CHAPTER

12

Estimation of Abundance from Counts in Metapopulation Designs Using the Binomial Mixture Model

OUTLINE

12.1 INTRODUCTION

The last two chapters of this book deal with the modeling of abundance and occurrence in a metapopulation design. We show how a sort of non-standard generalized linear mixed model (GLMM), a logistic regression

Bayesian Population Analysis using WinBUGS
DOI: 10.1016/B978-0-12-387020-9.00012-2

with Poisson or with Bernoulli random effects, can be used to estimate population size or species occurrence in systems of spatial replicate populations. At least some of them must be surveyed more than once during a short period, that is, replication is required in space *and* in time. Short means that the dynamics of the collection of populations (extinction, colonization, or emigration and immigration, as well as survival and recruitment) must be negligible over the time period over which replicate surveys are conducted. We call the design of studies with such systems of spatial replicate populations with temporally replicated samples a *metapopulation design* (Royle, 2004c; Kéry and Royle, 2010). The system of spatial replicates may or may not be inhabited by a metapopulation in the biological sense (Hanski, 1994, 1998).

In Chapters 4 and 5, we used two variants of hierarchical model that attempt to partition the observed data into contributions from the dynamics of the true ecological state and from an observation process. These models are useful because failure to distinguish between the two processes that generate the observed data will often lead to severely biased inferences, for example, about the presence or magnitude of density dependence. However, if any further information is absent (e.g., covariates informative about the observation process), these models cannot account for patterns in detection probability (see Link and Sauer, 1998 for good examples of such covariate modeling as a partial remedy for imperfect and patterned detection probability). The models in Chapters 4 and 5 are unable in principle to fully correct for the observation error. We have shown that with a single count per time interval, this framework simply models the expectation Np, where N is population size, and p is the average detection probability. What in these models is called "observation error" is simply the ups and downs around Np of the observed counts due to binomial sampling error. Therefore, such hierarchical models have been called implicit hierarchical models (Royle and Dorazio, 2008): the product Np is not a quantity with an explicit biological meaning.

In the present chapter, we extend these implicit hierarchical models for counts to become explicit hierarchical models for counts so that the two main parameters have the interpretation of local abundance N and of detection probability p. The binomial (also called N-) mixture model of Royle (2004c) jointly estimates local abundance and detection probability (Dodd and Dorazio, 2004; Royle, 2004a, 2004c; Kéry et al., 2005b; Dorazio, 2007; Royle and Dorazio, 2008; Wenger and Freeman, 2008; Joseph et al., 2009; Kéry et al., 2009a; Kéry and Royle, 2010; Post van der Burg et al., 2011). It takes as input spatially and temporally replicated counts of independent individuals within a period of closure and yields estimates of the parameters of the ecological and the observation processes. The parameters of the ecological process describe the spatial or spatio-temporal

variation in latent abundance. Abundance is a latent state because it is incompletely observed owing to detection error. The observation process is the process of detecting (or overlooking) individuals and is described by a binomial process as we have seen so many times.

The binomial mixture model has great appeal, since it allows us to estimate abundance, corrected for imperfect detection, from fairly "cheap" data: simple counts without any extra information, such as individual identification or distance measurements. Therefore, this model may be more widely applicable than capture–recapture methods or distance sampling. What the model does require, though, is replication in two dimensions, both spatially (at >10–20 sites, say) and temporally within a period of closure (at least two observations per site, though not necessarily at every site). Hence, it requires counts $y_{i,j}$ for a number of sites i and temporal replicates j.

Conceptually, such replicated counts arise from two distinct processes, one ecological and another observational. Accordingly, the simplest binomial mixture model for a single period of closure (season) can be written succinctly in just two lines:

$$N_i \sim \text{Poisson}(\lambda) \qquad \text{1. Ecological process yields latent state}$$

$$y_{i,j} \,|\, N_i \sim \text{Binomial}(N_i, p) \qquad \text{2. Observation process yields observations}$$

First, the spatial variation of local abundance at site i, N_i, for a collection of sites is described by a Poisson distribution with mean λ. Second, the observed counts $y_{i,j}$ (given N_i) at site i and during replicate survey j are described by a binomial distribution with sample size N_i and detection probability p. This model is sometimes also called a Poisson-binomial mixture model.

If we think of it, few things could be more natural than making these two distributional assumptions. The Poisson is the standard distribution assumed for spatial or temporal variation in abundance (McCullagh and Nelder, 1989), and the binomial distribution underlies a vast array of capture–recapture models as a formal description of the coin-flip-like detection process of individuals (Williams et al., 2002; see also Section 1.3). Thus, the binomial mixture model can be called a hierarchical Poisson regression, since the basic model for abundance is Poisson, but there is a logistic regression attached to account for imperfect detection. Several important extensions are possible, among them the adoption of distributions other than the Poisson for abundance or the binomial for detection, the introduction of covariates, and the relaxation of the independence of detection and of the closure assumption. We next briefly sketch each one in turn.

First, we could specify distributions other than a Poisson for the ecological part of the model, for instance, a negative binomial (Royle, 2004b, 2004c; Kéry et al., 2005b; Joseph et al., 2009), or a zero-inflated Poisson (Wenger and Freeman, 2008; Joseph et al., 2009). We will see a zero-inflated Poisson in Section 12.3.2 and the Poisson log-normal as an alternative to a negative binomial distribution in Section 12.3.3.

Second, since the binomial mixture model consists simply of two linked GLMs, it is natural to introduce the effects of covariates through a log- and a logit-link function, respectively, for abundance and detection. This means that neither the mean local abundance λ nor detection probability p needs to be constant; indeed, they rarely ever are! Rather, we can specify covariate relationships such as these:

$$\log(\lambda_i) = \alpha_0 + \alpha_1 * x_i \text{ and}$$

$$\text{logit}(p_{i,j}) = \beta_0 + \beta_1 * x_{i,j}$$

In the first case, mean abundance at site i, on the scale of the natural logarithm, is a linear function of site-specific covariate x_i (a "site covariate"), with intercept α_0 and slope α_1. Thus, the binomial mixture model allows one to model habitat relationships in abundance while accounting for detection error; for examples, see Kéry (2008), Webster et al. (2008), Chandler et al. (2009a, b), and Schlossberg et al. (2010). In the second case, the logit transform of detection probability at site i during survey j is a logit-linear function of the site- and survey-specific covariate $x_{i,j}$ (a "survey covariate"), with intercept β_0 and slope β_1. Of course, a site covariate (x_i) is also possible for detection.

Similarly, for abundance or detection, latent structure can be added by the introduction of random effects on the scale of the linear predictor. These account for additional variation in abundance or detection that is not accounted for by the nominal distributional assumptions along with the specified covariates. For instance, we can model extra-Poisson dispersion in the latent abundance parameters N_i by specifying the following Poisson-log-normal binomial mixture model (only linear predictor shown):

$$\log(\lambda_i) = \alpha_0 + \alpha_1 * x_i + \varepsilon_i,$$

$$\text{with } \varepsilon_i \sim \text{Normal}(0, \sigma^2_\lambda)$$

This modeling can be seen as a sort of correction for overdispersion in abundance (see also Section 4.2). Its result will be similar to the adoption of a negative binomial distribution instead of a Poisson. Random effects can likewise be introduced into the linear predictor for detection. Both will account for the increased uncertainty in parameter estimates owing to the effects of unmodeled covariates by spreading out the posterior distributions and, hence, increasing uncertainty intervals. We see an example in Section 12.3.3.

Third, the model assumes that all N_i individuals in each local population i behave and are detected independently. This assumption may be violated for animals that live in groups so that when one individual in a group is detected, others in the same group are more likely to be detected as well. An extension of the basic binomial mixture model to this situation has recently been described by Martin et al. (2011).

Fourth, the model as described so far is for static situations where replicate counts are available for a single, closed population of size N_i at each site i. This ideal will be impossible to attain in many situations, and indeed, in many cases, changes in abundance, for instance trends, are the focus of the modeling. The binomial mixture model can easily be extended to dynamic situations, for example, to several breeding seasons k, provided that data are available in the so-called robust design (Williams et al., 2002), with temporal replicate observations within each of multiple seasons. We then model counts $y_{i,j,k}$ from site i, replicate j, and season k and estimate different parameters for each season in the ecological process and possibly also for the observation process. The abundance parameters may be related to each other across years, for instance, to model a trend (Royle and Dorazio, 2008; Kéry and Royle, 2010; Kéry et al., 2010a). We will see an example of this in Section 12.3. In principle, it would also be possible to model dynamic (autoregressive) population models, such as the Ricker or Gompertz equations (Dennis et al., 2006), or the models in Chapter 5, within a binomial mixture framework, but this has not been done so far.

