APPENDIX 2

Two Further Useful Multistate Capture–Recapture Models

This appendix contains code for data simulation and analysis of two further multistate capture–recapture models.

2.1 ESTIMATION OF AGE-SPECIFIC SURVIVAL PROBABILITIES

2.1.1 Model Description

Age is one of the main predictors of survival in animal populations, and hence, estimation of age-specific survival is important in population analysis. Multistate models can be used to estimate age-specific survival; we define states as age classes. As a simple example, we consider the estimation of juvenile and adult survival. Juvenile survival refers to the interval from birth until the age of 1, and adult survival refers to all annual intervals after age 1. The states in this model are therefore "juvenile", "adult", and "dead". Here is the state-transition matrix:

$$\begin{array}{cc} & \begin{array}{ccc}\text{juvenile} & \text{adult} & \text{dead}\end{array} \\ \begin{array}{c}\text{juvenile}\\ \text{adult}\\ \text{dead}\end{array} & \begin{bmatrix} 0 & \phi_{\text{juv}} & 1-\phi_{\text{juv}} \\ 0 & \phi_{\text{ad}} & 1-\phi_{\text{ad}} \\ 0 & 0 & 1 \end{bmatrix}\end{array}$$

where ϕ_{juv} is juvenile, and ϕ_{ad} is adult apparent survival probability. As always, states are ordered from top down and from left to right in the same order as above. Juvenile survival is not on the diagonal of the transition matrix because juveniles become adult if they survive their first year of life.

The vector of observed states includes "seen as juvenile", "seen as adult", and "not seen"; hence, the observation matrix is

$$\begin{array}{cc} & \begin{array}{ccc}\text{seen as juvenile} & \text{seen as adult} & \text{not seen}\end{array} \\ \begin{array}{c}\text{juvenile}\\ \text{adult}\\ \text{dead}\end{array} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & p & 1-p \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

where p is recapture probability. The true states at time t correspond to the rows of the observation matrix in the order "alive as juvenile", "alive as adult", and "dead" from top down, and the observed states are in columns in the order "seen as juvenile", "seen as adult", and "not seen" from left to right. It may seem strange at first that juveniles cannot be seen—but if you think about it, the reason for it becomes clear juveniles may well be captured and marked, but the only possibility for them to be recaptured is in the following occasion at 1-year old and thus in the adult state.

2.1.2 Generation of Simulated Data

Let us assume that we marked young and adult little owls. Survival probability is 0.65 for adults and 0.3 for juveniles (from fledgling to their first anniversary), and recapture probability is 0.5 (note recapture refers only to adults). First, we simulate a data set, where juveniles are denoted as 1 and adults as 2.

```
# Define mean survival, transitions, recapture, as well as number of
  occasions, states, observations and released individuals
phi.juv <- 0.3
phi.ad <- 0.65
p <- 0.5
n.occasions <- 6
n.states <- 3
n.obs <- 3
marked <- matrix(NA, ncol = n.states, nrow = n.occasions)
marked[,1] <- rep(200, n.occasions)    # Juveniles
marked[,2] <- rep(30, n.occasions)     # Adults
marked[,3] <- rep(0, n.occasions)      # Dead individuals

# Define matrices with survival, transition and recapture
  probabilities
# These are 4-dimensional matrices, with
    # Dimension 1: state of departure
    # Dimension 2: state of arrival
    # Dimension 3: individual
    # Dimension 4: time
# 1. State process matrix
totrel <- sum(marked)*(n.occasions-1)
PSI.STATE <- array(NA, dim=c(n.states, n.states, totrel,
   n.occasions-1))
```

```
for (i in 1:totrel){
    for (t in 1:(n.occasions-1)){
        PSI.STATE[,,i,t] <- matrix(c(
        0, phi.juv, 1-phi.juv,
        0, phi.ad,  1-phi.ad,
        0, 0,       1          ), nrow = n.states, byrow = TRUE)
        } #t
    } #i
# 2.Observation process matrix
PSI.OBS <- array(NA, dim=c(n.states, n.obs, totrel, n.occasions-1))
for (i in 1:totrel){
    for (t in 1:(n.occasions-1)){
        PSI.OBS[,,i,t] <- matrix(c(
        0, 0, 1,
        0, p, 1-p,
        0, 0, 1      ), nrow = n.states, byrow = TRUE)
        } #t
    } #i

# Execute simulation function
sim <- simul.ms(PSI.STATE, PSI.OBS, marked)
CH <- sim$CH

# Compute vector with occasion of first capture
get.first <- function(x) min(which(x!=0))
f <- apply(CH, 1, get.first)

# Recode CH matrix: note, a 0 is not allowed in WinBUGS!
# 1 = seen alive as juvenile, 2 = seen alive as adult, 3 = not seen
rCH <- CH # Recoded CH
rCH[rCH==0] <- 3
```

We create only one data set, although we mark juveniles and adults. The capture-histories of individuals marked as juveniles start with a "1" and those of adults with a "2". If we analyzed these data with a single-state capture–recapture (CJS) model, we would have to create an individual covariate containing the age (class) at first capture (see Section 7.7).

