CHAPTER

21

Nonstandard GLMMs 2: Binomial Mixture Model to Model Abundance

OUTLINE

21.1 INTRODUCTION

Ecology has been defined as the study of distribution and abundance (Andrewartha and Birch, 1954; Krebs, 2001). However, in nature, neither of them can usually be observed without error, and methods may need to be applied to infer the true states of distribution and abundance from imperfect observations. In Chapter 20, we met a protocol, which we called a *metapopulation design*, where the same quantity was assessed in a similar way across R sites and T temporal replicates. We saw that such a metapopulation design enables the application of site-occupancy models, a kind of nonstandard generalized linear mixed model (GLMM) with binary random effects, to estimate true species distribution free of the distorting effects of detection probability. Temporal replicate observations in a closed system allowed us to resolve the confounding between occurrence and detection.

This chapter features another nonstandard GLMM where the random effects distribution is different from normal, namely Poisson. As for the

Introduction to WinBUGS for Ecologists
DOI: 10.1016/B978-0-12-378605-0.00021-1

site-occupancy model, the random effects in this model have a precise biological meaning, which is local population size in this case. Thus, this model estimates abundance corrected for imperfect detection from temporally and spatially replicated *counts*, rather than occurrence from detection/nondetection observations as does the site-occupancy model in Chapter 20.

Our ecological motivation in this chapter is that of Dutch sand lizards (*Lacerta agilis*), one of the most widespread reptiles in large parts of Western Europe (Fig. 21.1). For more than a decade, The Netherlands have had a rather interesting reptile-monitoring program where volunteers walk transects of about 2 km length repeatedly during spring and count all reptiles they see (Kéry et al., 2009). We will generate and analyze data in the format collected in this scheme.

FIGURE 21.1 Pair of sand lizards (*Lacerta agilis*), Switzerland, 2006. (*Photo: T. Ott*)

The typical question in monitoring programs is always "are things getting better or worse?," i.e., is there a trend over time in abundance (or distribution)? The usual type of analysis to answer this question for counts used to be linear regression, but since generalized linear models (GLMs) have become fashionable, some sort of Poisson regression has become the method of choice for the analysis of count data. And, as for other ecological analyses, random effects have become popular and thus, nowadays, inference from many time-series of animal count data is often based on variants of Poisson mixed models (e.g., Link and Sauer, 2002; Ver Hoef and Jansen, 2007).

When repeated counts are available, often only the maximum count is analyzed, although this approach simply throws out valuable information. In this chapter, we make full use of replicated counts and adopt the binomial mixture model (also called the N-mixture model) of Royle (2004a) to estimate true abundance, corrected for imperfect detection (also see Dodd and Dorazio 2004; Royle, 2004b; Kéry et al., 2005; Royle et al., 2005; Dorazio, 2007; Kéry, 2008; Wenger and Freeman, 2008; Joseph et al., 2009). For simplicity, we will consider only data from a single year, and hence, we assume population closure, but the model can also be fitted to multiyear data and a trend in abundance can be estimated directly (Royle and Dorazio, 2008, p. 4–7; Kéry and Royle, 2009; Kéry et al., 2010).

First, the model: we assume that a count y_{ij} at site i and made during survey j comes from a two-stage stochastic process. The first stochastic process is the biological process that distributes the animals among the sites. This process generates the site-specific abundance that we would like to model directly but cannot because we hardly ever see all individuals. The standard statistical model for such data is the Poisson distribution, governed by the intensity (density) parameter λ, which is typically conditional on a few habitat covariates. The result of this first stochastic process is the local, site-specific abundance N_i. Given that true state N_i, the second stochastic process is the observation process which, together with N_i, determines the data actually observed, i.e., the counts y_{ij}. A natural model for the observation process in the presence of imperfect detection is the binomial distribution; given that there are N_i sand lizards present and that each has a probability of p_{ij} to be observed at site i during replicate survey j, the number of lizards actually observed is binomially distributed. Two important consequences are that (1) we typically observe fewer than N_i lizards, and (2) the counts y_{ij} will vary automatically from survey to survey even under identical conditions (Kéry and Schmidt, 2008). Three important assumptions of the binomial mixture model are that of population closure, independent and identical detection probability for all individuals at site i and during survey j, except insofar as differences among sites or surveys are modeled by covariates, and

absence of double counts and other false positive errors. The effects of violations of these assumptions are still being investigated (e.g., Joseph et al., 2009).

In summary, the binomial mixture model to estimate abundance from temporally and spatially replicated counts can be written succinctly in just two lines:

$$\begin{array}{ll} N_i \sim \text{Poisson}(\lambda) & \text{Biological process yields true state} \\ y_{ij} \sim \text{Binomial}(N_i, p_{ij}) & \text{Observation process yields observations} \end{array}$$

It is fascinating to note the similarity of this hierarchical model (Royle and Dorazio, 2006, 2008) for abundance and that for occurrence, the site-occupancy model from Chapter 20. Recognizing that in the site-occupancy model, the observation process may also be described by a binomial distribution (rather than by a Bernoulli), the sole thing that changes when we go from the modeling of occurrence to that of abundance is the distribution used to model the biological process, a Poisson instead of a Bernoulli. The binomial mixture model can be described as a binomial GLMM with a discrete (Poisson-distributed) random effect or alternatively, as a logistic regression for the count observations coupled with a Poisson regression for the imperfectly observed abundances.

Furthermore, as in the site-occupancy model, covariate effects can be modeled into the Poisson parameter λ via a log link function and into the binomial success rate p via the logit link. We can add to the model expressions such as $\log(\lambda_i) = \alpha + \beta * x_i$ and $\text{logit}(p_{ij}) = \alpha + \beta * x_{ij}$, where x_i and x_{ij} are the values of a site-covariate measured at site i (x_i) or of a survey-covariate measured at site i during survey j (x_{ij}). Of course, more than a single covariate could be fitted and the covariates for detection can be of both types.