Key assumptions of the model are the following:

1. The ecological state is constant during the period over which replicate surveys are conducted (the traditional closure assumption). Its violation will lead to inflated abundance estimates. In benign cases, this may mean that N estimates the size of some superpopulation, that is, the number of all individuals that *ever* use a sample site during the surveys. In severe cases, though, N estimates may no longer be meaningful.
2. Detection probability is constant for all N_i individuals present at time j and equal to $p_{i,j}$. The model does not require individual identification but is not able to accommodate individual variation in detection probability either. In analogy to closed models (Chapter 6), intuition suggests that individual heterogeneity in detection probability at site i during period j leads to a negative bias in the abundance estimator (see Efford and Dawson (2009) for the special case of distance-related heterogeneity in detection).
3. The distribution of abundance N_i is adequately described by the chosen parametric form (e.g., a Poisson, possibly with covariates and latent effects). Similarly, detection probability is modeled adequately by the chosen parametric distribution (including possible covariates and other model structure). Especially the assumption about N_i is likely to be

more difficult to meet than the analogous assumption about occurrence in the site-occupancy models in Chapter 13. Effects of deviations of the data from the parametric model assumptions are hard to gauge, but posterior predictive checks (Section 12.3.) enable one to diagnose whether a model fits adequately.

4. There are no false positives such as double counts. The effect of the violation of this assumption has not been studied so far but is likely to induce a positive bias in the abundance estimator.

In Section 12.2, we first generate and analyze data from a single season (i.e., period of closure). In Section 12.3, we look at real-world data from multiple seasons in an insect species. A season may be one annual breeding season (for instance, for birds or reptiles, Kéry et al., 2005b, 2009a) or a single day, as in our insect example. We will fit a progression of increasingly complex models and see examples of zero-inflation and overdispersion correction. We will revisit (after their introduction in Chapter 7) an important and very general technique for goodness-of-fit assessment called a posterior predictive check (or Bayesian *p*-value).

12.2 GENERATION AND ANALYSIS OF SIMULATED DATA

12.2.1 The Simplest Case with Constant Parameters

To see the conceptual simplicity and beauty of the binomial mixture model, we first look at the simplest possible case, where both the ecological and the observation processes are described by an intercept only. To generate count data y under this Null model for R = 200 spatial replicates (sites) and T = 3 temporal replicates, we execute the following R commands:

```
# Determine sample sizes (spatial and temporal replication)
R <- 200
T <- 3

# Create structure to contain counts
y <- array(dim = c(R, T))

# Sample abundance from a Poisson(lambda = 2)
N <- rpois(n = R, lambda = 2)

# Sample counts from a Binomial(N, p = 0.5)
for (j in 1:T){
    y[,j] <- rbinom(n = R, size = N, prob = 0.5)
    }

# Look at realization of biological and observation processes
cbind(N, y)
```

We have assumed a mean abundance per site of 2 and mean detection per individual of 0.5. Note that in this model, the detection

parameter refers to each individual, whereas in the site-occupancy model (Chapter 13) it refers to the collection of *all* individuals inhabiting a site, that is, to an occupied site. Now, we will try to recover these parameter values when fitting the model in WinBUGS. Note how similar the BUGS code for this model is to the hierarchical way the data were created. Also, note that for this class of models, initial values for the latent states (the Ns) must sometimes be close to the solution; otherwise, WinBUGS may not achieve convergence or only do so very slowly. As our best guess, we choose the observed maximum count at each site, increased by 1, to save WinBUGS from having to update a Binomial with index 0.

```
# Specify model in BUGS language
sink("model.txt")
cat("
model {

# Priors
lambda ~ dgamma(0.005, 0.005)   # Standard vague prior for lambda
# lambda ~ dunif(0, 10)         # Other possibility
p ~ dunif(0, 1)

# Likelihood
# Biological model for true abundance
for (i in 1:R) {
    N[i] ~ dpois(lambda)
    # Observation model for replicated counts
    for (j in 1:T) {
        y[i,j] ~ dbin(p, N[i])
        } #j
    } #i
}
",fill = TRUE)
sink()

# Bundle data
win.data <- list(y = y, R = nrow(y), T = ncol(y))

# Initial values
Nst <- apply(y, 1, max) + 1 # This line is important
inits <- function() list(N = Nst)

# Parameters monitored
params <- c("lambda", "p")

# MCMC settings
ni <- 1200
nt <- 2
nb <- 200
nc <- 3

# Call WinBUGS from R (BRT 0.1 min)
out <- bugs(win.data, inits, params, "model.txt", n.chains = nc,
   n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
   bugs.dir, working.directory = getwd())
```

```
# Summarize posteriors
print(out, dig = 2)
        mean   sd  2.5%   25%   50%   75% 97.5%  Rhat  n.eff
lambda  2.20 0.18  1.89  2.08  2.19  2.30  2.57  1.01    440
p       0.48 0.03  0.42  0.46  0.49  0.51  0.55  1.03    120
```

This looks good. There is a fair amount of sampling variance and so repeated realizations of the analysis will yield fairly variable estimates.

12.2.2 Introducing Covariates

Next, we simulate more complex data for a single season. We introduce a single covariate that acts on both the ecological and the observation process. We model the effect of the covariate on the scale of the log and the logit, as is customary for generalized linear models. This example illustrates a so-called site covariate, that is, a covariate that varies by site only, but not among individual surveys (this would be called a survey or sampling covariate). Sampling covariates may be weather condition or survey duration, that is, something that may affect detection but not abundance. For sampling covariates, see the exercises at the end of this chapter. OpenBUGS (Examples > Ecology examples > Lizards) also contains an example of a binomial mixture model with covariates for both abundance and detection.

```
# Define function for generating binom-mix model data
data.fn <- function(R = 200, T = 3, xmin = -1, xmax = 1, alpha0 = 1,
   alpha1 = 3, beta0 = 0, beta1 = -5){

# R: number of sites at which counts were made (= number of spatial reps)
# T: number of times that counts were made at each site
# (= number of temporal reps)
# xmin, xmax: define range of the covariate X
# alpha0 and alpha1: intercept and slope of log-linear regression
# relating abundance to the site covariate A
# beta0 and beta1: intercept and slope of logistic-linear regression
# of detection probability on A

    y <- array(dim = c(R, T)) # Array for counts

    # Ecological process
    # Covariate values: sort for ease of presentation
    X <- sort(runif(n = R, min = xmin, max = xmax))

    # Relationship expected abundance - covariate
    lam <- exp(alpha0 + alpha1 * X)

    # Add Poisson noise: draw N from Poisson(lambda)
    N <- rpois(n = R, lambda = lam)
    table(N)                  # Distribution of abundances across sites
    sum(N > 0) / R            # Empirical occupancy
    totalN <- sum(N) ; totalN
```

```
# Observation process
# Relationship detection prob - covariate
p <- plogis(beta0 + beta1 * X)

# Make a 'census' (i.e., go out and count things)
for (i in 1:T){
    y[,i] <- rbinom(n = R, size = N, prob = p)
    }

# Naïve regression
naive.pred <- exp(predict(glm(apply(y, 1, max) ~ X + I(X^2),
   family = poisson)))

# Plot features of the simulated system
par(mfrow = c(2, 2))
plot(X, lam, main = "Expected abundance", xlab = "Covariate",
   ylab = "lambda", las = 1, type = "l", col = "red", lwd = 3,
   frame.plot = FALSE)
plot(X, N, main = "Realised abundance", xlab = "Covariate", ylab =
   "N", las = 1, frame.plot = FALSE, col = "red", cex = 1.2)
plot(X, p, ylim = c(0, 1), main = "Detection probability", xlab =
   "Covariate", ylab = "p", type = "l", col = "red", lwd = 3, las = 1,
   frame.plot = FALSE)
plot(X, naive.pred, main = "Actual counts \n and naïve regression",
   xlab = "Covariate", ylab = "Relative abundance", ylim = c(min(y),
   max(y)), type = "l", lty = 2, lwd = 4, col = "blue", las = 1,
   frame.plot = FALSE)
points(rep(X, T), y, col = "black", cex = 1.2)

# Return stuff
return(list(R = R, T = T, X = X, alpha0 = alpha0, alpha1 = alpha1,
   beta0 = beta0, beta1 = beta1, lam = lam, N = N, totalN = totalN,
   p = p, y = y))
}
```

We execute this function once to generate one data set and produce an overview of the simulation.

```
data <- data.fn()
str(data)
```

Figure 12.1 shows the main features of this stochastic system and its observation, that is, the data set generated. Abundance has a positive relation with the covariate (Fig. 12.1a and b). In contrast, detection probability has a negative relationship with that same covariate (Fig. 12.1c). The result of this is that the observed counts (Fig. 12.1d) suggest an intermediate optimum value of the covariate for abundance, and this false impression is confirmed by fitting a quadratic covariate effect in a Poisson regression of the max count at each site. The dashed blue line shows the prediction from that naïve analysis.

