CHAPTER

HIERARCHICAL MODELS FOR COMMUNITIES

11

CHAPTER OUTLINE

11.1 INTRODUCTION

In this chapter, we deal with the third main element in the subtitle of this book, species richness. But we do much more: we present a type of hierarchical model (HM) for communities, or more specifically for metacommunities, that was independently developed by Dorazio and Royle (2005) and by Gelfand et al. (2005). Of course, there is a vast body of models and methods for community analysis, and we won't even scratch the surface of community analysis. What we will do is showcase a single but extremely powerful approach for modeling metacommunities: the community occupancy model

Applied Hierarchical Modeling in Ecology. http://dx.doi.org/10.1016/B978-0-12-801378-6.00011-4

(Dorazio and Royle, 2005) and the community abundance or community *N*-mixture model (Yamaura et al., 2012; Chandler et al., 2013). We call them the Dorazio/Royle (DR) and the Dorazio/Royle/Yamaura (DRY) models, respectively. Both are based on the eminently sensible idea of describing a metacommunity as a collection of individual species. In this framework, plenty of additional complexity can easily be built in as needed, or as the data—or your patience—allow, when fitting the models. Most importantly, we can accommodate environmental and other effects on distribution or abundance of the individual species, false-negative measurement error, temporal dynamics of species-level occurrence and abundance, and species interactions. We cover the former two in this chapter extensively and discuss interactions, but will cover temporal dynamics and interactions formally only in volume 2. We remind you that for the modeling of a single community, i.e., for a collection of species at a single place, you can use a static occupancy model where individual species take the place of a 'site'; see Section 10.13. In contrast, Chapter 11 is about the modeling of several or many such communities.

The DR/DRY community models enable you to estimate and model species richness at both the community and the whole metacommunity level (Brown and Maurer, 1989; Holyoak et al., 2005). A community is an ensemble of species occurring at one site, and a metacommunity is therefore a collection of such communities. In keeping with one of the main themes of this book, which is the analysis of ecological data on distribution and abundance that are *site-structured*, we strictly deal with metacommunities only. However, we quite often use the terms community and metacommunity exchangeably when the meaning should be clear. (Another slight vagueness in this chapter is the use of N to denote the number of species, although sometimes N will be used for local abundance of individuals, as throughout the book. Again, the meaning of N should be clear from the context.)

In a sense, the DR/DRY community models represent the culmination of all the other models for occurrence and abundance in this book: the DR is simply a community site-occupancy model, and the DRY is a community *N*-mixture model. While the basic HMs in Chapters 6−10 have two levels, one for the true but latent ecological state and another for the measurement error, the basic DR/DRY community models have three levels; we add a third level on top that describes the sampling of each species from the metacommunity. Thus, the community model is a "hypermodel" for abundance or distribution for a collection of species. The parameters for each species are treated as random effects endowed with prior distributions, and the hyperparameters of those priors describe the community.

DR/DRY community models have several key advantages. First, they permit inference at the level of the whole community and at that of each individual species, and second, our usual binomial measurement error model ideally eliminates (or at least reduces) the bias in our inferences due to false-negative measurement errors. In these models, we will encounter data augmentation (DA) in a novel setting that allows us not to estimate how many *individuals* we miss in a population (as in Chapters 7−9), but instead now DA allows us to make a formal guess about how many *individual species* we miss in a metacommunity. In that way, both community models permit inference at three hierarchical levels: metacommunity, community, and individual species. This, combined with the explicit description of the ubiquitous false-negative measurement error, is a unique feature of this modeling framework. We will show how inferences about alpha (local scale), beta (landscape scale), and gamma (macroscale) diversity are naturally made as a function of the estimated presence/absence matrix, where *estimation* means that we have corrected for false-negative measurement errors. Many important generalizations of these models, including the dependence of occurrence or abundance among species (i.e., species interactions), are fairly straightforward, at least conceptually.

Ten years after its initial development, the DR community occupancy model (but not the DRY community abundance model; see Section 11.10) has been applied quite widely and in an increasing

number of papers, including the following: Kéry and Royle (2008), Kéry et al. (2008), Kéry and Royle (2009), Russell et al. (2009), Zipkin et al. (2009, 2010, 2012), DeWan and Zipkin (2010), Ruiz-Gutierrez et al. (2010), Holtrop et al. (2010), Kéry (2011a), Ruiz-Gutierrez and Zipkin (2011), Wells et al. (2011), Burton et al. (2012), Dorazio (2012), Jones et al. (2012), Chen et al. (2013), Giovanini et al. (2013), Guzy et al. (2013), Henden et al. (2013), Holt et al. (2013), Hunt et al. (2013), Linden and Roloff (2013), Mattsson et al. (2013), Sauer et al. (2013), Tingley and Beissinger (2013), White et al. (2013a,b), Carrillo-Rubio et al. (2014), Gilroy et al. (2014a,b, 2015), Homyack et al. (2014), Iknayan et al. (2014), Karanth et al. (2014), Kroll et al. (2014), Mata et al. (2014), McManamay et al. (2014), Pacifici et al. (2014), Sanderlin et al. (2014), Higa et al. (2015), Lewis et al. (2015), McNew and Handel (2015), Mihaljevic et al. (2015), Russell et al. (2015) and Sutherland et al. (in review).

Other authors have independently developed community models that are based on the notion of collecting together component models for individual species. Notably, first of all, the work by Gelfand and colleagues (e.g., Gelfand et al., 2005, 2006; Latimer et al., 2006) has important conceptual similarities to the models described in this chapter, though they lack an explicit description of presence/absence measurement error and thus cannot make explicit inferences about the community or estimates of species richness at any scale. Next, the principle of building community or joint species distribution models (JSDMs) from individual-species models is being reinvented quite regularly. Recent instances include Ovaskainen et al. (2010), Ovaskainen and Soininen (2011), Clark et al. (2014), and Pollock et al. (2014). However, we note that these authors all add interspecific dependency of occurrence in their models, which the basic DR/DRY models in this chapter do not contain.

The plan of this chapter is as follows: we use two broad strategies for describing the DR and the DRY models. The first is the simulation of a data set under these models, as we usually do at the start of a chapter. We provide a data simulation function for metacommunity presence/absence and count data, which hopefully provides you with deeper insight into some fundamentals of communities, such as the relationships between features of the individual species and emerging properties of the community, and the relationship between features of the observation process and the observed data. As always, when you really understand the simulation code for a data set under a specific model, you are well prepared for understanding the model as well. In addition, we hope that the simulation function will help you conduct simulation studies for investigation of topics about this type of model, or about communities in general, that are important for your work (for a recent example of such a simulation study see McNew and Handel, 2015). The second main feature of the chapter is that we approach the DR/DRY models by starting at a very basic and not even hierarchical model for a community, and then progressively change that until we are at the community model we desire. We hope that this *model progression* will help you understand our main hierarchical community models and clarify the ways in which they are different from other approaches.