Before we embark on our usual simulation-analysis exercise, we make two important observations on the abundance parameter of the binomial mixture model. First, even when correcting for imperfect detection (p_{ij}), the interpretation of the abundance parameter N_i is not what we might want it to be: the number of individuals that permanently reside within a well-defined plot of land. The reason for why this is not so is that animals move around, so the effective sampling area will be greater than the nominal sampling area. Hence, the estimate of N_i refers to a larger area, and we don't exactly know the size of it. The magnitude of this discrepancy depends on two things: the typical dispersal distances of the study species and the time frame of the repeated surveys. The discrepancy will be greater for greater dispersal and a longer total survey period. If we want to circumvent this difficulty, other sampling and analysis protocols must be used such as distance sampling (Buckland et al., 2001) or, more recently, spatial capture–recapture methods (Efford, 2004; Borchers and

Efford, 2008; Royle and Dorazio, 2008; Royle and Young, 2008; Efford et al., 2009; Royle et al., 2009).

Second, when animals move through the sampling area randomly, thus in effect violating the closure assumption, it appears that the estimate of N_i does not refer to the number of animals that permanently reside within the sample area, but to the number of animals that ever use an area during the entire sampling period (Joseph et al., 2009). This reasoning is analogous to the reasoning about the interpretation of the occupancy parameter in site-occupancy models in the face of temporary emigration (MacKenzie et al., 2006). In effect, it again makes the effective sampling area larger than the nominal sampling area.

These issues are not a fault of the binomial mixture model; rather, even in the absence of a formal framework for interpreting a count in the light of both the biological *and* the observation process, we never know exactly with which area the count is associated. Nor do we know by how much movement inflates our counts relative to the number of individuals that permanently reside within the area. Thus, a binomial mixture model solves the problem of imperfect detection when interpreting (i.e., analyzing) counts, but the issue of how exactly abundance should be interpreted may still remain a challenge.

21.2 DATA GENERATION

In this example of the analysis of data collected from a metapopulation design, we will choose a different format from that in Chapter 20. Instead of a rectangular or horizontal format (remember the data matrix y_{ij} in the site-occupancy analysis), we will assemble and analyze the data in a vertical format and use a population index covariate to keep track, for each count, of the population it was made in. This format is more convenient when there are many missing values, i.e., in unbalanced designs, where the number of replicate surveys is variable among sites. (It is fairly easy to formulate the site-occupancy model in this format also and the code in the current chapter may be used as a template. For an example of WinBUGS code for the binomial mixture model in horizontal format, see Kéry, 2008.) Also note the similarity of this to traditional (normal) repeated measures analysis of variance analysis, where some statistics packages allow the replicate observations to be stored in parallel columns (the horizontal format) and others prefer the observations in a single column (the vertical format) with an additional covariate that indexes "subjects." This reiterates the fact that both site-occupancy and binomial mixture models are a kind of repeated-measures analysis.

To simulate our data, we assume that we surveyed 200 sites. We choose a site covariate affecting the abundance of sand lizards and also their

detection probability. We take vegetation cover this time and assume that lizard abundance is highest at medium values: too open is bad, but too dense vegetation is also bad. We will model this as a quadratic effect of vegetation density on abundance.

```
n.site <- 200
vege <- sort(runif(n = n.site, min = -1.5, max =1.5))
```

We construct the relationship between vegetation density and abundance (Fig. 21.2).

```
alpha.lam <- 2          # Intercept
beta1.lam <- 2          # Linear effect of vegetation
beta2.lam <- -2         # Quadratic effect of vegetation
lam <- exp(alpha.lam + beta1.lam * vege + beta2.lam * (vege^2))

par(mfrow = c(2,1))
plot(vege, lam, main = "", xlab = "", ylab = "Expected abundance", las = 1)

N <- rpois(n = n.site, lambda = lam)
table(N)                # Distribution of abundances across sites
sum(N > 0) / n.site     # Empirical occupancy

> table(N)# Distribution of abundances across sites
N
```

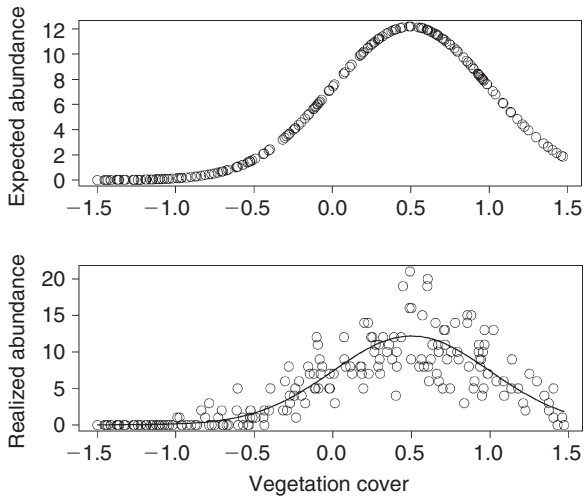

FIGURE 21.2 Expected (top) and realized relationship (bottom) between sand lizard abundance and vegetation cover. The expected abundance is shown as a black line in the bottom panel and is the same as the realized abundance minus Poisson variability.

```
 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 19 20 21
57 12 11  7  7 16  8 13 14 12  9  9  7  4  5  3  2  2  1  1

> sum(N > 0) / n.site # Empirical occupancy
[1] 0.715

plot(vege, N, main = "", xlab = "Vegetation cover", ylab = "Realized abundance")
points(vege, lam, type = "l", lwd = 2)
```

This concludes our description of the biological process: we have a random process that distributes the sand lizards across sites, and we assume that the result of this stochastic process at site i can be approximated by a conditional Poisson distribution, with rate parameter λ_i, that itself depends on vegetation density x_i in a quadratic fashion.