2.1.3 Analysis of the Model

```
# Specify model in BUGS language
sink("age.bug")
cat("
model {
# -----------------------------------------
# Parameters:
# phi.juv: juvenile survival probability
# phi.ad: adult survival probability
# p: recapture probability
# -----------------------------------------
# States (S):
# 1 alive as juvenile
# 2 alive as adult
# 3 dead
# Observations (O):
# 1 seen as juvenile
# 2 seen as adult
# 3 not seen
# -------------------------------------

# Priors and constraints
for (t in 1:(n.occasions-1)){
   phi.juv[t] <- mean.phijuv
   phi.ad[t] <- mean.phiad
   p[t] <- mean.p
   }
mean.phijuv ~ dunif(0, 1)
mean.phiad ~ dunif(0, 1)
mean.p ~ dunif(0, 1)

# Define state-transition and observation matrices
for (i in 1:nind){
   # Define probabilities of state S(t+1) given S(t)
   for (t in f[i]:(n.occasions-1)){
      ps[1,i,t,1] <- 0
      ps[1,i,t,2] <- phi.juv[t]
      ps[1,i,t,3] <- 1-phi.juv[t]
      ps[2,i,t,1] <- 0
      ps[2,i,t,2] <- phi.ad[t]
      ps[2,i,t,3] <- 1-phi.ad[t]
      ps[3,i,t,1] <- 0
      ps[3,i,t,2] <- 0
      ps[3,i,t,3] <- 1

      # Define probabilities of O(t) given S(t)
      po[1,i,t,1] <- 0
      po[1,i,t,2] <- 0
      po[1,i,t,3] <- 1
      po[2,i,t,1] <- 0
      po[2,i,t,2] <- p[t]
      po[2,i,t,3] <- 1-p[t]
      po[3,i,t,1] <- 0
      po[3,i,t,2] <- 0
      po[3,i,t,3] <- 1
      } #t
   } #i

# State-space model likelihood
for (i in 1:nind){
   z[i,f[i]] <- Y[i,f[i]]
   for (t in (f[i]+1):n.occasions){
      # State equation: draw S(t) given S(t-1)
      z[i,t] ~ dcat(ps[z[i,t-1], i, t-1,])
      # Observation equation: draw O(t) given S(t)
      Y[i,t] ~ dcat(po[z[i,t], i, t-1,])
      } #t
   } #i
}
```

```
",fill = TRUE)
sink()

# Bundle data
bugs.data <- list(Y = rCH, f = f, n.occasions = dim(rCH)[2], nind =
   dim(rCH)[1])

# Initial values
inits <- function(){list(mean.phijuv = runif(1, 0, 1), mean.phiad =
   runif(1, 0, 1), mean.p = runif(1, 0, 1), z = ch.init(rCH, f))}

# Parameters monitored
parameters <- c("mean.phijuv", "mean.phiad", "mean.p")

# MCMC settings
ni <- 2000
nt <- 3
nb <- 1000
nc <- 3

# Call WinBUGS from R (BRT 2 min)
age <- bugs(bugs.data, inits, parameters, "age.bug", n.chains = nc,
   n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
   bugs.directory = bugs.dir, working.directory = getwd())
```

The chains converge quite quickly.

```
print(age, digits = 3)

             mean    sd  2.5%   25%   50%   75% 97.5%  Rhat n.eff
mean.phijuv 0.287 0.022 0.244 0.272 0.287 0.301 0.331 1.000  1000
mean.phiad  0.634 0.026 0.584 0.615 0.634 0.651 0.684 1.002  1000
mean.p      0.517 0.031 0.456 0.497 0.516 0.538 0.577 1.000  1000
```

The same model structure (age-dependent survival with two age classes) can be used to account for transients in the analyzed population and to estimate the probability that a newly caught individual is a transient (Pradel et al., 1997; Schaub et al., 2004b).