11.2 SIMULATION OF A METACOMMUNITY

We use data simulation to clarify our thinking about two things. First, we emphasize how a metacommunity can be thought of as the result of a superposition of presence/absence or abundance patterns over a number of sites for a whole collection of species. Patterns at community (one site) or metacommunity (multiple sites) levels emerge as a function of community mean and among-species variance of species-specific traits that affect both the occupancy and the detection probability of all

occurring species. Second, you will not be surprised that we emphasize the observation process underlying all real-world community data. Therefore, we explicitly describe two processes that underlie an observed presence/absence or abundance pattern in a metacommunity: the first is the *ecological process* governing the true occurrence or abundance, and the second is the *measurement process* (we allow for various patterns in false-negative detection errors).

Our function `simComm` permits you to simulate metacommunity presence/absence or count data that are constructed by combining these two processes. Beyond any association created by a similar response to covariates, the function assumes that species occur and are detected independently from one another. The default function arguments are the following:

```
simComm(type="det/nondet", nsite=30, nrep=3, nspec=100,
mean.psi=0.25, sig.lpsi=1, mu.beta.lpsi=0, sig.beta.lpsi=0,
mean.lambda=2, sig.loglam=1, mu.beta.loglam=1, sig.beta.loglam=1,
mean.p=0.25, sig.lp=1, mu.beta.lp=0, sig.beta.lp=0, show.plot = TRUE)
```

Function `simComm` permits generation of replicated presence/absence (for `type="det/nondet"`) or abundance data (for `type="counts"`) with possible false-negative measurement error for a specified number of sites (`nsite`), replicates (or occasions, `nrep`), and size of a regional species pool (`nspec`) under the following models for occupancy or abundance, and detection for site i, replicate j, and species k. We generate the true presence/absence (z_{ik}) or abundance (N_{ik}) of species k ($k = 1 \ldots$nspec) at site i ($i = 1 \ldots$nsite) as follows. For now, we focus on the default function setting `type = "det/nondet"`—i.e., on the simulation of presence/absence measurements in a metacommunity.

Here is the model for presence/absence and occupancy probability for species k at site i:

$z_{ik} \sim Bernoulli(\psi_{ik})$ # True presence/absence
$\text{logit}(\psi_{ik}) = beta0_k + beta1_k * habitat_i$ # Occupancy probability affected by habitat
$beta0_k \sim Normal(\text{logit}(mean.psi), sig.lpsi^2)$ # Species heterogeneity in the intercept
$beta1_k \sim Normal(mu.beta.lpsi, sig.beta.lpsi^2)$ # Species heterogeneity in the slope

Hence, presence/absence z_{ik} of species k at site i is simulated as a Bernoulli trial with occupancy probability ψ_{ik}. A single site covariate, *habitat*, may affect the occupancy probability of a species via a logit-linear model. Both the intercept (the mean occupancy probability at a *habitat* value of zero) and the slope of the relationship are indexed by k; hence, they can differ by species. The average community response is $\text{logit}(\psi_i) = \text{logit}(mean.psi) + mu.beta.lpsi * habitat_i$, and the magnitude of the among-species variability around that community mean is governed by the standard deviations of the two normal distributions for the intercepts and the slopes. When a standard deviation is set to zero, the metacommunity becomes homogeneous in terms of that parameter. In contrast, when you set the standard deviation to a nonzero value, individual species will differ in their intercepts or their responses to the habitat. Depending on the magnitude of the variability among species, slopes may easily have opposing signs among the species in the community (see below).

All species occur and are detected independently; i.e., after accounting for covariate effects, the presence or the detection of one species does not have any effect on the likelihood of presence or detection of another species. If you want to relax that assumption, you could modify the function such that, for instance, $beta0_k$ is a draw from a multivariate normal distribution with a specified variance-covariance matrix that describes the residual associations among species. Nonzero off-diagonal

elements in that matrix specify positive or negative residual association of the species; see Ovaskainen et al. (2010), Ovaskainen and Soininen (2011), and Dorazio and Connor (2014) for examples, as well as Chapter 20 in volume 2.

We generate the observed presence/absence measurements, or detection/nondetection data, (y_{ijk}) of species k at site i during occasion j as follows.

$$
\begin{aligned}
&y_{ijk} \sim Bernoulli(z_{ik} * p_{ijk}) && \text{\# Observed detection/nondetection data}\\
&\text{logit}(p_{ijk}) = alpha0_k + alpha1_k * wind_{ij} && \text{\# Detection probability affected by } wind\\
&alpha0_k \sim Normal(\text{logit}(mean.p), sig.lp^2) && \text{\# Species heterogeneity in the intercept}\\
&alpha1_k \sim Normal(mu.beta.lp, sig.beta.lp^2) && \text{\# Species heterogeneity in the slope}
\end{aligned}
$$

Thus, the observed data y_{ijk} are simulated as a Bernoulli trial with a success probability that is the product of the presence/absence of species k at site i (z_{ik}) and detection probability p_{ijk} for species k during occasion j at the site i. Our method of data simulation excludes false-positive errors. Detection probability on the logit scale may be affected by an observation covariate, *wind* speed, and the intercepts and slopes of this relationship for each species (on the logit scale) may again vary according to two independent normal distributions with means and standard deviations that describe the community mean response and the among-species variability in that response, respectively.

With `type="counts"`, abundance data from a metacommunity are simulated under an analogous model:

$$
\begin{aligned}
&N_{ik} \sim Poisson(\lambda_{ik}) && \text{\# True abundance}\\
&\log(\lambda_{ik}) = beta0_k + beta1_k * habitat_i && \text{\# Expected abundance affected by } habitat\\
&y_{ijk} \sim Binomial(N_{ik}, p_{ijk}) && \text{\# Observed count data}\\
&\text{logit}(p_{ijk}) = alpha0_k + alpha1_k * wind_{ij} && \text{\# Detection probability affected by } wind
\end{aligned}
$$

There are again normal prior distributions for all species-specific parameters, with hyperparameters that describe the community to which these species belong. Most of what we say about the simulation of metacommunity presence/absence data (for `type = "det/nondet"`) has a simple analogy when the function is used to simulate metacommunity count data (with `type = "counts"`); therefore, we don't explain the latter specifically. By default, the function generates data where each species has its own constant value of occupancy probability and detection probability, but there are no covariate effects.

```
# Execute function with default arguments
set.seed(1234)
data <- simComm(type="det/nondet", nsite=30, nrep=3, nspec=100,
mean.psi=0.25, sig.lpsi=1, mu.beta.lpsi=0, sig.beta.lpsi=0,
mean.lambda=2, sig.loglam=1, mu.beta.loglam=1, sig.beta.loglam=1,
mean.p=0.25, sig.lp=1, mu.beta.lp=0, sig.beta.lp=0, show.plot = TRUE)
# data <- simComm() # same
```