Next, we need to simulate the observation process, i.e., something like a "machine" that maps abundance N_i onto lizard counts y_{ij}. We assume that the observation process is also affected by vegetation density: denser vegetation reduces the detection probability (Fig. 21.3 upper panel). I note in passing that in the binomial mixture model, detection probability is defined per individual animal, whereas in the occupancy model, it refers to the probability to detect *at least one* among the N_i animals or plants present at a site. In fact, there is a precise mathematical relationship between the two kinds of detection probability, which we do not show here (see Royle and Nichols, 2003; Dorazio, 2007).

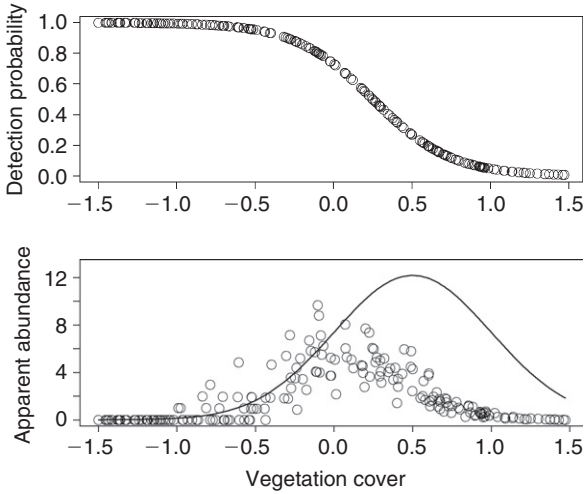

FIGURE 21.3 The relationship between vegetation cover and detection probability (top) and the expected sand lizard counts (=apparent abundance; bottom). Truth is shown as a black line in the bottom panel.

```
par(mfrow = c(2,1))
alpha.p <- 1            # Intercept
beta.p <- -4            # Linear effect of vegetation
det.prob <- exp(alpha.p + beta.p * vege) / (1 + exp(alpha.p + beta.p * vege))
plot(vege, det.prob, ylim = c(0,1), main = "", xlab = "", ylab = "Detection
probability")
```

Now for fun, let's see the expected lizard count at each site in relation to vegetation cover. The expected count at site i is given by the product of abundance N_i and detection probability at that site p_i. And let's put it all together and inspect the truth also.

```
expected.count <- N * det.prob
plot(vege, expected.count, main = "", xlab = "Vegetation cover", ylab = "Apparent
abundance", ylim = c(0, max(N)), las = 1)
points(vege, lam, type = "l", col = "black", lwd = 2) # Truth
```

A conventional analysis would use some sort of Poisson regression and model the expected count or apparent abundance. This is the bell-shaped cloud in the lower panel of Fig. 21.3, where abundance and detection are confounded. Thus, compared with the truth represented by the black line, a conventional analysis will underestimate average abundance and (in our case) estimate maximum abundance at too low of a value of vegetation cover. This is because the Poisson regression does not model abundance but rather the product of expected abundance with detection probability. This is hardly ever made explicit by authors and apparently often not even recognized.

Now let's simulate three replicate counts at each site and look at the data.

```
R <- n.site
T <- 3                  # Number of replicate counts at each site
y <- array(dim = c(R, T))

for(j in 1:T){
    y[,j] <- rbinom(n = n.site, size = N, prob = det.prob)
}
y
```

A species (occurrence) distribution is fundamentally the same as an abundance distribution, but with much reduced information: a species occurs at all sites where abundance $N > 0$ (Royle et al., 2005; Dorazio, 2007). Hence, any model of abundance is also a model of species distribution. It is seldom useful to think of distribution as something separate from abundance.

```
sum(apply(y, 1, sum) > 0)       # Apparent distribution (proportion occupied sites)
sum(N > 0)                      # True occupancy

> sum(apply(y, 1, sum) > 0)
[1] 126
> sum(N > 0)
[1] 143
```

Now stack the replicated counts on top of each other for a vertical data format (i.e., convert the matrix to a vector)

```
C <- c(y)
```

We also need a site index and a vegetation covariate that have the same length as the variable `C` (for the observation model, i.e., to model *p*; see WinBUGS code in section 21.3.). We will denote them by a p suffix in the variable name.

```
site = 1:R                      # 'Short' version of site covariate
site.p <- rep(site, T)          # 'Long' version of site covariate
vege.p <- rep(vege, T)          # 'Long' version of vegetation covariate
cbind(C, site.p, vege.p)        # Check that all went right
```

Here is a quick and dirty conventional analysis of the maximum counts assuming a Poisson distribution for the max(count) (A slightly better alternative would, perhaps, be to assume a normal distribution for the mean of the counts.):

```
max.count <- apply(y, 1, max)
naive.analysis <- glm(max.count ~ vege + I(vege^2), family = poisson)
summary(naive.analysis)
lin.pred <- naive.analysis$coefficients[1] + naive.analysis$coefficients[2] *
vege + naive.analysis$coefficients[3] * (vege*vege)
```

We compare truth and the naïve analysis in a graph (Fig. 21.4):

```
par(mfrow = c(1,1))
plot(vege, max.count, main = "", xlab = "Vegetation cover", ylab = "Abundance or
count", ylim = c(0,max(N)), las = 1)
points(vege, lam, type = "l", col = "black", lwd = 2)
points(vege, exp(lin.pred), type = "l", col = "red", lwd = 2)
```

Clearly, the predictions under the naïve analysis yield a biased picture of the relationship between abundance and vegetation cover because sand lizards are easier to see in more open vegetation.