2.2 ACCOUNTING FOR IMMEDIATE TRAP RESPONSE

2.2.1 Model Description

One assumption of standard capture–recapture models in Chapters 7 and 9 is that all marked animals alive and available for capture at a given occasion have the same recapture probability. Sometimes this assumption is violated in a very specific way, namely that individuals captured at time $t-1$ have a different recapture probability at time t than individuals not captured at time $t-1$. This is called trap response or behavioral response (see Sections 6.2.3 and 7.8). If recapture probability at time t for individuals captured at $t-1$ is higher, trap response is called "trap happiness", if recapture probability is lower, then it is called "trap shyness". There are several mechanisms that may induce one or the other effect. Sometimes the effects are also induced by the sampling and do not reflect a behavioral change of the individuals. Nevertheless, trap response must be modeled; otherwise estimates for the other parameters will be biased. To account for immediate trap response, a multistate model can be used (Schaub et al., 2009). For a solution for the CJS model, see Section 7.8 and Pradel (1993b).

The states in this model are "alive and seen", "alive and not seen" and "dead". By this definition of the states, the observation process is included in the state-transition process. This is necessary, because the trap-response is a Markovian process which can be implemented in the state-transition matrix, but not in the observation matrix. The state-transition matrix is then

$$\begin{array}{c} \\ \text{alive, seen} \\ \text{alive, not seen} \\ \text{dead} \end{array} \begin{array}{c} \begin{array}{ccc} \text{alive, seen} & \text{alive, not seen} & \text{dead} \end{array} \\ \begin{bmatrix} \phi p_{\text{ss}} & \phi(1-p_{\text{ss}}) & 1-\phi \\ \phi p_{\text{ns}} & \phi(1-p_{\text{ns}}) & 1-\phi \\ 0 & 0 & 1 \end{bmatrix} \end{array}.$$

Here, ϕ is survival probability, p_{ss} is recapture probability if captured at the previous occasion ("seen", "seen"), and p_{ns} is recapture probability if not captured at the previous occasion ("not seen", "seen").

The possible observations are "seen" and "not seen". Because "seen" and "not seen" are defined and modeled the previous matrix, which formally describes the state process, assignment in the observation process is now deterministic. The observation transition matrix becomes

$$\begin{array}{c} \\ \text{alive, seen} \\ \text{alive, not seen} \\ \text{dead} \end{array} \begin{array}{c} \begin{array}{cc} \text{seen} & \text{not seen} \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array}.$$

This model has identifiability problems. For example, when both recapture probabilities vary independently from each other over time, the model (often termed p_{t*m}) is parameter-redundant (Gimenez et al., 2003). By contrast, models where both recapture probabilities are either linked with an additive time structure (p_{t+m}) or are constant over time (p_m) are not parameter redundant.

2.2.2 Generation of Simulated Data

We now generate the data as used in Section 7.8. Recall that we estimate survival of adult red-backed shrikes. We catch adults during the breeding season and mark them with color rings to facilitate resighting and identification in the subsequent years. We survey all potential breeding territories each year. Typically we focus on breeding territories that were occupied by shrikes in previous years. If time allows, we search for other, newly established territories. Thus, marked individuals that survived and are philopatric to their territory have a higher chance of being resighted, while individuals that establish new territories are less likely to be found. However, once they are found, their chances of being resighted in the next year increases again. Such a sampling protocol induces a trap effect that biases survival unless accounted for. For data simulation, we assume survival $\phi = 0.55$ and resighting probabilities $p_{ss} = 0.75$ following a sighting in the preceding year and $p_{ns} = 0.35$ if there was no sighting in the preceding occasion.

```
# Define mean survival, transitions, recapture, as well as number of
  occasions, states, observations and released individuals
phi <- 0.55
pss <- 0.75
pns <- 0.3
n.occasions <- 10
n.states <- 3
n.obs <- 2
marked <- matrix(NA, ncol = n.states, nrow = n.occasions)
marked[,1] <- rep(100, n.occasions)    # Alive, seen
marked[,2] <- rep(0, n.occasions)      # Alive, not seen
marked[,3] <- rep(0, n.occasions)      # Dead

# Define matrices with survival, transition and recapture
  probabilities
# These are 4-dimensional matrices, with
    # Dimension 1: state of departure
    # Dimension 2: state of arrival
    # Dimension 3: individual
    # Dimension 4: time
# 1. State process matrix
totrel <- sum(marked)*(n.occasions-1)
PSI.STATE <- array(NA, dim=c(n.states, n.states, totrel,
   n.occasions-1))
for (i in 1:totrel){
    for (t in 1:(n.occasions-1)){
        PSI.STATE[,,i,t] <- matrix(c(
        phi*pss, phi*(1-pss), 1-phi,
        phi*pns, phi*(1-pns), 1-phi,
        0,       0,           1       ), nrow = n.states, byrow =
           TRUE)
        } #t
    } #i

# 2.Observation process matrix
PSI.OBS <- array(NA, dim=c(n.states, n.obs, totrel, n.occasions-1))
for (i in 1:totrel){
    for (t in 1:(n.occasions-1)){
        PSI.OBS[,,i,t] <- matrix(c(
         1, 0,
         0, 1,
         0, 1 ), nrow = n.states, byrow = TRUE)
         } #t
     } #i

# Execute simulation function
sim <- simul.ms(PSI.STATE, PSI.OBS, marked, unobservable = 2)
CH <- sim$CH

# Compute vector with occasion of first capture
get.first <- function(x) min(which(x!=0))
f <- apply(CH, 1, get.first)

# Recode CH matrix: note, a 0 is not allowed in WinBUGS!
# 1 = seen, 2 = not seen
rCH <- CH # Recoded CH
rCH[rCH==0] <- 2
```