Executing the function produces two plots that visualize the simulated system. The first plot shows the relationships between occupancy probability and the *habitat* covariate and between detection probability and the *wind* covariate (Figure 11.1). The function's default arguments specify zero effects of both covariates, and hence all lines are horizontal. If you specify nonzero standard deviations for the

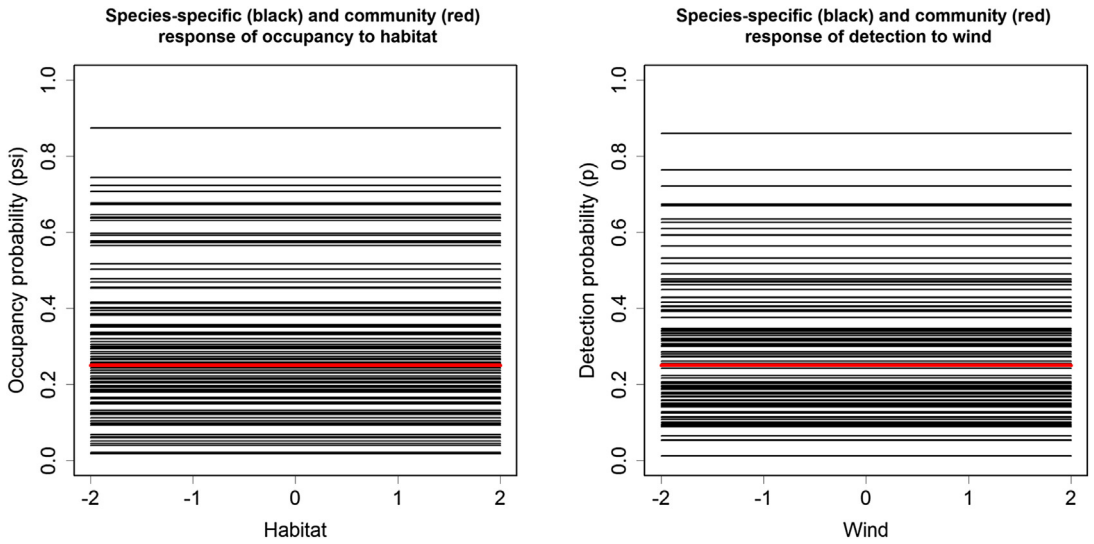

FIGURE 11.1

Species-specific and community-level responses of the probability of occupancy to the *habitat* covariate (left), and of the probability of detection to the *wind* covariate, in data simulated using function `simComm` with default settings. Black lines show the value of occupancy and detection probability for each of the 100 species and red lines show the community mean response. This is the first visualization produced by function `simComm`.

two normal distributions that govern species heterogeneity in the two covariate relationships, you will generate nonparallel lines (Figure 11.2), where you can see two important features of a community: (1) the community average of the response to some covariate may be very different from the response of the individual species; and (2) individual species may differ greatly in their response to a covariate. For instance, the community response of occupancy to the habitat is zero (the red line is horizontal), but most individual species are actually responding very clearly to that covariate. Moreover, there are many species with a positive response, but about just as many with a negative response.

The second plot produced when executing function `simComm` (Figure 11.3) depicts three matrices that show the true presence/absence pattern (z_{ik}) generated (top left), the observed detection frequency (i.e., the number of times that species k is detected at site i; top right), and the sites i where species k occurred but was never detected—that is, the combined presence/absence measurement error (bottom left). The frequency distributions of the true and observed numbers of species occurring per site are depicted in red and blue, respectively (bottom right). Finally, two interesting statistics are shown in the titles of the two plots on the left of this figure: one is the finite-sample number of occurring species (96 in this realization of the process), and the other is the observed number of species (92 in this realization). It is important to recognize that even though we specified 100 species in the regional pool in the area represented by our spatial sample of 30 sites, the 30 sampled sites only contained 96 species and failed to contain the remaining 4 species in the (meta)community. Thus, there is a sense in which there are two values for the "true species richness," one being 100 and the other (here) 96.

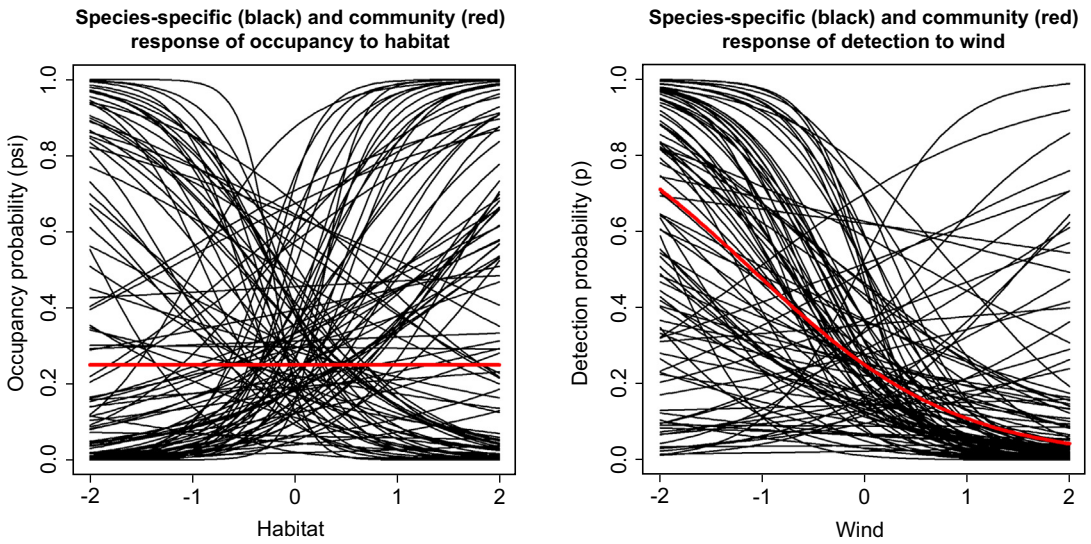

FIGURE 11.2

Species-specific and community-level response of the probability of occupancy to the *habitat* covariate (left) and of the probability of detection to the *wind* covariate in data simulated using function `simComm` with the following arguments: `mu.beta.lpsi=0, sig.beta.lpsi=2, mu.beta.lp=-1, sig.beta.lp=1` (other settings were at default values).

Dupuis et al. (2011) call them the *unconditional* and the *conditional* species richness, where conditional means "conditional on occurrence anywhere in the studied area." You will see below that the finite-sample (or conditional) total number of species (relative to the unconditional total) depends on the number of sites sampled, and on the mean and the among-species variability of occupancy probability.