Next, we use a binomial mixture model to see whether we can tease apart the opposing effects of the covariate on the ecological state and on

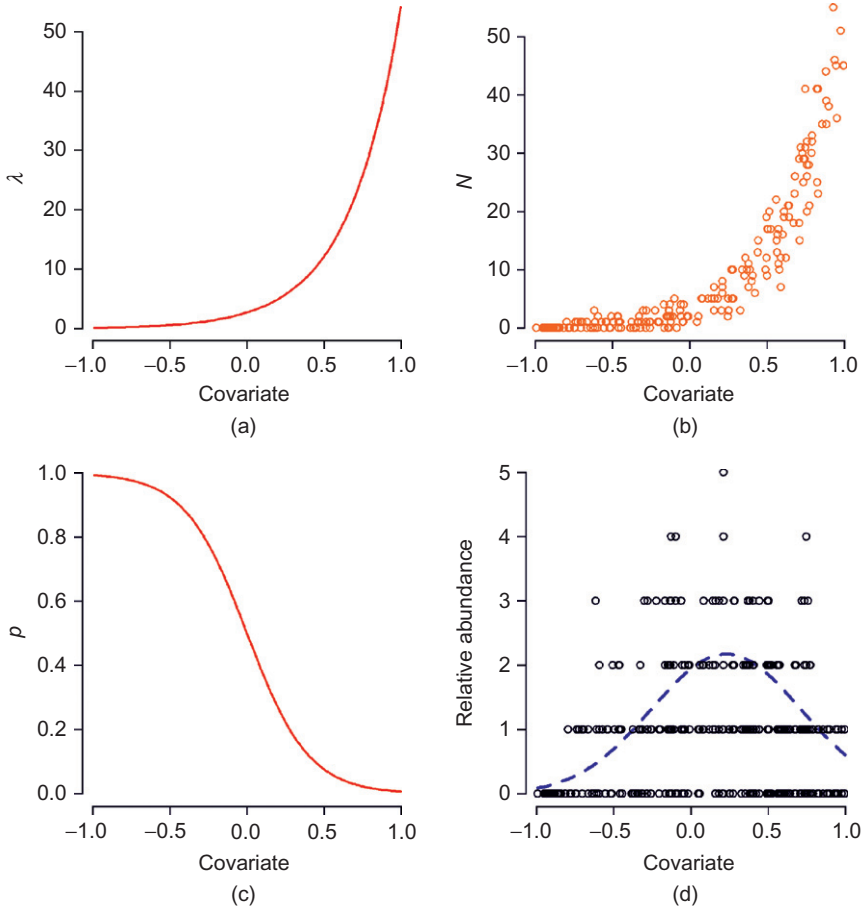

FIGURE 12.1 Features of the ecological and of the observation process that generated our data set and a conventional (naïve) analysis of counts in relation to an environmental covariate (dashed blue line in (d)): (a) Expected abundance, (b) realized abundance, (c) detection probability, (d) actual counts and naïve regression. The truth is shown in red and observed data in black. See text and R code for further explanations.

the observation of that state or, in plain English, whether we can recover estimates of the two GLM regressions that resemble the known input values. We are also interested in an estimate of the total population size across all surveyed plots, which, in our simulated data, was 1938.

```
# Specify model in BUGS language
sink("model.txt")
cat("
model {
```

```
# Priors
alpha0 ~ dunif(-10, 10)
alpha1 ~ dunif(-10, 10)
beta0 ~ dunif(-10, 10)
beta1 ~ dunif(-10, 10)

# Likelihood
# Ecological model for true abundance
for (i in 1:R){
    N[i] ~ dpois(lambda[i])
    log(lambda[i]) <- alpha0 + alpha1 * X[i]

    # Observation model for replicated counts
    for (j in 1:T){
        y[i,j] ~ dbin(p[i,j], N[i])
        p[i,j] <- exp(lp[i,j])/(1+exp(lp[i,j]))
        lp[i,j] <- beta0 + beta1 * X[i]
        } #j
    } #i

# Derived quantities
totalN <- sum(N[])
}
",fill = TRUE)
sink()

# Bundle data
y <- data$y
win.data <- list(y = y, R = nrow(y), T = ncol(y), X = data$X)

# Initial values
Nst <- apply(y, 1, max) + 1 # Important to give good inits for latent N
inits <- function() list(N = Nst, alpha0 = runif(1, -1, 1), alpha1 = runif
   (1, -1, 1), beta0 = runif(1, -1, 1), beta1 = runif(1, -1, 1))

# Parameters monitored
params <- c("totalN", "alpha0", "alpha1", "beta0", "beta1")

# MCMC settings
ni <- 22000
nt <- 20
nb <- 2000
nc <- 3

# Call WinBUGS from R (BRT 4 min)
out <- bugs(win.data, inits, params, "model.txt", n.chains = nc,
   n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
   bugs.dir, working.directory = getwd())

# Summarize posteriors
print(out, dig = 3)
```

```
            mean      sd     2.5%      25%      50%      75%    97.5%  Rhat  n.eff
totalN  1866.708 574.032  982.975 1464.750 1759.000 2192.000 3214.225 1.068     37
alpha0     0.981   0.123    0.736    0.900    0.983    1.064    1.223 1.027     83
```

```
alpha1      2.941   0.319   2.303   2.726   2.931   3.151   3.575  1.060     39
beta0       0.191   0.187  -0.178   0.064   0.192   0.319   0.551  1.023     98
beta1      -5.078   0.370  -5.806  -5.328  -5.078  -4.834  -4.339  1.040     58
deviance  853.819  14.566 826.897 843.875 853.000 863.300 883.700  1.013    160
```

We recover estimates that appear unbiased. The estimate of total population size is fairly imprecise, but will often, and here does, contain the true value (1938) within its 95% CRI. The sum of the maximum counts at each site, a conventional estimator of total population size, is only 240 individuals (type `sum(apply(data$y, 1, max))`). It is remarkable that the binomial mixture model gets so close to the truth! Let us look at further inferences and plot the posterior distribution for the four regression parameters (Fig. 12.2).

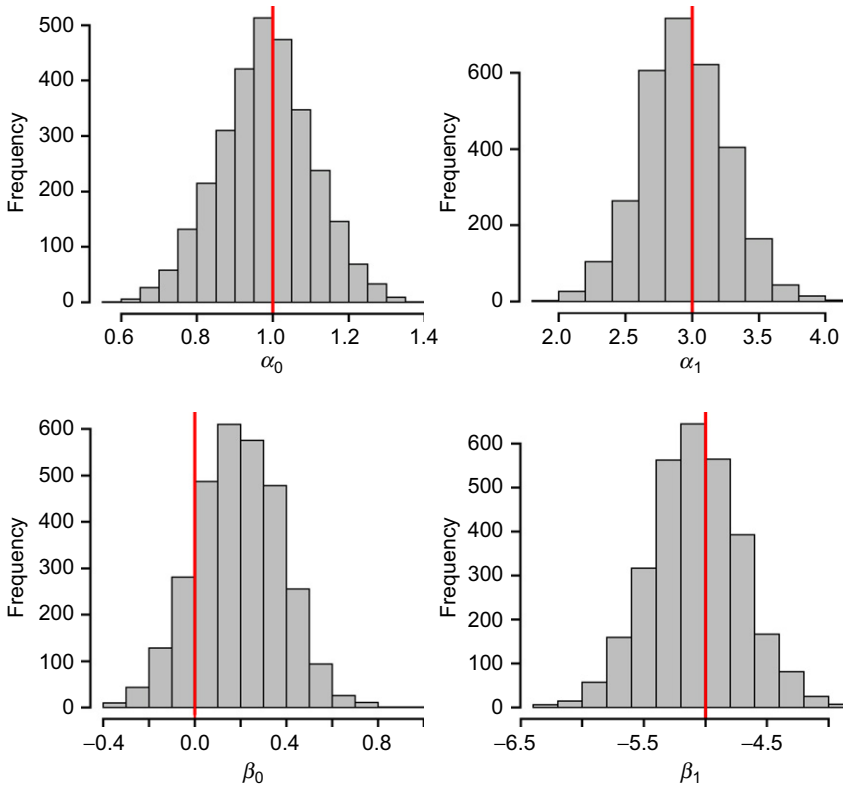

FIGURE 12.2 Posterior distributions for the two log- and logit-linear regressions of expected abundance and detection probability on a covariate. The red lines show the values used for these parameters in the simulation of the data.

