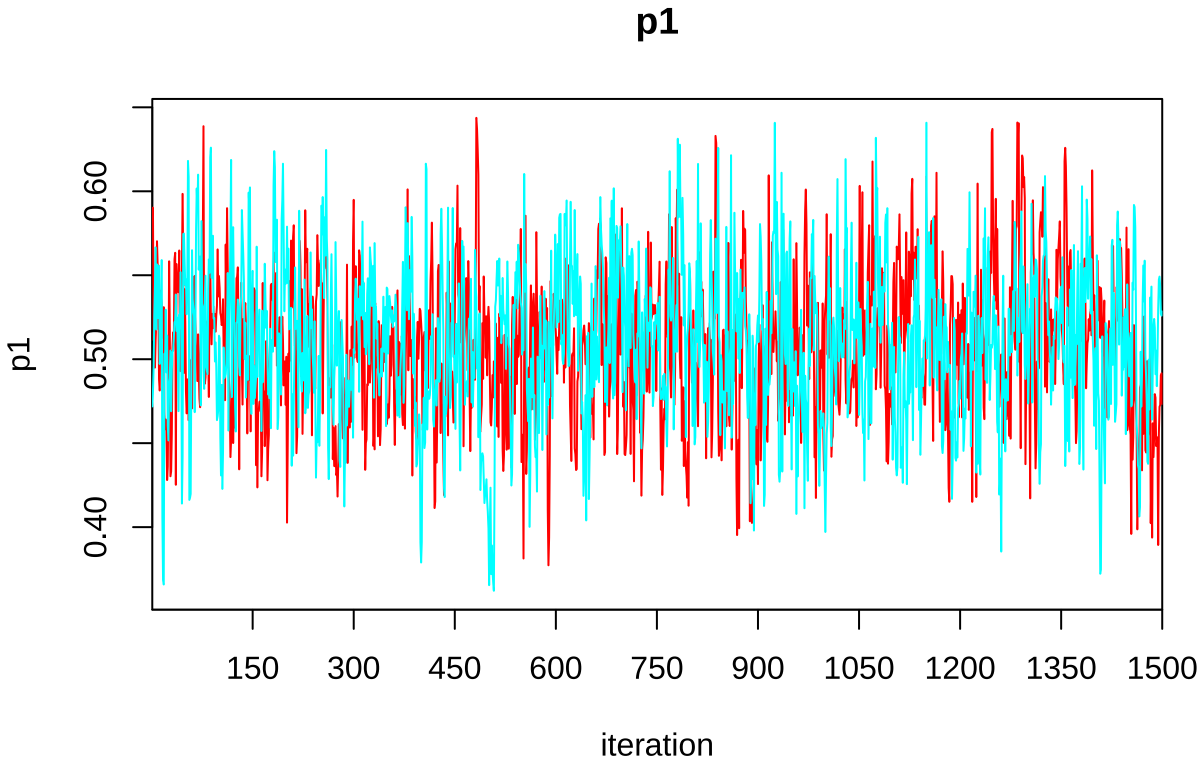
Results

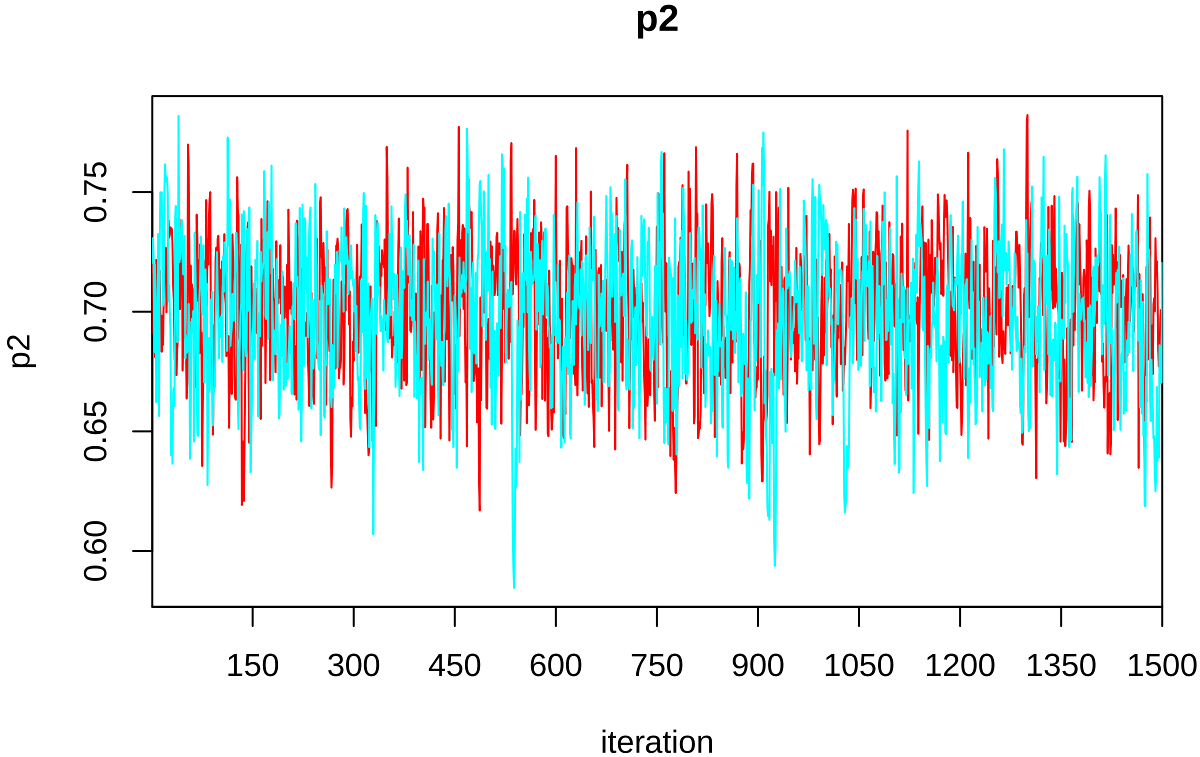
Check convergence:

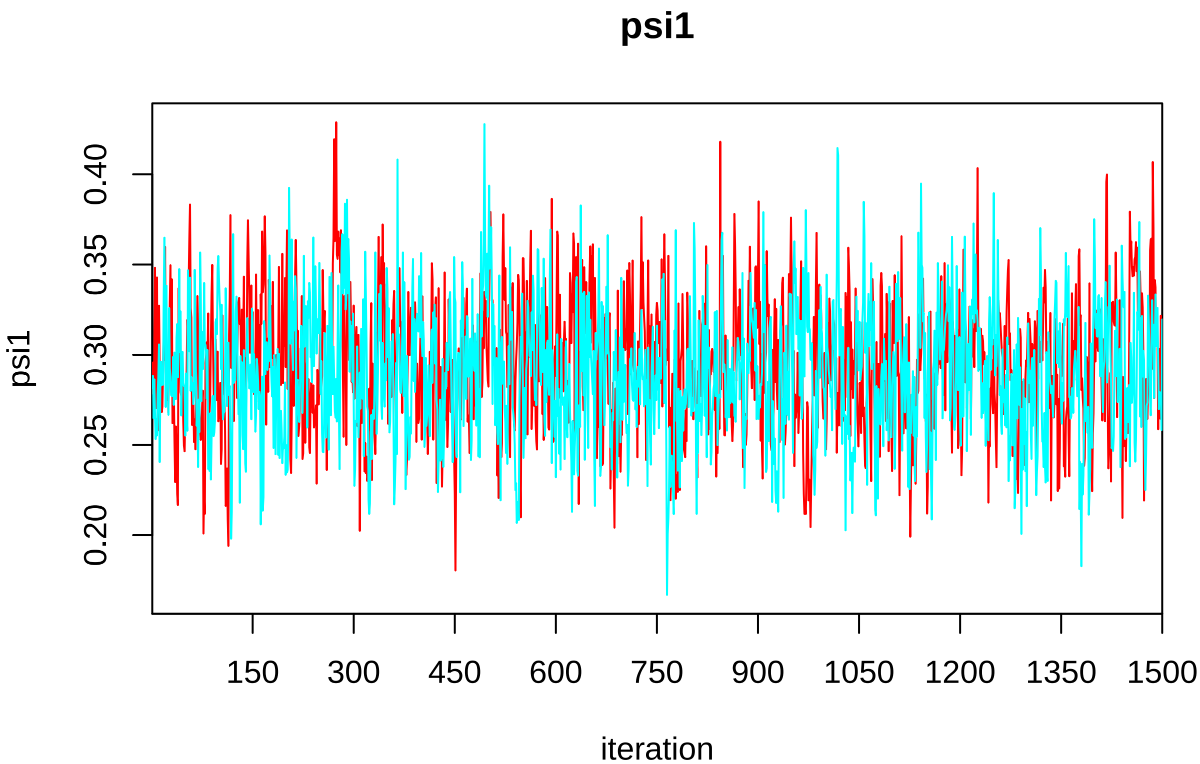
```r
traceplot(out,ask=F)
```

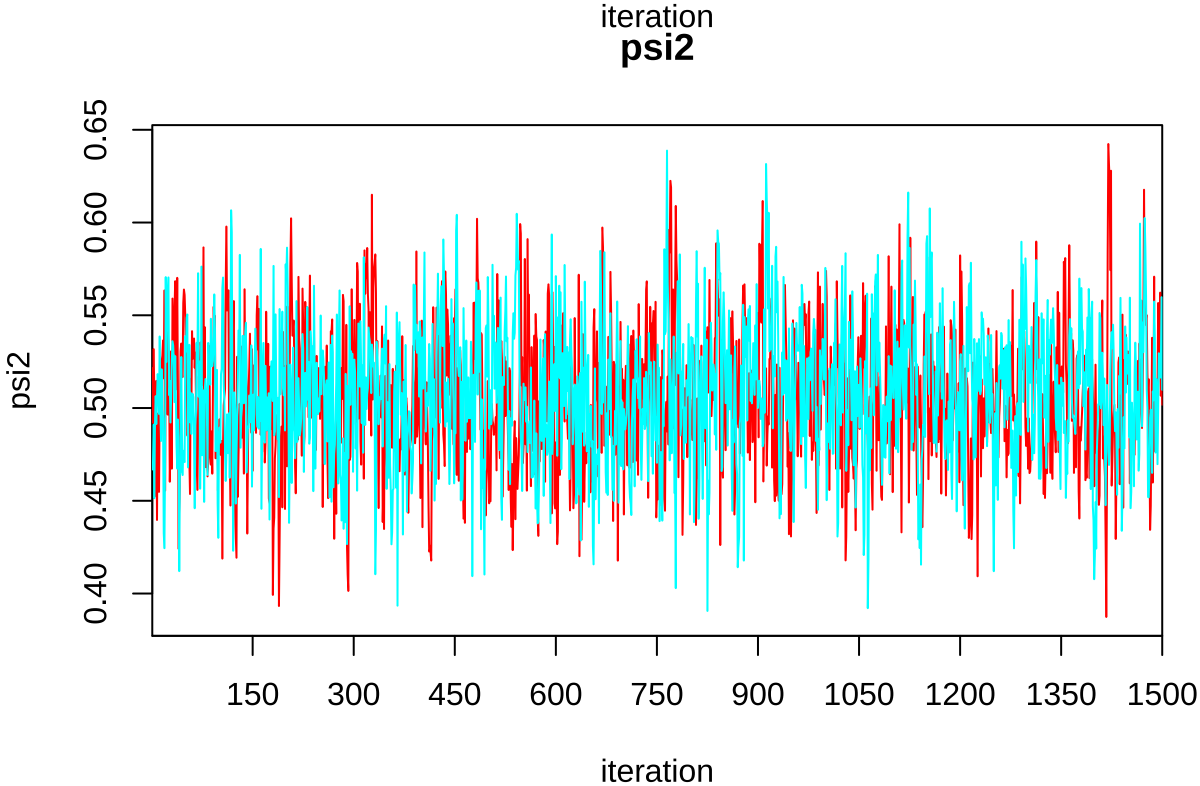

delta
delta
0.70
0.75
0.80
0.85
150
300
450
600
750
900
1050
1200
1350
1500
iteration


deviance
deviance
1000
1050
1100
1150
150
300
450
600
750
900
1050
1200
1350
1500
iteration