Apart from the two plots, executing the function produces a large list of output that we here explain interspersed into an example. The first 14 elements simply repeat the function arguments that were operative during the function call. The remaining elements represent things produced by the function.

```
str(data)
List of 26
 $ type          : chr "det/nondet"
 $ nsite         : num 30
 $ nrep          : num 3
 $ nspec         : num 100
 $ mean.psi      : num 0.25
 $ mu.lpsi       : num -1.1
 $ sig.lpsi      : num 1
 $ mu.beta.lpsi  : num 0
 $ sig.beta.lpsi : num 0
 $ mean.p        : num 0.25
 $ mu.lp         : num -1.1
```

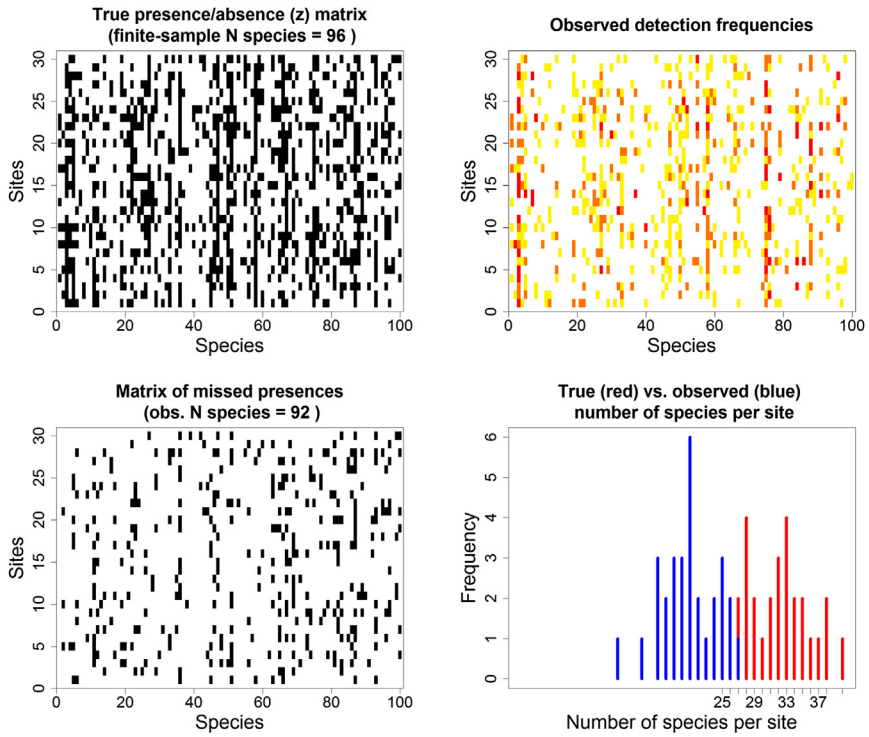

FIGURE 11.3

Top left: true presence and absence (z_{ik}) of species k at site i. Top right: number of times that species k was detected at site i (detection frequency; redder is more). Bottom left: Sites i where species k occurred, but was missed—i.e., never detected (this represents presence/absence measurement error). Bottom right: true (red) and observed (blue) frequency distribution of the number of species per site. This is the second visualization produced by function `simComm`.

```
 $ sig.lp        : num 1
 $ mu.beta.lp    : num 0
 $ sig.beta.lp   : num 0
# psi[i,k]: occupancy probability for site i and species k
 $ psi           : num [1:30, 1:100] 0.216 0.216 0.216 0.216 0.216 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:30] "site1" "site2" "site3" "site4" ...
  .. ..$ : chr [1:100] "sp1" "sp2" "sp3" "sp4" ...
# p[i,j,k]: detection probability for site i, occasion j and species k
 $ p             : num  [1:30, 1:3, 1:100] 0.449 0.449 0.449 0.449 0.449 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : chr [1:30] "site1" "site2" "site3" "site4" ...
  .. ..$ : chr [1:3] "rep1" "rep2" "rep3"
  .. ..$ : chr [1:100] "sp1" "sp2" "sp3" "sp4" ...
```

```
# z[i,k]: true presence/absence for site i and species k
 $ z            : int [1:30, 1:100] 0 0 0 1 0 0 0 1 0 1 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:30] "site1" "site2" "site3" "site4" ...
  .. ..$ : chr [1:100] "sp1" "sp2" "sp3" "sp4" ...
# z.obs[i,k]: observed presence/absence for site i and species k (i.e., an indicator for
whether species k was ever detected at site i during J surveys)
 $ z.obs        : int [1:30,   1:100] 0 0 0 1 0 0 0 1 0 1 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:30] "site1" "site2" "site3" "site4" ...
  .. ..$ : chr [1:100] "sp1" "sp2" "sp3" "sp4" ...
# y[i,j,k]: detection/nondetection data for site i, occasion j and species k (for all 100
species)
 $ y.all        : int [1:30, 1:3, 1:100] 0 0 0 0 0 0 0 1 0 1 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : chr [1:30] "site1" "site2" "site3" "site4" ...
  .. ..$ : chr [1:3] "rep1" "rep2" "rep3"
  .. ..$ : chr [1:100] "sp1" "sp2" "sp3" "sp4" ...
# y[i,j,k]: detection/nondetection data for site i, occasion j and species k (only for the 92
species that were detected at least once)
 $ y.obs        : int [1:30, 1:3, 1:92] 0 0 0 0 0 0 0 1 0 1 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : chr [1:30] "site1" "site2" "site3" "site4" ...
  .. ..$ : chr [1:3] "rep1" "rep2" "rep3"
  .. ..$ : chr [1:92] "sp1" "sp2" "sp3" "sp4" ...
# detection frequency for all 100 species
 $ y.sum.all    : int [1:30, 1:100] 0 0 0 1 0 0 0 1 0 1 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:30] "site1" "site2" "site3" "site4" ...
  .. ..$ : chr [1:100] "sp1" "sp2" "sp3" "sp4" ...
# detection frequency only for the 92 species detected at least once
 $ y.sum.obs    : int [1:30, 1:92] 0 0 0 1 0 0 0 1 0 1 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:30] "site1" "site2" "site3" "site4" ...
  .. ..$ : chr [1:92] "sp1" "sp2" "sp3" "sp4" ...
# Finite sample (or conditional) species richness: number of species that occur in the 30
sampled sites
 $ Ntotal.fs    : int 96
# The observed version of the same (difference being due to imperfect detection)
 $ Ntotal.obs   : int 92
# The true number of species occurring at each of the 30 sites
 $ S.true       : Named int [1:30] 27 26 30 25 35 33 33 33 34 38 ...
  ..- attr(*, "names")= chr [1:30] "site1" "site2" "site3" "site4" ...
# The number of species observed at each of the 30 sites
 $ S.obs        : Named int [1:30] 17 17 16 11 21 20 19 25 20 24 ...
  ..- attr(*, "names")= chr [1:30] "site1" "site2" "site3" "site4" ...
```