Typically, an ecologist would want estimates of abundance for each site also. Within the Bayesian framework, this is trivial: just add N to the list of quantities monitored!

```
# Plot posteriors
par(mfrow = c(2, 2))
hist(out$sims.list$alpha0, col = "gray", main = "", xlab = "alpha0",
   las = 1)
abline(v = data$alpha0, lwd = 3, col = "red")
hist(out$sims.list$alpha1, col = "gray", main = "", xlab = "alpha1",
   las = 1)
abline(v = data$alpha1, lwd = 3, col = "red")
hist(out$sims.list$beta0, col = "gray", main = "", xlab = "beta0", las=1)
abline(v = data$beta0, lwd = 3, col = "red")
hist(out$sims.list$beta1, col = "gray", main = "", xlab = "beta1", las = 1)
abline(v = data$beta1, lwd = 3, col = "red")
```

How well can the naive and the binomial mixture analysis recover the covariate relationship? Figure 12.3 shows that for the conventional model, the answer is: not very well! In contrast, the binomial mixture model seems to do an excellent job at recovering the true positive relationship between abundance and the covariate.

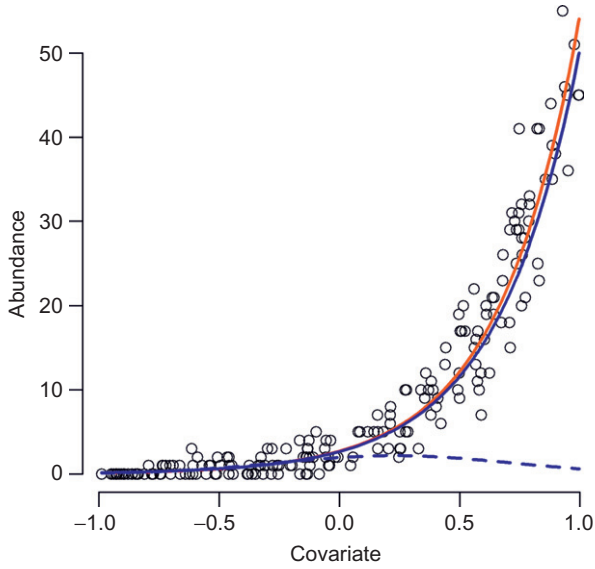

FIGURE 12.3 Relationship between abundance and covariate in a simulated system. The red line shows truth, with realized abundance at 200 sites in black. The blue lines show the estimates under a binomial mixture model (solid line) and under a conventional nonhierarchical Poisson regression (dashed).

```
# Plot predicted covariate relationship with abundance
plot(data$X, data$N, main = "", xlab = "Covariate", ylab = "Abundance",
   las = 1, ylim = c(0, max(data$N)), frame.plot = FALSE)
lines(data$X, data$lam, type = "l", col = "red", lwd = 3)
GLM.pred <- exp(predict(glm(apply(data$y, 1, max) ~ X + I(X^2),
   family = poisson, data = data)))
lines(data$X, GLM.pred, type = "l", lty = 2, col = "blue", lwd = 3)
Nmix.pred <- exp(out$mean$alpha0 + out$mean$alpha1 * data$X)
points(data$X, Nmix.pred, type = "l", col = "blue", lwd = 3)
```

12.3 ANALYSIS OF REAL DATA: OPEN-POPULATION BINOMIAL MIXTURE MODELS

Frequently, counts are made at two temporal scales. That is, one does not only have count data from a single period of closure, but in addition from several seasons. For instance, for birds, we may have repeated samples within a breeding season (visits within a year are secondary occasions) that are repeated across several years (years are primary occasions). In capture–recapture, this sampling design is called the robust design (Williams et al., 2002): a population is repeatedly sampled over a short period, during which the population is considered to be closed, and these sampling sessions are repeated over a longer period, over which the population is considered to be open. The primary occasions may also consist of (much) shorter time intervals than years if the population dynamics of the study organism is fast relative to the duration of a study.

In our real-world example, we will estimate population size of a butterfly, the silver-washed fritillary (Fig. 12.4), over an entire summer. The information for application of the binomial mixture model comes from the fact that each of 95 sites was surveyed twice along a 2.5 km transect on each of 4–7 survey days (Kéry et al., 2009a; Dorazio et al., 2010). Survey days are separated by at least two weeks, and butterflies are rather short-lived; hence, it would not be sensible to assume a constant population size at each site over repeated days or even during the entire summer. Hence, we shall consider each day a primary occasion and model different parameters for each day. The two replicate observations of each transects within a day represent the secondary occasions between which the populations are assumed constant. We will apply the open-population binomial mixture model of Dorazio and Royle (2008), Kéry et al. (2009a), and Kéry and Royle (2010) to estimate population size during each day.

We denote as $y_{i,j,k}$ the count from site i ($i = 1 \ldots 95$), within-day temporal replicate j ($j = 1, 2$), and day k ($k = 1 \ldots 7$). Note that both j and k index temporal replicates, but the population is assumed to be static over the former and allowed to change over the latter. In program R, we format

FIGURE 12.4 Silver-washed fritillary *Argynnis paphia*, Switzerland, 2005 (Photograph by T. Marent).

the counts into a three-dimensional array, since this is convenient for the modeling. In our experience, one of the most difficult things about statistical modeling in WinBUGS is to keep track of the dimensions of multidimensional arrays, including putting data into such arrays in the first place. So, expect some time to learn how to do this, either in a clumsy way like we do here or using R functions like those in the reshape package.

We will revisit posterior predictive distributions and Bayesian *p*-values in this example, introduced in Section 7.10, a very general method of checking the goodness-of-fit of a model fit using Bayesian posterior sampling (see Gelman et al., 2004; Gelman et al., 1996, Kéry, 2010). The posterior predictive distribution is the distribution of replicate data sets or statistics computed from data sets, given the model, its parameter values, and the observed data set. We do a posterior predictive check of whether the model used to analyze the observed data fits them, in the sense that the model, with the estimated parameters, could plausibly have generated data such as the data set that we actually observed. The idea behind a posterior predictive check is rather similar to a parametric bootstrap: use the parameter values estimated from the actual data set under a given model to generate new data sets. Then, calculate some discrepancy measure for the new data set (e.g., chi-squared) and repeat that many times to get the reference distribution for that discrepancy measure for a model

that fits. Then, compare the fit of the model with the actual data with that reference distribution. For example, we see whether a, say, chi-squared test statistic computed for the actual data set falls within the distribution of chi-squared statistics that was computed for the replicate data sets.

We load the data, put them into a 3D array, and look at some summary statistics.

```
# Get the data and put them into 3D array
bdat <- read.table("fritillary.txt", header = TRUE)
y <- array(NA, dim = c(95, 2, 7))   # 95 sites, 2 reps, 7 days

for(k in 1:7){
    sel.rows <- bdat$day == k
    y[,,k] <- as.matrix(bdat)[sel.rows, 3:4]
    }
y                       # Look at data set in 3D layout
str(y)

# Have a look at raw data
day.max <- apply(y, c(1, 3), max, na.rm = TRUE)    # Max count each site
                                                     and day
day.max
site.max <- apply(day.max, 1, max, na.rm = TRUE)  # Max count each site
site.max
table(site.max)             # Frequency distribution of max counts
plot(table(site.max))
table(site.max>0)           # Observed occupancy is only 56%

# Sum of observed max as conventional estimator of total abundance
max1 <- apply(y, c(1, 3), max)
obs.max.sum <- apply(max1, 2, sum, na.rm = TRUE)

obs.max.sum
[1]  4  0 15 32 99 85 63
```

Very few butterflies were observed during the first day and none during the second day.