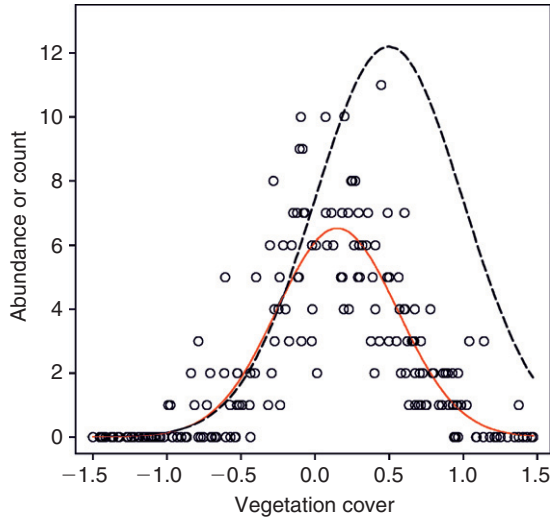

FIGURE 21.4 Relationship between abundance of Dutch sand lizards and vegetation cover as inferred by a naïve analysis not accounting for detection probability. Circles show the maximum count at each site and the black solid line the predicted relationship under the naïve model. Truth is a dashed line.

21.3 ANALYSIS USING WinBUGS

Now let's see how the binomial mixture model can do better. As for the site-occupancy model, a variety of binomial mixture models can be fitted using maximum likelihood in the free Windows programs MARK and PRESENCE. R code for obtaining maximum likelihood estimates under the model can be found in Kéry et al. (2005), Royle and Dorazio (2008) and its Web appendix, and Wenger and Freeman (2008). Furthermore, the model can be fitted using functions in the new R package unmarked (Fiske and Chandler, 2010). We show a Bayesian solution here.

```
# Define model
sink("BinMix.txt")
cat("
model {

# Priors
 alpha.lam ~ dnorm(0, 0.1)
 beta1.lam ~ dnorm(0, 0.1)
 beta2.lam ~ dnorm(0, 0.1)
 alpha.p ~ dnorm(0, 0.1)
 beta.p ~ dnorm(0, 0.1)
```

```
# Likelihood
# Biological model for true abundance
 for (i in 1:R) {               # Loop over R sites
   N[i] ~ dpois(lambda[i])
   log(lambda[i]) <- alpha.lam + beta1.lam * vege[i] + beta2.lam * vege2[i]
 }

# Observation model for replicated counts
 for (i in 1:n) {               # Loop over all n observations
      C[i] ~ dbin(p[i], N[site.p[i]])
      logit(p[i]) <- alpha.p + beta.p * vege.p[i]
 }

# Derived quantities
 totalN <- sum(N[])        # Estimate total population size across all sites
}
",fill=TRUE)
sink()

# Bundle data
R = dim(y)[1]
n = dim(y)[1] * dim(y)[2]
vege2 = (vege * vege)
win.data <- list(R = R, vege = vege, vege2 = vege2, n = n, C = C, site.p =
site.p, vege.p = vege.p)
```

As for the site-occupancy model, clever starting values for the latent states (the N_i's) are essential. We use the maximum count at each site as a first guess of what N might be and add 1 to avoid zeros. WinBUGS cannot use zeroes for the N value for the binomial and will crash if you initialize the model with zeroes.

```
# Inits function
Nst <- apply(y, 1, max) + 1
inits <- function(){list(N = Nst, alpha.lam=rnorm(1, 0, 1), beta1.lam=rnorm(1, 0,
1), beta2.lam=rnorm(1, 0, 1), alpha.p=rnorm(1, 0, 1), beta.p=rnorm(1, 0, 1))}

# Parameters to estimate
params <- c("N", "totalN", "alpha.lam", "beta1.lam", "beta2.lam", "alpha.p",
"beta.p")

# MCMC settings
nc <- 3
nb <- 200
ni <- 1200
nt <- 5
```

```
# Start Gibbs sampler
out <- bugs(win.data, inits, params, "BinMix.txt", n.chains=nc, n.iter=ni, n.burn
= nb, n.thin=nt, debug = TRUE)
```

Here is a first note on the practical implementation of such slightly more complex models in WinBUGS: This code works fine and appears to converge surprisingly quickly for our data set. But as an illustration of how "difficult" WinBUGS can sometimes be, try widening the range of the priors by increasing some or all of the precisions from 0.1 to 0.01: WinBUGS will crash. Many modeling choices that are not wrong, but simply not chosen in an optimal manner, can throw you off the track in your attempts to exploit the great modeling freedom that WinBUGS gives you in principle. It is true that with experience, the reasons for many crashes can be diagnosed, but for a beginner, they may represent major stumbling blocks.

Let's now compare the known true values with what the analysis has recovered.

```
cat("\n *** Our estimate of truth *** \n\n")
print(out, dig = 2)

cat("\n *** Compare with known truth ***\n\n")
alpha.lam    ;    beta1.lam    ;    beta2.lam    ;    alpha.p    ;    beta.p
sum(N)          # True total population size across all sites
sum(apply(y, 1, max))        # Sum of site max counts

> cat("\n *** Our estimate of truth *** \n\n")

 *** Our estimate of truth ***

> print(out, dig = 2)
Inference for Bugs model at "BinMix.txt", fit using WinBUGS,
 3 chains, each with 1200 iterations (first 200 discarded), n.thin = 5
 n.sims = 600 iterations saved
               mean      sd    2.5%     25%     50%     75%   97.5%  Rha n.eff
[ ... ]
totalN       957.01  160.99  683.95  836.00  948.00 1066.00 1282.22 1.01   310
alpha.lam      2.47    0.18    2.11    2.35    2.48    2.59    2.80 1.00   340
beta1.lam      0.71    0.23    0.27    0.54    0.72    0.85    1.17 1.01   480
beta2.lam     -2.81    0.23   -3.25   -2.97   -2.80   -2.65   -2.33 1.02   150
alpha.p        0.14    3.16   -5.80   -2.10   -0.06    2.57    6.32 1.00   600
beta.p        -0.14    2.95   -5.95   -2.15   -0.17    1.96    5.76 1.00   600
deviance    5844.93 4799.46 1385.70 2547.73 4475.00 7278.66 18910.75 1.00   600
[ ... ]
>
> cat("\n *** Compare with known truth ***\n\n")
```

```
 *** Compare with known truth ***
> alpha.lam    ;    beta1.lam    ;    beta2.lam    ;    alpha.p    ;    beta.p
[1] 2
[1] 2
[1] -2
[1] 1
[1] -4
> sum(N) # True total population size across all sites
[1] 1073
> sum(apply(y, 1, max))# Sum of site max counts
[1] 481
```