As always, we would like to warmly encourage you to *play community*—i.e., execute this function many times, with changed function arguments, and observe how the output is affected by

your choices and be astonished at how one realization from a given stochastic process can differ strikingly from another one. This will be a big help for your general understanding of some fundamentals of community ecology as well as of their measurement when the latter is contaminated by false-negative errors. Here we show a couple of settings that may be interesting. We note that you must have >1 site and replicate and usually >2 species, otherwise the function breaks. If you want to simulate data for a single site, replicate, or species, you must run the function for multiple sites, replicates, or species and then pull out a subset of the data representing a single site, replicate, or species. Don't specify no occurring species (`mean.psi = 0`) or invisible species (`mean.p = 0`)—both cause the function to crash—but the opposite (all species occurring everywhere, `mean.psi = 1`, or perfect detectability, `mean.p = 1`) works, though the plots don't look very nice anymore.

```
# Some possibly interesting settings of the function
data <- simComm(nsite = 267, nspec = 190, mean.psi = 0.25, sig.lpsi = 2,
mean.p = 0.12, sig.lp = 2) # similar to Swiss MHB; see Section 11.3
data <- simComm(mean.psi = 1)          # all species occur at every site
data <- simComm(mean.p = 1)            # no measurement error (perfect detection)

# Effect of spatial sample size (nsite) on species richness in sample (Ntotal.fs)
data <- simComm(nsite=50, nspec = 200) # 1-3 are usually missed in sample
data <- simComm(nsite=30, nspec = 200) # 4-6 usually missed
data <- simComm(nsite=10, nspec = 200) # around 30 typically missed
```

You should always repeatedly execute the function to study the average and the variation in the behavior of some feature in the metacommunity or of their measurement. Here is a simple simulation that tells you how many species in a metacommunity of 200 species will typically be missed when sampling 10 sites (using default function arguments otherwise): the answer is about 25 to 30 (under the assumptions about community assembly embodied by the function).

```
# Check for frequentist characteristics of such statistics
temp <- rep(NA, 100)
for(i in 1:100){
   cat("\nSimrep", i)
   temp[i] <- simComm(nsite=10, nspec = 200, show.plot = F)$Ntotal.fs
}
hist(200-temp, breaks = 30, main = "Number of species in the metacommunity
\nthat do not occur in the 10 sampled sites", col = "gold")
```

We conduct two further small simulations to illustrate an important feature of a metacommunity that holds regardless of any measurement error, and then a feature of presence/absence *measurements* of a metacommunity (i.e., which is due to measurement error when studying a metacommunity). In both, we simulate metacommunities composed of 200 species that are sampled at 50 sites. With the default function arguments, about 95% of the realizations will have at least 196 species actually occurring at the 50 studied sites. In the first simulation, we look at the effects of community mean occupancy probability (ψ) and among-species variability in logit(ψ) on the proportion of these 200 species in the metacommunity that actually occur in the sampled 50 sites. We generate 50 data sets for each combination of two factors with 10 levels, the first being the community mean occupancy probability (`mean.psi`, which we vary between 0.01 and 0.25) and the second being the interspecific variability of occupancy (`sig.lpsi`, which we vary between 0.1 and 5).

```
# Simulation 1: effects of psi and sd(logit(psi)) on number of species actually occurring in
the 50 sampled sites
simrep <- 50           # Run 50 simulation reps
mpsi <- seq(0.01, 0.25,,10)
slpsi <- seq(0.1, 5,,10)
results1 <- array(NA, dim = c(10, 10, simrep))
for(i in 1:10){    # Loop over levels of factor mean.psi (mpsi)
 for(j in 1:10){   # Loop over levels of factor sig.lpsi (slpsi)
  for(k in 1:simrep){
   cat("\nDim 1:",i, ", Dim 2:", j, ", Simrep", k)
   tmp <- simComm(nsite=50, nspec = 200, show.plot = F, mean.psi = mpsi[i],
     sig.lpsi = slpsi[j])
   results1[i,j,k] <- tmp$Ntotal.fs
  }
 }
}
```

In the second simulation, we look at the effects of community mean detection probability (p) and among-species variability in logit(p) on the proportion of species occurring in the 50 sample sites that are detected at least once—i.e., on the converse of the measurement error in conditional species richness (Dupuis et al., 2011). We again generate 50 data sets for each combination of two factors with 10 levels, the first being the community mean detection probability (`mean.p`, which we vary between 0.01 and 0.25), and the second being the interspecific variability of detection (`sig.lp`, which we vary between 0.1 and 5).

```
# Simulation 2: effects of p and sd(logit(p)) on the proportion of the species occurring in
the 50 sampled sites that are detected at least once
simrep <- 50        # Run 50 simulation reps again
mp <- seq(0.01, 0.25,,10)
slp <- seq(0.1, 5,,10)
results2 <- array(NA, dim = c(10, 10, simrep, 2))
for(i in 1:10){    # Loop over levels of factor mean.p (mp)
  for(j in 1:10){  # Loop over levels of factor sig.lp (slp)
    for(k in 1:simrep){
      cat("\nDim 1:",i, ", Dim 2:", j, ", Simrep", k)
      tmp <- simComm(nsite=50, nspec = 200, show.plot = F, mean.p = mp[i],
        sig.lp = slp[j])
      results2[i,j,k,] <- c(tmp$Ntotal.fs, tmp$Ntotal.obs)
    }
  }
}
```

We visualize the results in two image plots that show the mean (over the 50 simulated data sets) as a function of the 10 levels of each simulation factor (Figure 11.4).

```
# Plot these two prediction matrices
par(mfrow = c(1, 2), mar = c(5,5,2,2), cex.lab = 1.5, cex.axis = 1.5)
mapPalette <- colorRampPalette(c("grey", "yellow", "orange", "red"))
```