p1
p1
0.40
0.50
0.60
150
300
450
600
750
900
1050
1200
1350
1500
iteration

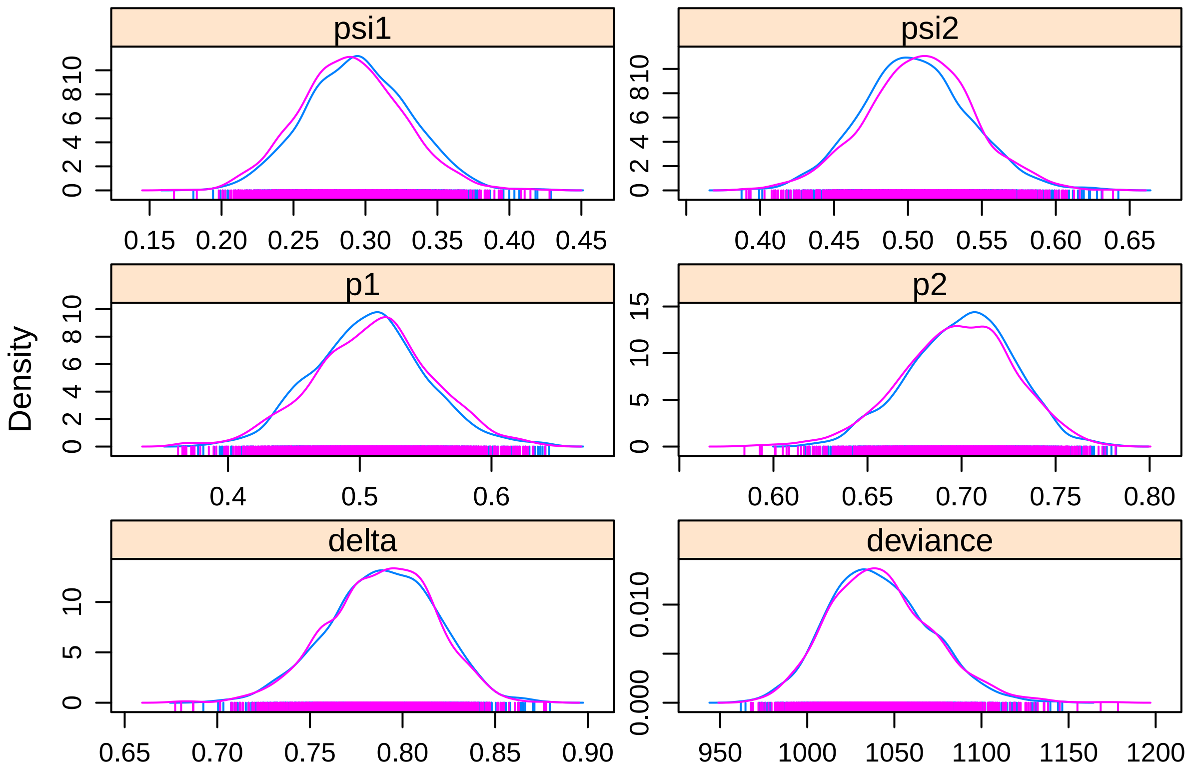

p2
p2
0.60
0.65
0.70
0.75
150
300
450
600
750
900
1050
1200
1350
1500
iteration





Posterior densities:

```
library(lattice)
jagsfit.mcmc <- as.mcmc(out)
densityplot(jagsfit.mcmc)
```



Print results:

```r
print(out,digits = 2)
```

```
## Inference for Bugs model at "/var/folders/c5/2p8dsf5n1_z4l7kb7y008dcm0000gp/T//RtmpRgqd0P/model4f264
##  2 chains, each with 2000 iterations (first 500 discarded)
##  n.sims = 3000 iterations saved
##          mu.vect sd.vect   2.5%     25%     50%     75%   97.5% Rhat n.eff
## delta       0.79    0.03   0.73    0.77    0.79    0.81    0.84 1.00  3000
## p1          0.51    0.04   0.42    0.48    0.51    0.54    0.60 1.00  1400
## p2          0.70    0.03   0.64    0.68    0.70    0.72    0.75 1.01   530
## psi1        0.29    0.04   0.22    0.27    0.29    0.32    0.36 1.01   210
## psi2        0.51    0.04   0.43    0.48    0.51    0.53    0.58 1.00   540
## deviance 1041.99   29.33 990.43 1020.86 1039.78 1060.54 1104.73 1.00  2100
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
##
## DIC info (using the rule, pD = var(deviance)/2)
## pD = 430.0 and DIC = 1472.0
## DIC is an estimate of expected predictive error (lower deviance is better).
```

And compare with E-SURGE results:

```r
#Par# 54# C( 3, 3)( 1, 1)( 1 2) | 0.794364460 delta 0.731225549 0.845798972 0.029218536
#Par# 24# E( 2, 2)( 1, 1)( 1 1) | 0.515786096 p1 0.425952230 0.604611235 0.046070203
#Par# 25# E( 3, 3)( 1, 1)( 1 1) | 0.704231715 p2 0.646969850 0.755712647 0.027819907
#Par# 2# IS( 1, 2)( 1, 1)( 1 1) | 0.295674999 psi1 0.228163996 0.373495359 0.037265126
#Par# 3# IS( 1, 3)( 1, 1)( 1 1) | 0.502904937 psi2 0.432898323 0.572797837 0.035924248
```

References

Donovan, T. M. and J. Hines (2007) Exercises in occupancy modeling and estimation - Exercise 16 'Multiple occupancy states models'. http://www.uvm.edu/rsenr/vtcfwru/spreadsheets/?Page=occupancy/occupancy.htm

Gimenez, O., L. Blanc, A. Besnard, R. Pradel, P. F. Doherty Jr, E. Marboutin and R. Choquet (2014). Fitting occupancy models with E-SURGE: hidden-Markov modelling of presence-absence data. Methods in Ecology and Evolution. 5: 592–597.

Nichols, J. D., Hines, J. E., Mackenzie, D. I., Seamans, M. E. and Gutiérrez, R. J. (2007). Occupancy estimation and modeling with multiple states and state uncertainty. Ecology 88: 1395-1400.