With truth being 1073 sand lizards, the estimate of total N (957 lizards) *appears* decent with respect to the sum of the max counts across all 200 sites, which was only 481. However, there is a slight correspondence for the coefficients in the biological process model (`alpha.lam`, `beta1.lam`, `beta2.lam`), but no similarity at all for the coefficients in the observation process model … That is disappointing! What has happened? Note again that WinBUGS claims that the chains have converged.

Seeing these results for the first time, I couldn't believe that this would go wrong because I knew that these parameters should all be identifiable (and were so for an only slightly different model in Kéry, 2008). I tried out several things to see whether I could get a better result; I increased sample sizes (e.g., to `R = 2000` and `T = 10`), dropped the quadratic term in the biological process model, but none of those helped. In the end, I tried out an alternative set of uniform priors instead of the fairly uninformative normals used previously. I also avoided the WinBUGS logit and defined that function myself (see WinBUGS tricks in the Web appendix). And this worked!

That is, I fitted the following version of the model, where I also added code to assess model goodness of fit using a posterior predictive check for a Chi-square discrepancy measure. Convergence in a binomial mixture model is attained notoriously much more slowly than, say, in a site-occupancy model. Therefore, I ran considerably longer chains.

```
# Define model with new uniform priors
sink("BinMix.txt")
cat("
model {

# Priors (new)
 alpha.lam ~ dunif(-10, 10)
 beta1.lam ~ dunif(-10, 10)
 beta2.lam ~ dunif(-10, 10)
 alpha.p ~ dunif(-10, 10)
 beta.p ~ dunif(-10, 10)
```

```
# Likelihood
# Biological model for true abundance
 for (i in 1:R) {
   N[i] ~ dpois(lambda[i])
   log(lambda[i]) <- alpha.lam + beta1.lam * vege[i] + beta2.lam * vege2[i]
 }

# Observation model for replicated counts
 for (i in 1:n) {
   C[i] ~ dbin(p[i], N[site.p[i]])
   lp[i] <- alpha.p + beta.p * vege.p[i]
   p[i] <- exp(lp[i])/(1+exp(lp[i]))
 }

# Derived quantities
 totalN <- sum(N[])

# Assess model fit using Chisquare discrepancy
 for (i in 1:n) {

# Compute fit statistic for observed data
     eval[i]<-p[i]*N[site.p[i]]
     E[i] <- pow((C[i] - eval[i]),2) / (eval[i] + 0.5)

# Generate replicate data and compute fit stats for them
     C.new[i] ~ dbin(p[i], N[site.p[i]])
     E.new[i] <- pow((C.new[i] - eval[i]),2) / (eval[i] + 0.5)
 }
 fit <- sum(E[])
 fit.new <- sum(E.new[])
}
",fill=TRUE)
sink()

# Parameters to estimate
params <- c("N", "totalN", "alpha.lam", "beta1.lam", "beta2.lam", "alpha.p",
"beta.p", "fit", "fit.new")

# MCMC settings
nc <- 3
nb <- 10000
ni <- 60000
nt <- 50                    # Takes about 20 mins on my laptop

# Start Gibbs sampler
out <- bugs(win.data, inits, params, "BinMix.txt", n.chains=nc, n.iter=ni, n.burn
= nb, n.thin=nt, debug = TRUE)
```

Because convergence in Bayesian analyses of binomial mixture models may be hard to achieve, we first check whether this run has converged. We find that the specifications of this Markov chain Monte Carlo (MCMC) run seem to have been long enough. The Markov chains of all primary, structural model parameters seem to have converged, only those for some latent (local abundance N_i) parameters have not converged. Their convergence is of lesser concern, unless of course the chains for local abundance of your favorite population happened not to converge.

```
print(out, dig = 3)
which(out$summary[,8] > 1.1)
```

Next, we check whether the uniform prior on `beta.p` was not too restrictive by producing a histogram of the posterior (the posterior of this parameter seemed to come closest to the bounds of the uniform prior). However, there is no indication for any mass being piled up toward one of the bounds (Fig. 21.5), and therefore, we conclude that there was no undue influence of this prior on our inference.

```
hist(out$sims.list$beta.p, col = "grey", main = "", xlab = "")
```

Next, we check the adequacy of the model for the data set first using a posterior predictive check, before inspecting the parameter estimates (Fig. 21.6):

```
plot(out$sims.list$fit, out$sims.list$fit.new, main = "", xlab = "Discrepancy
measure for actual data set", ylab = "Discrepancy measure for perfect data sets")
abline(0,1, lwd = 2, col = "black")
```

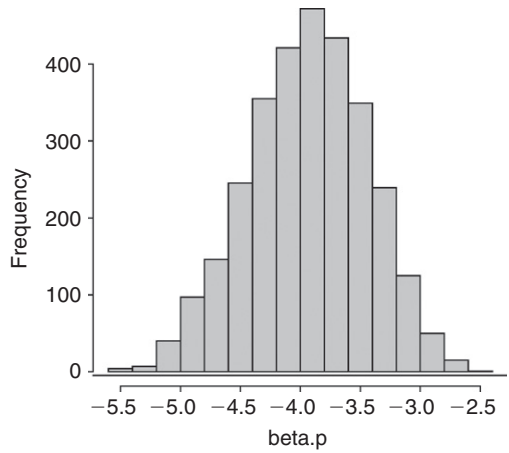

FIGURE 21.5 Posterior distribution of the slope estimate of sand lizard detection probability on vegetation cover (`beta.p`).