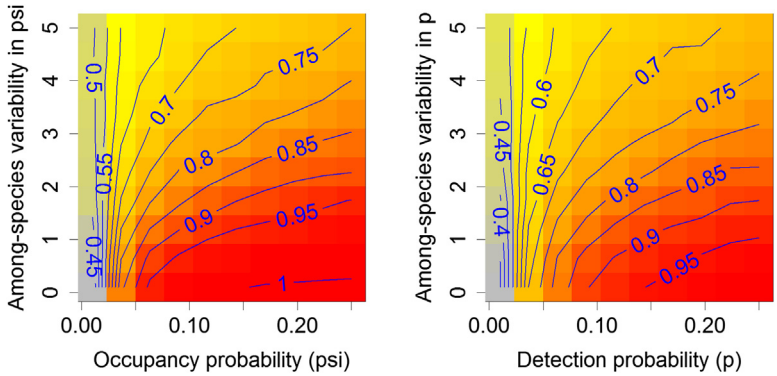

FIGURE 11.4

Left: Effects of community mean occupancy probability (ψ) and among-species variability in occupancy probability (logit(ψ)) on the proportion of 200 species in the wider community that actually *occur* somewhere within the 50 sampled sites (this is a true pattern in the ecological process). Right: Effects of community mean detection probability (p) and among-species variability in detection probability (logit(p)) on the proportion of *detected* species among those that occur in the sampled 50 sites (this is a pattern in the measurement error process). Each cell is the mean of 50 simulated data sets.

```
# Plot proportion of species occurring in sampled sites (Fig. 11-4 left)
z1 <- apply(results1/200, c(1,2), mean) # Prop species occurring
image(x=mpsi, y=slpsi, z=z1, col = mapPalette(100), axes = T, xlab = "Occupancy probability
(psi)", ylab = "Among-species variability in psi")
contour(x=mpsi, y=slpsi, z=z1, add = T, col = "blue", labcex = 1.5, lwd = 1.5)

# Plot proportion of species detected in sampled sites (Fig. 11-4 right)
z2 <- apply(results2[,,,2] / results2[,,,1], c(1,2), mean)
image(x=mp, y=slp, z=z2, col = mapPalette(100), axes = T, xlab = "Detection probability
(p)", ylab = "Among-species variability in p")
contour(x=mp, y=slp, z=z2, add = T, col = "blue", labcex = 1.5, lwd = 1.5)
```

This concludes our introduction to metacommunity studies based on the occurrence of individual species. We have seen that the average metacommunity may behave quite differently from the behaviour of the individual species. We have also seen that we can distinguish two characterizations of the size of a metacommunity: one unconditional, which is the number of species that *would* be sampled at saturation effort in a region (in terms of the spatial replicates, the number of sites), and the other the realized or conditional on occurrence in at least one of the sampled sites (Dupuis et al., 2011). The former is also the asymptote of a species accumulation curve (Dorazio et al., 2006); see also Section 11.9. Whether one of them is relevant for you, and if so which, is something each ecologist must decide independently. In addition to metacommunity size, we have the size of local communities (local species richness); see Figure 11.3 (bottom left). Other terms for these descriptors of the richness of a metacommunity include gamma and alpha diversity (Whittaker et al., 2001), where the term beta diversity is used to characterize the dissimilarity of the communities among different sites—i.e., the spatial variability in species occurrence within a metacommunity.

The other important topic in the analysis of a metacommunity that we have emphasized is that of the measurement error induced by imperfect detection. Measurement error in community studies may bias *any* descriptor of a community or metacommunity, including all three types of diversity and community and individual-species responses to covariates.

11.3 METACOMMUNITY DATA FROM THE SWISS BREEDING BIRD SURVEY MHB

To illustrate HMs for communities we use data collected in the Swiss breeding bird survey MHB (Monitoring Häufige Brutvögel; Schmid et al., 2004). The most complex models that we will fit will be constructed exactly analogous to the concept of our data simulation in Section 11.2. That is, we will assume independence among species in both occurrence and detection. Independence of occurrence may seem like a strong assumption, but whether it holds is likely to be scale-dependent: for relatively large sites such as the 1-km^2 quadrats in the Swiss breeding bird survey MHB, it appears unlikely that the presence or absence or the abundance of one species has any effect on that of another beyond what we can explain by the habitat (and it is this *conditional independence* that we mean when we make this assumption).

In the MHB, all individuals of all breeding bird species are surveyed using territory mapping during three surveys in the breeding season (approx. mid-April through the end of June) in 267 1-km^2 sampling units laid out as a grid over Switzerland. Only two surveys are conducted in quadrats above the tree line. Surveys are conducted along irregular and quadrat-specific transects of length ranging from 1.2 to 9.4 km (mean 5.1). Thus, in the MHB counts and territory ID data are collected, but we will first reduce the information content of the original data and simply model the species-specific detection/nondetection data—i.e., the replicated presence/absence measurements. In Section 11.10 we will model the counts directly. We chose data from 2014, when 266 quadrats were surveyed and 145 species detected. The variables in our data set are explained in the following interspersed R output.

```
# Read in data set and look at data first
data <- read.csv2("MHB_2014.csv", header = T, sep = ";", dec= ".")
str(data)   # (Later read in as a system file of R package AHM)

'data.frame':  42186 obs. of  24 variables:
# record number
 $ id           : int  1 2 3 4 5 6 7 8 9 10 ...
# species ID: numeric, English name, Latin abbreviation and Latin name
 $ specid       : int  50 50 50 50 50 50 50 50 50 50 ...
 $ engname      : Factor w/ 158 levels "Alpine Accentor",..: 98 98 98 98 98 98 98 ...
 $ latabb       : Factor w/ 158 levels "ACCGEN","ACCNIS",..: 145 145 145 145 145 145 ...
 $ latname      : Factor w/ 158 levels "Accipiter gentilis",..: 146 146 146 146 146 146 ...
# species traits: body length (cm), wing span (cm), body mass (g)
 $ body.length : int  27 27 27 27 27 27 27 27 27 27 ...
 $ wing.span    : int  43 43 43 43 43 43 43 43 43 43 ...
 $ body.mass    : int  150 150 150 150 150 150 150 150 150 150 ...
# x and y coordinates of sample 1km^2 quadrat
 $ coordx       : int  922942 928942 928942 934942 934942 946942 946942 952942 ...
 $ coordy       : int  63276 79276 103276 95276 111276 95276 111276 119276 111276 ...
```

```
# length of survey route and number of surveys per breeding season
 $ rlength   : num 6.4 5.5 4.3 4.5 5.4 3.6 3.9 6.1 5.8 4.5 ...
 $ nsurvey   : int 3 3 3 3 3 3 3 3 3 ...
# mean quadrat elevation (in metres) and forest cover (as a percentage)
 $ elev      : int 450 450 1050 950 1150 550 750 650 550 550 ...
 $ forest    : int 3 21 32 9 35 2 6 60 5 13 ...
# number of birds counted for 1st through 3rd survey
 $ count141  : int 0 0 0 0 0 0 0 0 0 0 ...
 $ count142  : int 0 0 0 0 0 0 0 0 0 0 ...
 $ count143  : int 0 0 0 0 0 0 0 0 0 0 ...
# survey date (1 = 1 April 2014)
 $ date141   : int 21 26 25 40 16 52 18 17 18 25 ...
 $ date142   : int 52 47 52 55 38 61 40 39 45 50 ...
 $ date143   : int 70 59 73 65 62 69 60 61 59 76 ...
# survey duration (in minutes)
 $ dur141    : int 215 195 210 310 240 180 180 195 190 195 ...
 $ dur142    : int 220 175 270 300 240 145 195 225 180 203 ...
 $ dur143    : int 240 185 210 285 210 140 180 210 205 215 ...
# id for surveyor who did the 3 surveys in 2014
 $ obs14     : int 386 147 77 293 77 361 77 77 179 165 ...

# Create various species lists (based on English names and systematic order)
(species.list <- levels(data$engname))  # alphabetic list
(spec.name.list <- tapply(data$specid, data$engname, mean)) # species ID
(spec.id.list <- unique(data$specid))  # ID list
(ordered.spec.name.list <- spec.name.list[order(spec.name.list)]) # ID-order list
```

