Ecography

E6433
Rota, C. T., Fletcher, R. J. Jr, Evans, J. M. and Hutto, R. L. 2010. Does accounting for imperfect detection improve species distribution models? – Ecography 33: xxx–xxx.

Supplementary material

Example WinBugs code for a hierarchical occupancy model that uses a removal design for estimating detection probability

```
model{
#Prior distributions for fixed effects
  bMean ~ dunif(0,1)
  b0 <- log(bMean) - log(1-bMean)
  b1 ~ dnorm(0,.001)
  b2 ~ dnorm(0,.001)
  aMean ~ dunif(0,1)
  a0 <- log(aMean) - log(1-aMean)
  a1 ~ dnorm(0,.001)
  a2 ~ dnorm(0,.001)
  a3 ~ dnorm(0,.001)
  a4 ~ dnorm(0,.001)
  a5 ~ dnorm(0,.001)
  a6 ~ dnorm(0,.001)
  a7 ~ dnorm(0,.001)
  a8 ~ dnorm(0,.001)
  a9 ~ dnorm(0,.001)
  w1 ~ dbern(.5)
  w2 ~ dbern(.5)
  w3 ~ dbern(.5)
  w4 ~ dbern(.5)
  w5 ~ dbern(.5)
  w6 ~ dbern(.5)
  w7 ~ dbern(.5)
  w8 ~ dbern(.5)
  w9 ~ dbern(.5)
  w10 ~ dbern(.5)
  w11 ~ dbern(.5)

#Prior distributions for random effect variance parameters
  sd.transect ~ dunif(0,10)
  tau.transect <- pow(sd.transect,-2)
  sd.year ~ dunif(0,10)
  tau.year <- pow(sd.year,-2)
  sd.observer ~ dunif(0,10)
  tau.observer <- pow(sd.observer,-2)
```

```
#Prior distributions for random effects
  for(o in 1:nObserver){
    observer.effect[o] ~ dnorm(a0, tau.observer)}
  for(r in 1:nTransect){
    transect.effect[r] ~ dnorm(b0, tau.transect)}
  for(t in 1:nYear){
    year.effect[t] ~ dnorm(b0,tau.year)}

  for(i in 1:n){
    #eqn 2
    logit(psi[i]) <- w1 * b1 * Precip[i] + w2 * b2 * Mesic[i] +
      transect.effect[Transect[i]] + year.effect[Year[i]]
    #eqn 6
    logit(p[i]) <- w3 * a1 * Date[i] + w3 * w4 * a2 * Date_Sq[i] + w5 * a3 *
      Time[i]+ w5 * w6 * a4 * Time_Sq[i]+ w7 * a5 * Wind[i] + w8 * a6 *
      Sky[i] + w9 * a7 * Stream[i] + w10 * a8 * DBH[i] + w11 * a9 * CC[i] +
      observer.effect[Observer[i]]

    #Process Model, eqn 1
    z[i] ~ dbern(psi[i])

    #Observation model, eqn 3
    mu.y[i] <- 1 - pow(1-p[i]*z[i], J)
    y[i] ~ dbern(mu.y[i])

    #eqn 4 & 5
    for (k in 1:J) {
      pi[i,k] <- y[i]*prob.y1[i,k] + (1-y[i])*prob.y0[i,k]}
      j[i] ~ dcat(pi[i,])}

#eqn 4 & 5
for(i in 1:n){
    for (k in 1:J) {
      prob.y1[i,k] <- (p[i] * pow(1-p[i], k-1)) / (1 - pow(1-p[i], J))
      prob.y0[i,k] <- step(k-J)}}}
```