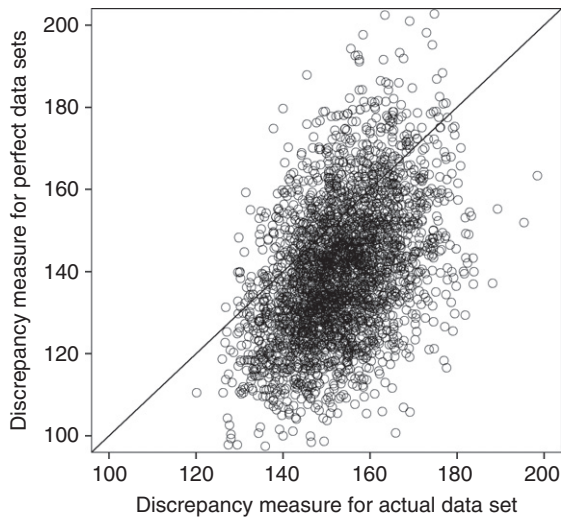

FIGURE 21.6 Posterior predictive check of the binomial mixture model using a Chi-square discrepancy.

```
mean(out$sims.list$fit.new > out$sims.list$fit)
> mean(out$sims.list$fit.new > out$sims.list$fit)
[1] 0.1906667
```

Both the graphical check and the Bayesian *p*-value, indicate an adequate model for our data set, so we inspect the parameter estimates and compare them with truth in the data-generating process.

```
cat("\n *** Our estimate of truth *** \n\n")
print(out, dig = 2)

cat("\n *** Compare with known truth ***\n\n")
alpha.lam    ;    beta1.lam    ;    beta2.lam    ;    alpha.p    ;    beta.p
sum(N)           # True total population size across all sites
sum(apply(y, 1, max))        # Sum of site max counts

> cat("\n *** Our estimate of truth *** \n\n")

 *** Our estimate of truth ***

> print(out, dig = 2)
Inference for Bugs model at "BinMix.txt", fit using WinBUGS,
 3 chains, each with 60000 iterations (first 10000 discarded), n.thin = 50
 n.sims = 3000 iterations saved
               mean      sd     2.5%      25%      50%      75%    97.5% Rhat  n.eff
N[1]           0.00    0.00     0.00     0.00     0.00     0.00     0.00 1.00      1
[ ... ]
```

```
N[200]          2.26    3.07    0.00    0.00    1.00    3.00    11.00 1.03    290
totalN       1014.12  268.27  683.00  822.00  942.00 1136.25 1741.17 1.02    160
alpha.lam       1.96    0.07    1.82    1.91    1.96    2.00     2.10 1.00   1300
beta1.lam       1.81    0.26    1.34    1.62    1.79    1.99     2.36 1.01    190
beta2.lam      -1.99    0.33   -2.64   -2.22   -1.99   -1.76    -1.35 1.01    280
alpha.p         1.12    0.17    0.78    1.02    1.13    1.24     1.44 1.01    310
beta.p         -3.94    0.49   -4.94   -4.28   -3.92   -3.59    -3.02 1.01    260
fit           153.07   10.64  133.10  145.70  152.70  160.10   175.10 1.00    550
fit.new       139.98   16.73  110.60  128.00  139.10  150.60   176.20 1.01    290
[ ... ]

DIC info (using the rule, pD = var(deviance)/2)
pD = 209.2 and DIC = 1161.9
DIC is an estimate of expected predictive error (lower deviance is better).
>
> cat("\n *** Compare with known truth ***\n\n")

 *** Compare with known truth ***

> alpha.lam    ;    beta1.lam    ;    beta2.lam    ;    alpha.p    ;    beta.p
[1] 2
[1] 2
[1] -2
[1] 1
[1] -4

> sum(N) # True total population size across all sites
[1] 1073

> sum(apply(y, 1, max))# Sum of site max counts
[1] 481
```

Our model recovered adequate parameter estimates for the covariate relationships and estimated a total population size across all 200 sites of 1014 sand lizards (95% CI: 683–1741). As a comparison, truth was 1073 lizards, and the sum of the max counts, a conventional estimate of total population size, was only 481.

Figure 21.7 gives a graphical comparison between the parameter estimates under the binomial mixture model and the values of the associated data-generating parameters. It shows again that the model does a good job at estimating abundance.

```
par(mfrow = c(3,2))
hist(out$sims.list$alpha.lam, col = "grey", main = "alpha.lam", xlab = "")
abline(v = alpha.lam, lwd = 3, col = "black")
hist(out$sims.list$beta1.lam, col = "grey", main = "beta1.lam", xlab = "")
```