12.3.1 Simple Poisson Model

We start with the simplest binomial mixture model that appears to make sense in this case. It has time-specific, constant parameters for both abundance and detection probability. Note that in the computation of the chi-squared discrepancy measure for the posterior predictive check, a small constant (0.5) is added in the denominator to avoid possible divisions by zero.

```
# Specify model in BUGS language
sink("Nmix0.txt")
cat("
model {
```

```
# Priors
for (k in 1:7){
    alpha.lam[k] ~ dnorm(0, 0.01)
    p[k] ~ dunif(0, 1)
    }

# Likelihood
# Ecological model for true abundance
for (k in 1:7){                          # Loop over days (7)
    lambda[k] <- exp(alpha.lam[k])
    for (i in 1:R){                      # Loop over R sites (95)
       N[i,k] ~ dpois(lambda[k])         # Abundance

       # Observation model for replicated counts
       for (j in 1:T){                   # Loop over temporal reps (2)
          y[i,j,k] ~ dbin(p[k], N[i,k])  # Detection

          # Assess model fit using Chi-squared discrepancy
          # Compute fit statistic E for observed data
          eval[i,j,k] <- p[k] * N[i,k]    # Expected values
          E[i,j,k] <- pow((y[i,j,k] - eval[i,j,k]),2) / (eval[i,j,k] +
             0.5)
          # Generate replicate data and compute fit stats for them
          y.new[i,j,k] ~ dbin(p[k], N[i,k])
          E.new[i,j,k] <- pow((y.new[i,j,k] - eval[i,j,k]),2) /
             (eval[i,j,k] + 0.5)

          } #j
       } #i
    } #k

# Derived and other quantities
for (k in 1:7){
    totalN[k] <- sum(N[,k])     # Total pop. size across all sites
    mean.abundance[k] <- exp(alpha.lam[k])
    }
fit <- sum(E[,,])
fit.new <- sum(E.new[,,])
}
",fill = TRUE)
sink()

# Bundle data
R = nrow(y)
T = ncol(y)
win.data <- list(y = y, R = R, T = T)

# Initial values
Nst <- apply(y, c(1, 3), max) + 1
Nst[is.na(Nst)] <- 1
inits <- function(){list(N = Nst, alpha.lam = runif(7, -1, 1))}

# Parameters monitored
params <- c("totalN", "mean.abundance", "alpha.lam", "p", "fit",
   "fit.new")
```

```
# MCMC settings
ni <- 10000
nt <- 8
nb <- 2000
nc <- 3

# Call WinBUGS from R (BRT 1 min)
out0 <- bugs(win.data, inits, params, "Nmix0.txt", n.chains = nc,
   n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
   bugs.dir, working.directory = getwd())

# Summarize posteriors
print(out0, dig = 3)
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
totalN[1]	35.133	59.426	4.000	8.000	15.000	32.000	233.148	1.090	49
totalN[2]	0.162	0.902	0.000	0.000	0.000	0.000	2.000	1.132	330
totalN[3]	20.003	5.097	15.000	17.000	19.000	22.000	34.000	1.002	1800
totalN[4]	39.511	5.117	33.000	36.000	38.000	42.000	52.000	1.003	2400
totalN[5]	135.943	13.689	115.000	126.000	134.000	144.000	168.000	1.001	3000
totalN[6]	99.653	6.869	90.000	95.000	99.000	103.000	116.000	1.004	750
totalN[7]	90.296	10.605	74.000	83.000	89.000	96.000	115.000	1.001	3000
mean.abundance[1]	0.371	0.631	0.030	0.088	0.163	0.348	2.447	1.077	56
mean.abundance[2]	0.003	0.010	0.000	0.000	0.000	0.001	0.023	1.001	3000
mean.abundance[3]	0.211	0.071	0.109	0.165	0.201	0.245	0.389	1.002	1900
mean.abundance[4]	0.415	0.086	0.271	0.356	0.407	0.465	0.608	1.001	3000
mean.abundance[5]	1.430	0.191	1.091	1.299	1.415	1.549	1.843	1.001	3000
mean.abundance[6]	1.048	0.128	0.820	0.960	1.040	1.127	1.316	1.003	930
mean.abundance[7]	0.952	0.151	0.686	0.850	0.939	1.041	1.277	1.001	3000
[...]									
p[1]	0.212	0.167	0.012	0.077	0.170	0.312	0.625	1.070	60
p[2]	0.485	0.291	0.017	0.229	0.489	0.731	0.972	1.003	1300
p[3]	0.545	0.124	0.282	0.465	0.552	0.634	0.766	1.002	3000
p[4]	0.600	0.087	0.416	0.543	0.605	0.661	0.761	1.002	1700
p[5]	0.515	0.057	0.401	0.475	0.518	0.556	0.619	1.001	3000
p[6]	0.653	0.054	0.539	0.619	0.656	0.692	0.750	1.002	1500
p[7]	0.550	0.068	0.416	0.505	0.552	0.597	0.677	1.001	3000
fit	147.924	15.887	121.000	136.675	146.400	157.625	183.102	1.001	3000
fit.new	97.796	11.908	76.427	89.490	97.220	104.900	123.000	1.002	3000
deviance	765.186	42.162	689.300	735.575	763.400	792.200	855.505	1.001	2700

We note that the estimates for the first two days, when very few or no butterflies were observed, are very imprecise and entirely driven by the priors, respectively: the 95% CRI for p at day 2 extends almost from 0 to 1. We look at the posterior predictive check of goodness-of-fit for this model (Fig. 12.5a).

```
# Evaluation of fit
plot(out0$sims.list$fit, out0$sims.list$fit.new, main = "", xlab =
   "Discrepancy actual data", ylab = "Discrepancy replicate data",
   frame.plot = FALSE)
abline(0, 1, lwd = 2, col = "black")
```

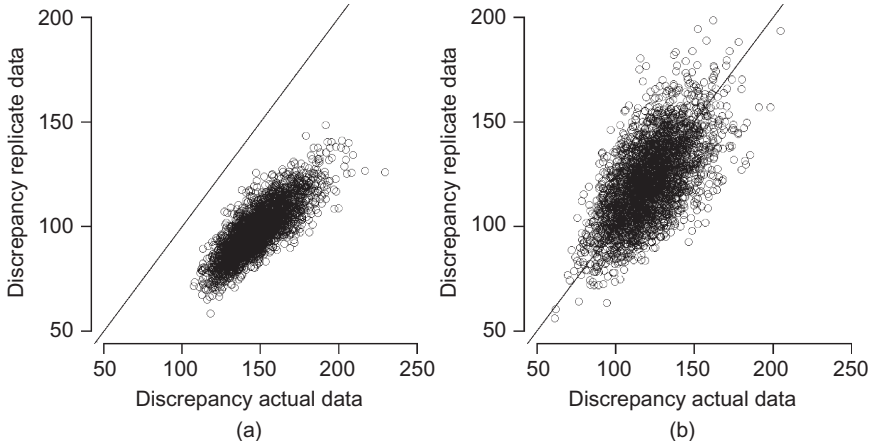

FIGURE 12.5 Posterior predictive checks of model adequacy of two binomial mixture models fit to the Swiss fritillary data; (a) Null model, (b) model with random effects in abundance and detection (see Section 12.3.3.).