We first grab the three counts in an array and convert them into simple replicated presence/absence measurements or detection/nondetection data. Note how this illustrates once again the (one-way) deterministic relationship between these two types of data.

```
COUNTS <- cbind(data$count141, data$count142, data$count143) # Counts 2014
DET <- COUNTS
DET[DET > 1] <- 1          # now turned into detection/nondetection data
```

In BUGS, it is convenient to fit the model to the data formatted in a three-dimensional array, because the model description is much neater if we can use the dimensions of a multidimensional array to convey the information about factors such as site, species, and replicate survey. We do this next. There are different ways in which you may organize this array. We organize it such that the third dimension of the array indexes species, and we name it using the appropriate species list. This array is a direct generalization of the typical data array for single-species occupancy models, where we usually model a site-by-replicate matrix of species detections (Chapter 10). Here, we stack the species-specific matrices, and the species represent the slices of the resulting 3-D array.

```
# Put detection data into 3D array: site x rep x species
nsite <- 267                    # number of sites in Swiss MHB
nrep <- 3                       # number of replicate surveys per season
nspec <- length(species.list)  # 158 species occur in the 2014 data
y <- array(NA, dim = c(nsite, nrep, nspec))
```

```
for(i in 1:nspec){
  y[,,i] <- DET[((i-1)*nsite+1):(i*nsite),]
}
dimnames(y) <- list(NULL, NULL, names(ordered.spec.name.list))

# Check data for one species, here chaffinch, and pull them out from 3D array
which(names(ordered.spec.name.list) == "Common Chaffinch")
(tmp <- y[,,which(names(ordered.spec.name.list) == "Common Chaffinch")])

# Frequency distribution of number of surveys per site in chosen year
table(nsurveys <- apply(y[,,1], 1, function(x) sum(!is.na(x))))
 0  2  3
 1 47 219

# Which site has NA data in 2014 ?
(NAsites <- which(nsurveys == 0) )
[1] 30
```

Hence, 219 sites were surveyed three times, 47 twice, and one site was not surveyed in 2014. We next look at the observed number of occupied sites among the 266 with nonmissing values in 2014.

```
# Observed number of occupied sites
tmp <- apply(y, c(1,3), max, na.rm = TRUE)
tmp[tmp == -Inf] <- NA      # Only 266 quadrats surveyed in 2014
sort(obs.occ <- apply(tmp, 2, sum, na.rm = TRUE))

          Little Bittern       Little Ringed Plover                  Barn Owl
                       0                          0                         0
     Eurasian Eagle-Owl                 Little Owl          Eurasian Hoopoe
                       0                          0                         0
White-backed Woodpecker                Bluethroat       Great Reed Warbler
                       0                          0                         0
Red-breasted Flycatcher           Woodchat Shrike         Common Rosefinch
                       0                          0                         0
        Ortolan Bunting                White Stork             Greylag Goose
                       0                          1                         1
[ ... Output truncated ..... ]
           Mistle Thrush                  Great Tit                  Coal Tit
                     190                        198                       199
             Song Thrush          Common Chiffchaff         Eurasian Blackcap
                     210                        213                       217
          European Robin           Common Blackbird               Winter Wren
                     219                        223                       234
       Common Chaffinch             Black Redstart
                     243                        244

# Plot species 'occurrence frequency' distribution (not shown)
plot(sort(obs.occ), xlab = "Species number", ylab = "Number of quads with detections")
```

```
# Drop data from species that were not observed in 2014
toss.out <- which(obs.occ == 0)
y <- y[,,-toss.out]
obs.occ <- obs.occ[-toss.out]
# Redefine nspec as the observed # species: 145
( nspec <- dim(y)[3] )
```

We are now left with binary detection/nondetection data from 267 sites (of which one has missing data) and two to three replicate surveys for 145 species observed during the 2014 surveys. We could toss out the data from site 30 with the missing data, but we keep the site to illustrate the ability of some models to estimate things for "new" or unsurveyed sites.

```
str(y)
num [1:267, 1:3, 1:145] 0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "dimnames")=List of 3
..$ : NULL
..$ : NULL
..$ : chr [1:145] "Little Grebe" "Great Crested Grebe" "Grey Heron" "White Stork" ...
```

We continue our descriptive overview of the data and compute the observed number of species per quadrat ("observed species richness").

```
# Get observed number of species per site
tmp <- apply(y, c(1,3), max, na.rm = TRUE)
tmp[tmp == "-Inf"] <- NA
sort(C <- apply(tmp, 1, sum))  # Compute and print sorted species counts
  [1]  6  6  6  7  7  7  8  8  9  9  9 10 10 10 11 12 12 12 13 14 14 14 15 15 15
 [26] 15 16 17 17 18 18 19 19 19 19 19 19 20 21 24 24 24 24 25 25 25 25 25 25 26
 [51] 26 26 26 27 27 27 28 28 28 28 29 29 29 29 29 29 30 30 30 30 30 30 30 30 30
 [76] 30 31 31 31 31 31 31 32 32 32 32 32 32 32 32 32 32 32 32 32 33 33 33 33 33 33
[101] 33 33 33 33 33 34 34 34 34 34 34 34 34 34 34 34 34 34 34 34 35 35 35 35 35 35
[126] 35 35 35 35 36 36 36 36 36 36 36 37 37 37 37 37 37 37 37 37 37 38 38 38 38
[151] 38 38 38 38 38 38 38 38 38 38 38 38 39 39 39 39 39 39 39 39 39 39 39 39 39
[176] 40 40 40 40 40 40 40 40 40 40 40 40 41 41 41 41 41 41 41 41 41 41 42 42 42
[201] 42 42 42 42 42 42 42 42 42 43 43 43 43 43 43 43 43 43 44 44 44 44 44 44 44
[226] 44 44 44 44 44 45 45 45 45 45 45 45 45 45 45 46 46 46 46 46 46 46 47 47 47
[251] 47 47 47 47 48 48 48 48 48 48 49 50 51 51 56 59
```

So the *observed* species richness, or *observed* community size, in 266 surveyed 1-km^2 quadrats in Switzerland varied from 6 to 59, with an average of 34. Let's plot that (Figure 11.5).

```
plot(table(C), xlim = c(0, 60), xlab = "Observed number of species", ylab = "Number of
quadrats", frame = F)
abline(v = mean(C, na.rm = TRUE), col = "blue", lwd = 3)
```