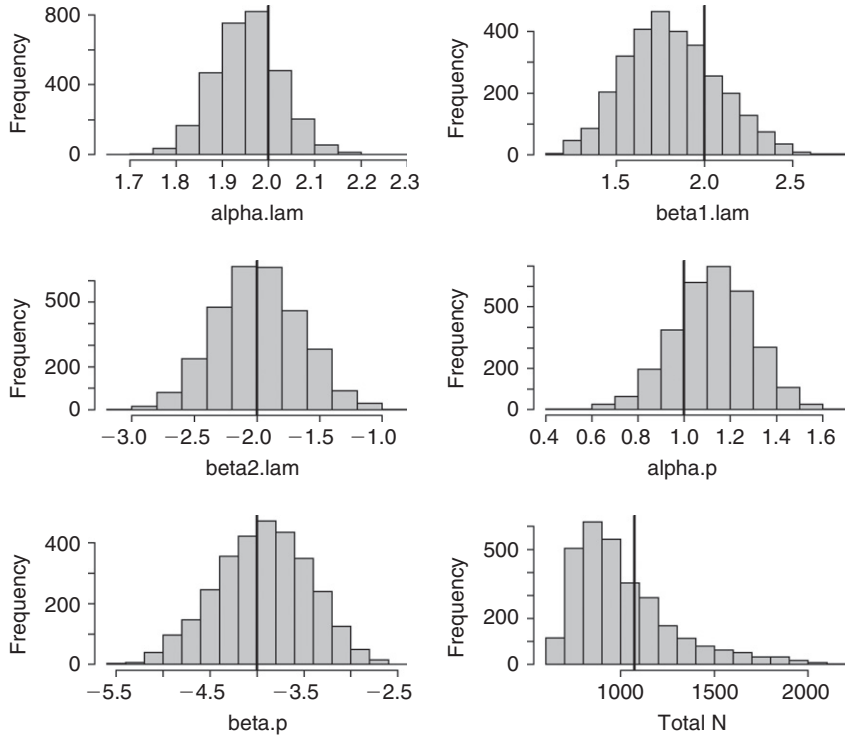

FIGURE 21.7 Comparison of estimates under the binomial mixture model (posterior distributions) and truth in the data-generating algorithm (black line) for six estimands.

```
abline(v = beta1.lam, lwd = 3, col = "black")
hist(out$sims.list$beta2.lam, col = "grey", main = "beta2.lam", xlab = "")
abline(v = beta2.lam, lwd = 3, col = "black")
hist(out$sims.list$alpha.p, col = "grey", main = "alpha.p", xlab = "")
abline(v = alpha.p, lwd = 3, col = "black")
hist(out$sims.list$beta.p, col = "grey", main = "beta.p", xlab = "")
abline(v = beta.p, lwd = 3, col = "black")
hist(out$sims.list$totalN, col = "grey", , main = "Total N", xlab = "")
abline(v = sum(N), lwd = 3, col = "black")
```

Here is a second note on the practical implementation of such slightly more complex models in WinBUGS: We saw that a relatively slight change (here, in the priors) had a very large effect on the inference. This could also be called an example of prior sensitivity of the inference. It is definitely a good idea to check the Bayesian analysis of more complex models by exploring "neighboring model regions." Vary the likelihood (e.g., the covariates that are in or not), priors, model parameterization or other things slightly and see whether your inference is robust.

In our case, we now get rather decent estimates fairly close to the known truth. Hence, we illustrate a few further inferences that can be made under the model by looking at a few further posterior distributions. In Fig. 21.7, we have seen those for some primary parameters of the model. One of the most interesting things in the binomial mixture model is that site-specific estimates, N_i, can be obtained. Let's now have a look at these estimates of local abundance for a random sample of sites (Fig. 21.8).

```
sel <- sort(sample(1:200, size = 4))
sel

par(mfrow = c(2,2))
hist(out$sims.list$N[,sel[1]], col = "grey", xlim = c(Nst[sel[1]]-1,
max(out$sims.list$N[,sel[1]])), main = "Site 48", xlab = "")
abline(v = Nst[sel[1]]-1, lwd = 3, col = "red")
```