```
mean(out0$sims.list$fit.new > out0$sims.list$fit)
[1] 0

mean(out0$mean$fit) / mean(out0$mean$fit.new)
[1] 1.512573
```

The model does not fit well at all (Fig. 12.5a). Indeed, there is a "lack-of-fit ratio" of 1.51. This informal quantity compares the mean of the fit statistic for the actual data with that for the perfect data sets and thus gives a numerical expression of how bad the lack of fit is. What could be wrong? The fritillary was never detected at 42 sites and detected at least once at 53; yielding an observed occupancy of 56%. It may well be that a Poisson distribution for the spatial variation in abundance is not flexible enough to account for that many zeroes. Therefore we will next model abundance with a zero-inflated Poisson distribution (Wenger and Freeman, 2008; Joseph et al., 2009).

12.3.2 Zero-Inflated Poisson Binomial Mixture Model (ZIP Binomial Mixture Model)

The ZIP binomial mixture model appears like a sensible model for these data. It divides the sites into those that are suitable and those that are not suitable. A Poisson distribution for abundance is assumed for suitable sites only. To obtain a ZIP binomial mixture model, we simply add another hierarchical layer to the model. This additional layer is binary: we decide by a coin flip whether a site is suitable in principle or not. Only if a site is suitable in principle will Nature roll her Poisson die

to determine the actual number of butterflies living there. Here is the resulting hierarchical model:

Level 1 (suitability of site i): $z_i \sim \text{Bernoulli}(\Omega)$
Level 2 (realized abundance at i, k): $N_{i,k} \mid z_i \sim \text{Poisson}(z_i \lambda_k)$
Level 3 (observed count at i, j, k): $y_{i,j,k} \mid N_{i,k} \sim \text{Binomial}(N_{i,k}, p_{i,j,k})$

We have three main structural parameters (Ω, λ_k, $p_{i,j,k}$), one for each level in the model hierarchy. We could model each of them as a function of covariates through a GLM link function. In our case, we do not have any covariates, so we simply fit group effects, that is, estimate a separate parameter for abundance and detection at every time period.

To describe the model in the BUGS language, we want to define the latent suitability indicators z first, that is, in the outermost loop of the likelihood definition. Thus, we simply flip the order in which we loop over the dimensions of the site-by-rep-by-day data array $y_{i,j,k}$.

```
# Specify model in BUGS language
sink("Nmix1.txt")
cat("
model {

# Priors
omega ~ dunif(0, 1)
for (k in 1:7){
    alpha.lam[k] ~ dnorm(0, 0.01)
    p[k] ~ dunif(0, 1)
    }

# Likelihood
# Ecological model for true abundance
for (i in 1:R){                          # Loop over R sites (95)
    z[i] ~ dbern(omega)                  # Latent suitability state
    for (k in 1:7){                      # Loop over survey periods (seasons)
        N[i,k] ~ dpois(lam.eff[i,k]) # Latent abundance state
        lam.eff[i,k] <- z[i] * lambda[i,k]
        log(lambda[i,k]) <- alpha.lam[k]
        # Observation model for replicated counts
        for (j in 1:T){                      # Loop over temporal reps (2)
            y[i,j,k] ~ dbin(p[k], N[i,k])  # Detection
            # Assess model fit using Chi-squared discrepancy
            # Compute fit statistic for observed data
            eval[i,j,k] <- p[k] * N[i,k]
            E[i,j,k] <- pow((y[i,j,k] - eval[i,j,k]),2) / (eval[i,j,k] +
               0.5)
            # Generate replicate data and compute fit stats for them
            y.new[i,j,k] ~ dbin(p[k], N[i,k])
            E.new[i,j,k] <- pow((y.new[i,j,k] - eval[i,j,k]),2) /
               (eval[i,j,k]+0.5)
            } #j
        } #k
    } #i
```

```
# Derived and other quantities
for (k in 1:7){
    # Estimate total pop. size across all sites
    totalN[k] <- sum(N[,k])
    mean.abundance[k] <- exp(alpha.lam[k])
    }
fit <- sum(E[,,])
fit.new <- sum(E.new[,,])
}
",fill = TRUE)
sink()
# Bundle data
R = nrow(y)
T = ncol(y)
win.data <- list(y = y, R = R, T = T)
# Initial values
Nst <- apply(y, c(1, 3), max) + 1
Nst[is.na(Nst)] <- 1
inits <- function(){list(N = Nst, alpha.lam = runif(7, -1, 1))}
# Parameters monitored
params <- c("omega", "totalN", "alpha.lam", "p", "mean.abundance",
   "fit", "fit.new")
# MCMC settings
ni <- 30000
nt <- 15
nb <- 15000
nc <- 3
# Call WinBUGS from R (BRT 3 min)
out1 <- bugs(win.data, inits, params, "Nmix1.txt", n.chains = nc,
   n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
   bugs.dir, working.directory = getwd())
# Summarize posteriors
print(out1, dig = 3)
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
omega	0.561	0.050	0.464	0.527	0.561	0.594	0.655	1.001	3000
totalN[1]	21.063	31.485	4.000	7.000	11.000	21.000	99.025	1.050	57
totalN[2]	0.145	0.756	0.000	0.000	0.000	0.000	2.000	1.068	300
totalN[3]	21.074	6.506	15.000	17.000	19.000	23.000	38.000	1.005	1000
totalN[4]	40.953	7.002	33.000	36.747	39.000	43.000	59.000	1.003	1900
totalN[5]	154.783	23.744	122.000	139.000	151.000	166.000	212.000	1.003	760
totalN[6]	105.251	9.486	91.975	99.000	104.000	110.000	128.000	1.002	3000
totalN[7]	91.561	14.688	72.000	81.000	89.000	98.000	129.025	1.002	2500
[...]									
p[1]	0.235	0.170	0.017	0.096	0.197	0.342	0.647	1.043	63
p[2]	0.479	0.295	0.016	0.212	0.474	0.736	0.970	1.001	3000
p[3]	0.527	0.132	0.262	0.437	0.535	0.624	0.766	1.006	580
p[4]	0.585	0.094	0.384	0.526	0.591	0.651	0.748	1.001	3000
p[5]	0.458	0.067	0.322	0.415	0.459	0.504	0.582	1.003	1000
p[6]	0.621	0.060	0.496	0.582	0.623	0.665	0.725	1.001	3000
p[7]	0.503	0.079	0.338	0.452	0.507	0.558	0.651	1.001	2400
[...]									

```
fit        147.329  16.573  119.900  135.700  145.800  157.000  184.500  1.003    780
fit.new     97.575  11.791   76.604   89.475   96.855  105.000  122.902  1.003    990
deviance   751.830  38.437  680.582  725.000  749.800  775.825  834.300  1.004    610
```

Under this model, the deviance is somewhat improved (here, from 765 to 752). The proportion of suitable sites, `omega`, is estimated identically with the proportion of sites at which the butterfly was ever detected. But does this model fit?

```
# Evaluation of fit
plot(out1$sims.list$fit, out1$sims.list$fit.new, main = "", xlab =
   "Discrepancy actual data", ylab = "Discrepancy replicate data",
   frame.plot = FALSE)
abline(0, 1, lwd = 2, col = "black")
mean(out1$sims.list$fit.new > out1$sims.list$fit)
[1] 0
mean(out1$mean$fit) / mean(out1$mean$fit.new)
[1] 1.509898
```

Unfortunately, the model still does not fit the data according to our chosen discrepancy measure, which is much greater for the actual data set than for the replicate data sets (i.e., the reference distribution of the test statistic). The Bayesian *p*-value, the proportion of symbols above the 1:1 line in the figure, is equal to zero. Compared with the model without zero-inflation, the lack-fit-ratio has barely gone down (from 1.513 to 1.510). Hence, a zero-inflated binomial mixture model is not flexible enough to capture the variability in the system adequately.

12.3.3 Binomial Mixture Model with Overdispersion in Both Abundance and Detection

Finally, we drop zero-inflation and instead try a model that accounts for extra-Poisson dispersion in both abundance and detection. The introduction of latent (random) effects into either or both linear predictors can be seen as sort of overdispersion correction, and it increases the uncertainty in the estimates. Thus, as in overdispersion correction in capture–recapture for instance (Burnham and Anderson, 2002), we buy a fitting model by losing precision in our estimates.