The capture–recapture data now consist only of 1 at occasions when an individual was seen and of a 2 at occasions when an individual was not seen.

2.2.3 Analysis of the Model

Again, to fit the trap response model in the BUGS code, we only alter the definition of matrices, priors, and possibly of the constraints.

```
# Specify model in BUGS language
sink("immtrap.bug")
cat("
model {
# -------------------------------------------------
# Parameters:
# phi: survival probability
# pss: recapture probability at t, given captured at t-1
# pns: recapture probability at t, given not captured at t-1
# -------------------------------------------------
# States (S):
# 1 alive, seen at t-1
# 2 alive, not seen at t-1
# 3 dead
# Observations (O):
# 1 seen
# 2 not seen
# -------------------------------------------------

# Priors and constraints
for (t in 1:(n.occasions-1)){
   phi[t] <- mean.phi
   pss[t] <- mean.pss
   pns[t] <- mean.pns
   }
mean.phi ~ dunif(0, 1)
mean.pss ~ dunif(0, 1)
mean.pns ~ dunif(0, 1)

# Define state-transition and observation matrices
for (i in 1:nind){
    # Define probabilities of state S(t+1) given S(t)
    for (t in f[i]:(n.occasions-1)){
        ps[1,i,t,1] <- phi[t]*pss[t]
        ps[1,i,t,2] <- phi[t]*(1-pss[t])
        ps[1,i,t,3] <- 1-phi[t]
        ps[2,i,t,1] <- phi[t]*pns[t]
        ps[2,i,t,2] <- phi[t]*(1-pns[t])
        ps[2,i,t,3] <- 1-phi[t]
        ps[3,i,t,1] <- 0
        ps[3,i,t,2] <- 0
        ps[3,i,t,3] <- 1

        # Define probabilities of O(t) given S(t)
        po[1,i,t,1] <- 1
        po[1,i,t,2] <- 0
        po[2,i,t,1] <- 0
        po[2,i,t,2] <- 1
        po[3,i,t,1] <- 0
        po[3,i,t,2] <- 1
        } #t
    } #i
# State-space model likelihood
for (i in 1:nind){
    z[i,f[i]] <- Y[i,f[i]]
    for (t in (f[i]+1):n.occasions){
        # State equation: draw S(t) given S(t-1)
        z[i,t] ~ dcat(ps[z[i,t-1], i, t-1,])
        # Observation equation: draw O(t) given S(t)
        Y[i,t] ~ dcat(po[z[i,t], i, t-1,])
        } #t
    } #i
}
",fill = TRUE)
sink()

# Bundle data
bugs.data <- list(Y = rCH, f = f, n.occasions = dim(rCH)[2], nind =
   dim(rCH)[1], z = known.state.ms(rCH, 2))

# Initial values
inits <- function(){list(mean.phi = runif(1, 0, 1), mean.pss =
   runif(1, 0, 1), mean.pns = runif(1, 0, 1), z = ms.init.z(rCH, f))}

# Parameters monitored
parameters <- c("mean.phi", "mean.pss", "mean.pns")

# MCMC settings
ni <- 20000
nt <- 6
nb <- 10000
nc <- 3
```

```
# Call WinBUGS from R (BRT 33 min)
immtrap <- bugs(bugs.data, inits, parameters, "immtrap.bug",
   n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE,
   bugs.directory = bugs.dir, working.directory = getwd())

print(immtrap, digits = 3)

             mean    sd  2.5%   25%   50%   75% 97.5%  Rhat n.eff
mean.phi    0.558 0.028 0.513 0.538 0.555 0.575 0.621 1.006  1000
mean.pss    0.776 0.040 0.692 0.750 0.779 0.805 0.847 1.006   660
mean.pns    0.312 0.097 0.145 0.239 0.307 0.376 0.517 1.006   790
```

Convergence is not obtained very quickly, so long MCMC runs are necessary. The use of information about the latent state variable z helps to obtain convergence more swiftly.