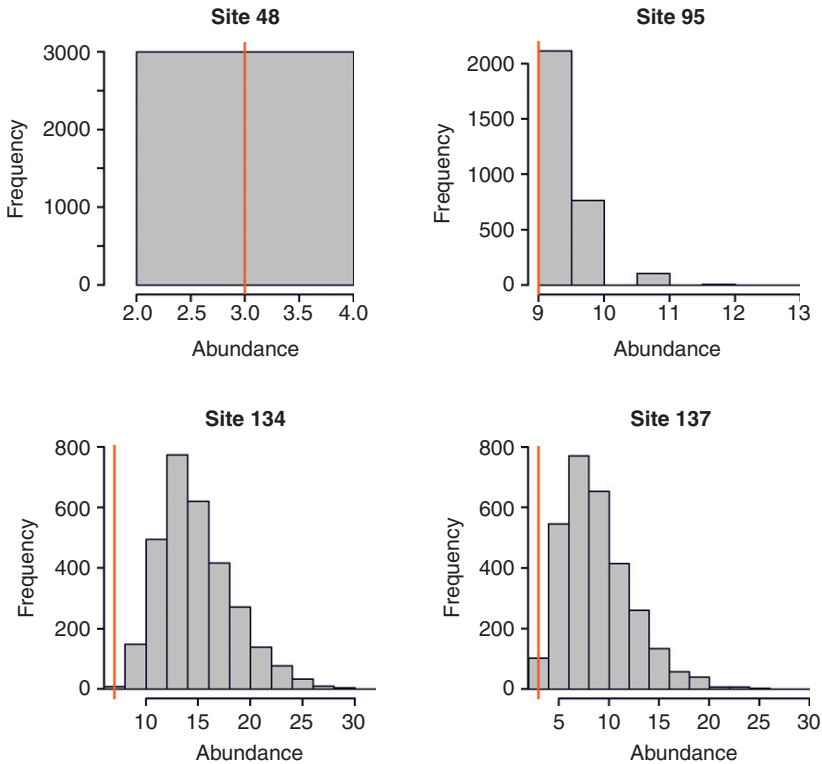

FIGURE 21.8 Comparison of estimates of local abundance (N_i) under the binomial mixture model (posterior distributions) and the maximum count (black line) at a sample of four sites.

```
hist(out$sims.list$N[,sel[2]], col = "grey", xlim = c(Nst[sel[2]]-1,
max(out$sims.list$N[,sel[2]])), main = "Site 95", xlab = "")
abline(v = Nst[sel[2]]-1, lwd = 3, col = "red")

hist(out$sims.list$N[,sel[3]], col = "grey", xlim = c(Nst[sel[3]]-1,
max(out$sims.list$N[,sel[3]])), main = "Site 134", xlab = "")
abline(v = Nst[sel[3]]-1, lwd = 3, col = "red")

hist(out$sims.list$N[,sel[4]], col = "grey", xlim = c(Nst[sel[4]]-1,
max(out$sims.list$N[,sel[4]])) , main = "Site 137", xlab = "")
abline(v = Nst[sel[4]]-1, lwd = 3, col = "red")

> sel
[1]     48    95    134    137
```

The posterior distributions show the likely size of the local populations (N_i) of sand lizards at sites number 48, 95, 134, and 137. We can compare this to the observed data at these sites:

```
y[sel,]
> y[sel,]
     [,1] [,2] [,3]
[1,]    3    3    3
[2,]    6    7    9
[3,]    7    7    3
[4,]    1    3    2
```

And since we know truth, why not have a look at it? Here are the true population sizes at these sites:

```
N[sel]
> N[sel]
[1]  3  9  21  8
```

Compare this with the estimates of these local N_i:

```
print(out$mean$N[sel], dig = 3)
> print(out$mean$N[sel], dig = 3)
[1]  3.00  9.34  15.31  9.31
```

Finally, Fig. 21.9 shows a comparison of the relationship between sand lizard abundance and vegetation cover using a naïve analysis and under the binomial mixture model.

```
par(mfrow = c(1,1))
plot(vege, N, main = "", xlab = "Vegetation cover", ylab = "Abundance", las = 1,
ylim = c(0,max(N)))
```

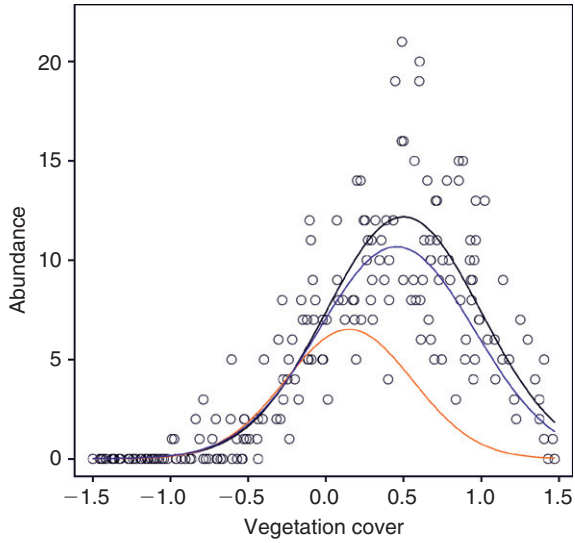

FIGURE 21.9 Comparison of the estimated abundance–vegetation relationship in Dutch sand lizards under a naïve approach that ignores imperfect detection (red) with that under the binomial mixture model (blue). Truth is shown in black: circles are the realized abundances at each site (N_i) and the black line is their expectation (as in Fig. 21.1).

```
points(sort(vege), lam[order(vege)], type = "l", col = "black", lwd = 2)
points(vege, exp(lin.pred), type = "l", col = "red", lwd = 2)
BinMix.pred <- exp(out$mean$alpha.lam + out$mean$beta1.lam * vege +
out$mean$beta2.lam * (vege*vege))
points(vege, BinMix.pred, type = "l", col = "blue", lwd = 2)
```

21.4 SUMMARY

Binomial mixture modeling in metapopulation designs offers great opportunities for the estimation of animal or plant abundance corrected for imperfect detection probability (p). Essentially, the model is simply a generalized version of the familiar Poisson regression model that accommodates imperfect detection; when $p = 1$, we are back to a classical Poisson generalized linear model. However, it is more complex than a simpler Poisson regression and even more so than a conventional hierarchical Poisson regression or GLMM (see Chapter 16). There are many ways in which the practical implementation in WinBUGS may fail, and a lot of trial and error and model checking may be required. Nevertheless, if replicate count data are available from a number of sites, you should absolutely try out this new and exciting model.

EXERCISES

1. *Survey covariates*: In metapopulation designs, we frequently have detection-relevant covariates that vary by site *and* survey (i.e., survey covariates). For our Dutch sand lizards, one such covariate is ambient temperature (Kéry et al., 2009): presumably, lizard activity depends on the temperature and this affects their detection probability. Modify the data generation code as well as the WinBUGS model to include the effects of a temperature covariate.
2. *Prior sensitivity*: Play around with prior settings in the last model we ran. Change the uniform distributions to have a very wide range and see whether the model converges. Conversely, set the range very narrow and see whether the inference is affected, i.e., whether the parameter estimates are changed.
3. *Swiss hare data*: Fit the binomial mixture model to the hare data from a single year (e.g., 2000) and see whether there is a difference in the probability of detection in grassland and arable areas. By what proportion will mean or max counts underestimate true population size?
4. *Simulation exercise*: The binomial mixture model was described in 2004 (Royle, 2004a) and so is still fairly young. Five years later, it had been applied in hardly more than 10 publications, and much remains to be found out about the model that can be tackled by simulation studies. For instance, doing a simulation study (vary R, T, covariate effects, and other things) for models with covariates similar as what was suggested for the site-occupancy model (Exercise 4 in Chapter 20) might be worthwhile. This is another serious study that could easily yield a decent article or thesis chapter.