Level 1 (realized abundance at i, k):	$N_{i,k} \sim \text{Poisson}(\lambda_{i,k})$
GLM for level 1:	$\log(\lambda_{i,k}) = \alpha_k + \varepsilon_i$
Level 1b (random site effects):	$\varepsilon_i \sim \text{Normal}(0, \sigma^2_\lambda)$
Level 2 (observed count at i, j, k):	$y_{i,j,k} \mid N_{i,k} \sim \text{Binomial}(N_{i,k}, p_{i,j,k})$

GLM for level 2: $\text{logit}(p_{i,j,k}) = \beta_k + \delta_{i,j,k}$
Level 2b (random survey effects): $\delta_{i,j,k} \sim \text{Normal}(0, \sigma_p^2)$

So, we combine a naturally hierarchical model with two additional hierarchical levels, one for the sites in the ecological process and another for the individual surveys in the observation process. We assume the presence of "residual" site- and site-day-replicate-specific contributions to abundance (ε) and detection probability (δ), respectively. By specifying a Poisson-log-normal (Millar, 2009) model for the ecological state description, we account for the fact that there may be extra-Poisson dispersion in the distribution used to model spatio-temporal variation in the latent abundance parameters. Similarly, observation conditions may vary among sites, days, and replicates, and the random effect $\delta_{i,j,k}$ adequately accounts for that additional variability. As is customary for this type of modeling, we assume a normal distribution for both random noise terms. Note that for the extra variability in detection (delta), we could model $\delta_{i,j,k}$, $\delta_{i,k}$, or δ_i. Our treatment here is consistent with the published analyses in Kéry et al. (2009a) and Kéry and Royle (2010) as well as with an unpublished study by L. Tanadini and M. Kéry.

```
# Specify model in BUGS language
sink("Nmix2.txt")
cat("
model{

# Priors
for (k in 1:7){
    alpha.lam[k] ~ dnorm(0, 0.1)
    beta[k] ~ dnorm(0, 0.1)
    }

# Abundance site and detection site-by-day random effects
for (i in 1:R){
    eps[i] ~ dnorm(0, tau.lam)                  # Abundance noise
    }
tau.lam <- 1 / (sd.lam * sd.lam)
sd.lam ~ dunif(0, 3)
tau.p <- 1 / (sd.p * sd.p)
sd.p ~ dunif(0, 3)

# Likelihood
# Ecological model for true abundance
for (i in 1:R){                                 # Loop over R sites (95)
    for (k in 1:7){                             # Loop over days (7)
        N[i,k] ~ dpois(lambda[i,k])             # Abundance
        log(lambda[i,k]) <- alpha.lam[k] + eps[i]

        # Observation model for replicated counts
        for (j in 1:T){                         # Loop over temporal
                                                  reps (2)
```

```
            y[i,j,k] ~ dbin(p[i,j,k], N[i,k])    # Detection
            p[i,j,k] <- 1 / (1 + exp(-lp[i,j,k]))
            lp[i,j,k] ~ dnorm(beta[k], tau.p)    # Random delta defined
                                                   implicitly

            # Assess model fit using Chi-squared discrepancy
            # Compute fit statistic for observed data
            eval[i,j,k] <- p[i,j,k] * N[i,k]
            E[i,j,k] <- pow((y[i,j,k] - eval[i,j,k]),2) / (eval[i,j,k]
               +0.5)
            # Generate replicate data and compute fit stats for them
            y.new[i,j,k] ~ dbin(p[i,j,k], N[i,k])
            E.new[i,j,k] <- pow((y.new[i,j,k] - eval[i,j,k]),2) /
               (eval[i,j,k]+0.5)
            } #j
            ik.p[i,k] <- mean(p[i,,k])
        } #k
    } #i

# Derived and other quantities
for (k in 1:7){
    totalN[k] <- sum(N[,k])    # Estimate total pop. size across all
       sites
    mean.abundance[k] <- mean(lambda[,k])
    mean.N[k] <- mean(N[,k])
    mean.detection[k] <- mean(ik.p[,k])
    }
fit <- sum(E[,,])
fit.new <- sum(E.new[,,])
}
",fill = TRUE)
sink()

# Bundle data
R = nrow(y)
T = ncol(y)
win.data <- list(y = y, R = R, T = T)

# Initial values
Nst <- apply(y, c(1, 3), max) + 1
Nst[is.na(Nst)] <- 1
inits <- function(){list(N = Nst, alpha.lam = runif(7, -3, 3), beta =
   runif(7, -3, 3), sd.lam = runif(1, 0, 1), sd.p = runif(1, 0, 1))}

# Parameters monitored
params <- c("totalN", "alpha.lam", "beta", "sd.lam", "sd.p",
   "mean.abundance", "mean.N", "mean.detection", "fit", "fit.new")
```

Models with lots of random effects always need much longer to enable the Markov chains to mix properly. Hence, we greatly increase the number of MCMC iterations (obviously, we did this after some initial experimenting). We also increase the thinning rate to avoid having to save huge results files.

```
# MCMC settings
ni <- 350000
nt <- 300
nb <- 50000
nc <- 3
# Call WinBUGS from R (BRT 215 min)
out2 <- bugs(win.data, inits, params, "Nmix2.txt", n.chains = nc,
   n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
   bugs.dir, working.directory = getwd())
```

We get done after almost 4 h. First, we evaluate the fit of the model (Fig. 12.5b): it does fit now!

```
# Evaluation of fit
plot(out2$sims.list$fit, out2$sims.list$fit.new, main = "", xlab =
   "Discrepancy actual data", ylab = "Discrepancy replicate data",
   frame.plot = FALSE, xlim = c(50, 200), ylim = c(50, 200))
abline(0, 1, lwd = 2, col = "black")
mean(out2$sims.list$fit.new > out2$sims.list$fit)
[1] 0.505
mean(out2$mean$fit) / mean(out2$mean$fit.new)
[1] 0.999935
```

```
# Summarize posteriors
print(out2, dig = 2)
```

	mean	sd	2.5%	25%	50%	75%	97.5%	Rhat	n.eff
totalN[1]	94.48	163.99	5.00	13.00	35.00	95.00	593.05	1.01	370
totalN[2]	120.22	461.11	0.00	0.00	2.00	18.00	1762.90	1.36	25
totalN[3]	225.13	381.69	19.00	45.00	101.00	234.25	1537.05	1.06	55
totalN[4]	55.89	26.21	34.00	41.00	48.00	61.00	126.02	1.00	1100
totalN[5]	830.35	575.54	204.97	395.00	640.50	1119.25	2353.15	1.02	180
totalN[6]	128.36	30.90	96.00	109.00	121.00	138.00	209.00	1.01	390
totalN[7]	158.64	86.42	83.00	107.00	131.00	178.00	402.05	1.00	630
[...]									
sd.lam	1.87	0.23	1.46	1.70	1.84	2.01	2.37	1.00	1000
sd.p	1.05	0.21	0.70	0.91	1.03	1.17	1.50	1.00	980
mean.abundance[1]	1.00	1.73	0.04	0.14	0.37	1.00	6.30	1.01	340
mean.abundance[2]	1.27	4.86	0.00	0.01	0.03	0.18	18.73	1.07	48
mean.abundance[3]	2.37	4.01	0.18	0.48	1.07	2.46	16.44	1.06	55
mean.abundance[4]	0.59	0.29	0.31	0.43	0.52	0.66	1.38	1.00	1100
mean.abundance[5]	8.72	6.06	2.10	4.16	6.78	11.68	24.91	1.02	190
mean.abundance[6]	1.35	0.35	0.93	1.14	1.29	1.48	2.18	1.01	420
mean.abundance[7]	1.67	0.92	0.83	1.13	1.38	1.88	4.32	1.00	760
mean.N[1]	0.99	1.73	0.05	0.14	0.37	1.00	6.24	1.01	370
mean.N[2]	1.27	4.85	0.00	0.00	0.02	0.19	18.55	1.36	25
mean.N[3]	2.37	4.02	0.20	0.47	1.06	2.47	16.18	1.06	55
mean.N[4]	0.59	0.28	0.36	0.43	0.51	0.64	1.33	1.00	1100
mean.N[5]	8.74	6.06	2.16	4.16	6.74	11.78	24.77	1.02	180
mean.N[6]	1.35	0.33	1.01	1.15	1.27	1.45	2.20	1.01	390
mean.N[7]	1.67	0.91	0.87	1.13	1.38	1.87	4.23	1.00	630
mean.detection[1]	0.12	0.14	0.00	0.02	0.06	0.16	0.49	1.01	310

```
mean.detection[2]   0.23   0.32   0.00   0.01   0.04   0.39    0.99  1.06    62
mean.detection[3]   0.14   0.14   0.01   0.04   0.08   0.19    0.52  1.06    55
mean.detection[4]   0.47   0.14   0.18   0.38   0.48   0.56    0.71  1.01   810
mean.detection[5]   0.14   0.08   0.03   0.07   0.12   0.18    0.34  1.02   200
mean.detection[6]   0.51   0.10   0.29   0.45   0.52   0.58    0.69  1.01   460
mean.detection[7]   0.34   0.11   0.12   0.27   0.35   0.43    0.55  1.00   610
fit               121.41  18.67  88.48 108.60 120.20 133.22  161.41  1.01   240
fit.new           121.42  19.04  86.02 108.07 120.60 134.20  161.40  1.01   240
deviance          640.44  49.86 540.00 607.37 641.30 674.42  737.21  1.01   250
```

We note that convergence for some quantities associated with day 2 (when no fritillaries were observed at all) is less good. This illustrates the fact that often, though not always, lack of identifiability is associated with lack of convergence.

Since we have a fitting model now, we produce some plots of the estimates. In particular, we compare the mean count per day with the estimated mean abundance of the fritillary on each day. For mean daily abundance, we use the posterior median as our measure of central tendency, since the posterior is highly skewed (you can check that by typing `hist(out2$sims.list$mean.abundance[,4], breaks = 40)`).

```
max.day.count <- apply(y, c(1, 3), max, na.rm = TRUE)
max.day.count[max.day.count == "-Inf"] <- NA
mean.max.count <- apply(max.day.count, 2, mean, na.rm = TRUE)
mean.max.count

par(mfrow = c(2, 1))
plot(1:7, mean.max.count, xlab = "Day", ylab = "Mean daily abundance",
   las = 1, ylim = c(0, 16), type = "b", main = "", frame.plot = FALSE,
   pch = 16, lwd = 2)
lines(1:7, out2$summary[24:30,5], type = "b", pch = 16, col = "blue",
   lwd = 2)
segments(1:7, out2$summary[24:30,3], 1:7, out2$summary[24:30,7],
   col = "blue")

plot(1:7, out2$summary[38:44,1], xlab = "Day", ylab = "Detection
   probability ", las = 1, ylim = c(0, 1), type = "b", col = "blue",
   pch = 16, frame.plot = FALSE, lwd = 2)
segments(1:7, out2$summary[38:44,3], 1:7, out2$summary[38:44,7],
   col = "blue")
```

In Fig. 12.6, we see well that we now have a model that fits the fritillary counts, but that the fit comes at the expense of much less precision. Indeed, the Bayesian credible intervals are huge; much larger at any rate than under the two simpler models. This uncertainty around the estimates is a direct consequence of the introduction of the two sets of random effects. Thus, a refined analysis might try to get rid of one or both of them by introducing covariates that are informative about that variation in abundance or detection.

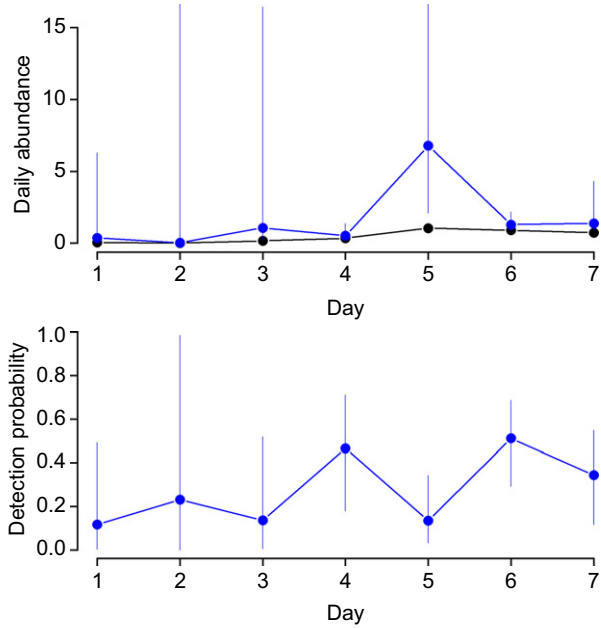

FIGURE 12.6 Abundance and detection of silver-washed fritillary in the Swiss biodiversity monitoring program under a binomial mixture model with random effects in the linear predictors of both abundance and detection. Top: Mean daily abundance per transect (black: raw counts, blue: posterior median with 95% CRI, upper bound on day 5 (23) truncated). Bottom: Detection probability per individual fritillary during each of two passes on a transect.

12.4 SUMMARY AND OUTLOOK

Abundance N is the key numerical descriptor of a central concept in ecology, the population. Since we virtually always overlook individuals, we must usually estimate N and can not directly observe it. Classical capture–recapture methods (Williams et al., 2002; and in Chapters 6 and 10), distance sampling (Buckland et al., 2001), and spatial capture–recapture methods young (Royle and Young, 2008; Borchers and Efford, 2008; Royle et al., 2011) are well developed and can be applied to data from a single site or also to data from multiple sites. However, they are costly, in the sense that extra information in the form of individual identification or accurate distance or location measurements is needed. Frequently, ecologists are interested in abundance within a metapopulation design, where the size of a collection of local populations is needed. When replicate counts are conducted over a reasonably short period of

time, the binomial mixture model (or N-mixture model) is useful for estimation of abundance based on such relatively cheap count data. This model is an extension of the Poisson models in Chapters 3 and 4 to account for imperfect detection. The binomial mixture model is a powerful model with a big scope of application in ecology and management, such as monitoring. However, we saw that it can be difficult to find models that fit the data. Furthermore, the standard Bayesian AIC-analog, DIC, should not be used for hierarchical models such as this mixture model (Millar, 2009). Hence, in the Bayesian framework, model selection can be a challenge. A somewhat ad hoc alternative might then consist in doing model selection in the frequentist framework using AIC, for example, using functions in the new R package **unmarked** (Fiske and Chandler 2011), and then fit the best model in the Bayesian framework.

Recently, an exciting generalization of the binomial mixture model to fully open metapopulation designs has been developed by Dail and Madsen (2011); see Chandler and King (2011) for an application. This model describes the openness of local populations between successive sample periods as a function of parameters for local survival and recruitment and hence, achieves two things at the same time: providing a framework of estimating abundance, corrected for detection, without any period of closure and estimating two key parameters of population dynamics from comparatively "cheap" data. This model is an important conceptual advance for attempts at making inferences about population abundance from counts of unmarked individuals. This opens up exciting possibilities for the study of spatial population dynamics. Unfortunately, the model has so far resisted to all attempts at fitting it in WinBUGS (D. Dail, R. Chandler, A. Royle, pers. comm.), but it can be fitted using maximum likelihood in the R package **unmarked** (Fiske and Chandler, 2011).

All previous applications of open-population binomial mixture models have been more or less naive in terms of the modeled biological process. That is, explicit population dynamics models such as the Ricker model for density-dependence or a dynamics description in terms of survival and recruitment processes await to be couched within the framework of binomial mixture models. Such an integration of large-scale population dynamics modeling within an estimation framework for the latent states appears to have much promise for population ecology (Buckland et al., 2007; Hooten et al., 2007; see also Pagel and Schurr, 2011). It would open up the avenue towards the study of spatial population dynamics.

Abundance is the key state variable in ecology, so when possible we would always try to model abundance rather than simply the occurrence (distribution, "presence/absence") of an organism in a metapopulation. However, there may be doubts about the validity of the closure assumption

(i.e., whether $N_{i,j}$ remains constant over replicates j). In this case, it may be adequate to reduce count data to detection/nondetection data and use another variant of a hierarchical metapopulation model called a site-occupancy model. This is the topic of the final main chapter in this book.

12.5 EXERCISES

1. With hierarchical models such as the binomial mixture model, we have several kinds of covariates: here, we have covariates that vary among sites (site covariates) and those that vary among individual surveys (sampling covariates). It is important in practice to know how to fit both kinds. Invent a sampling covariate in the example of Section 12.2.2 and fit it also to see how this works.
2. In the fritillary data, fit a simpler binomial mixture model than the one in Section 12.3.3 with detection random effects specific to day and site (i.e., drop the index j in the $\delta_{i,j,k}$). See whether that model also fits.
3. In the fritillary data, fit a more complex binomial mixture model by introducing (in addition to the random site-day-rep effect) a random site effect in the linear predictor for detection in the model in Section 12.3.3. Compare the estimates under the model in Section 12.3.3 and those in exercises 2 and 3. Explain.