11.4 OVERVIEW OF SOME MODELS FOR METACOMMUNITIES

We use the MHB 2014 data set to fit a series of models for species richness and other features of a metacommunity. These models will form a progression of 11 models that range from very simplistic to increasingly realistic/mechanistic; see overview in Table 11.1. We believe that such a progression will help you to both *understand* the differences between different models for communities and better grasp

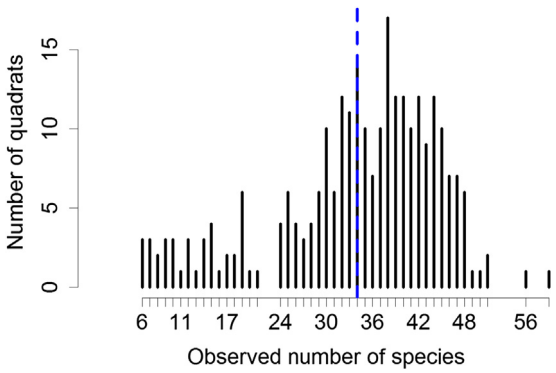

FIGURE 11.5

Frequency distribution of the observed number of breeding bird species per 1-km^2 quadrat (that is, observed species richness) in the Swiss MHB survey in 2014 (n = 267 quadrats, with data from one quadrat missing). Blue line shows the mean of 34 species.

the difficulties when fitting them in BUGS. The original MHB data can be denoted y_{ijk} and contain the count of species k ($k = 1…145$) at site i ($i = 1…267$) during replicate survey (or occasion) j ($j = 1…3$). In models 1−10, we quantize these data to become binary detection/nondetection data, so that y_{ijk} contains the detection (1) or nondetection (0) of species k at site i during occasion j. In addition, in models 1−4 we summarize these data one more step and treat as a response the number of detected species. In model 11 we will model the counts directly.

The 11 different modeling frameworks can be described as follows:

1. Our simplest models (1−3) for the community consist of regression models that relate the *number of observed species* to covariates. Two drawbacks of such an analysis are that species richness measurement error is not allowed for, and that the species identities are lost—i.e., individual species are not distinguished across sites nor even across replicate surveys of each site.
2. A slightly more advanced approach (model 4) to community modeling is the adoption of a straightforward *N*-mixture model for the replicated counts of observed species (model 4). Under this approach, imperfect detection of species is accounted for, but species heterogeneity in detection cannot be modeled apart from differences among sites and surveys, leading to underestimation of local species richness due to the second law of capture-recapture (which is that unmodelled detection heterogeneity induces a negative bias in abundance and occupancy estimators). Furthermore, species identities are again not retained. In terms of the use of information about species identities, there is also an intermediate model between the N-mixture and the occupancy models (in 3., below): the multinomial mixture model (see Chapter 7). When applied to *species* detection/nondetection data, it would only ignore species identity *across* sites, but retain species identity *within* a site across replicate surveys. Indeed, we could use `unmarked` to fit a multinomial mixture model of the M_h type to estimate site-specific species richness in a heterogeneity model that would allow each species to have its own detection probability, exactly as shown in Section 7.8.3 for site-stratified estimation of population abundance. One interesting side effect of this formulation of the species richness estimation problem would be that we could then directly model local species richness as a function of covariates. See Exercise 2 for an example of this idea applied to the MHB data set using both `unmarked` and BUGS.

Table 11.1 Overview of hierarchical (and some "flat") models for (meta)community inference in this chapter.

Nr.	Model Name/ Description	Modeled State	Inference on Community	Inference on Individual Species	Serial Autocorrelation	Measurement Error Model (Detection Probability)	Allows Detection Heterogeneity	Inference About Unseen Species	Inference About Community Composition
1	Poisson GLM for maximum species count	Species count	yes	no	NA	no, only covs.	NA	no	no
2	Poisson GLM for replicated species counts	Species count	yes	no	no	only covs.	no	no	no
3	Random-effects Poisson GLM for replicated species counts	Species count	yes	no	yes	only covs.	no	no	no
4	*N*-mixture model for replicated species counts	Species count	yes	no	yes	yes	no	no	no
5	Fixed-effects community occupancy model	Pres/Abs	yes	yes	yes	yes	yes	no	yes
6	Random-effects community occupancy model with (ψ, p) correlation	Pres/Abs	yes	yes	yes	yes	yes	no	yes
7	Random-effects community occupancy model with categorical covariate to explain species effects (simple DR model without DA)	Pres/Abs	yes	yes	yes	yes	yes	no	yes

8	Random-effects community occupancy model with continuous covariate to explain species effects (simple DR model, no DA)	Pres/Abs	yes	yes	yes	yes	yes	no	yes
9	Full DR community occupancy model with DA, no covariates	Pres/Abs	yes	yes	yes	yes	yes	yes	yes
10	Full DR community occupancy model with DA and with covariates	Pres/Abs	yes	yes	yes	yes	yes	yes	yes
11	DRY community *N*-mixture model with covariates	Abund.	yes	yes	yes	yes	yes	yes[a]	yes

[a]*The DRY community N-mixture model (model 11) can be specified with or without DA.*

3. Model 5 is a straightforward site-occupancy model with one component model per observed species. Hence, *species identity is retained*, so the model accommodates features of the individual member species of a (meta)community as well as emerging characteristics of the metacommunity. Furthermore, heterogeneity among species in occurrence and detection can be fully accounted for. The simplest such model consists of fitting *separate occupancy model parameters* to each observed species, but inside a single model. No relationship among species will be imposed on the parameters. In ANOVA terms, species will be treated as fixed effects; the estimates for one species will not be affected by the data from another species. This analysis allows one to estimate (correcting for imperfect detection) the number of species occurring at each site among the total number of species ever detected in the metacommunity, but not the total number of species occurring in the entire metacommunity, because some species may have been missed everywhere. This is the approach taken in calculating the popular wildlife picture index (O'Brien et al., 2010).
4. In models 6–8, we treat each species as a random sample from the studied community; i.e., each species is assigned a random effect. These analyses allow us to *formally estimate characteristics of the* ***observed*** *community*—e.g., the mean probability of occupancy or detection, or the mean response of occupancy probability to a covariate for some environmental conditions. This approach also allows us to estimate the number of species occurring at each site among all the species that were ever detected anywhere in the metacommunity, but not the total size of the metacommunity. That is, no inference is made about those species that were never observed anywhere. These models are quite similar to the models developed by Gelfand et al. (2005, 2006) and Latimer et al. (2006).
5. The final two models for detection/nondetection data (models 9 and 10) accommodate the fact that some species in the metacommunity may never be observed anywhere at all. We can extend the random-effects multispecies model to those unseen species in an attempt to make an inference about the entire community. This analysis uses parameter-expanded DA (Tanner and Wong, 1987; Royle et al., 2007a; Royle and Dorazio, 2012)—i.e., the fitting of a more complex HM (with one additional hierarchical layer) to a modified data set, which includes an added portion of data to accommodate potential unseen species. This is the full Dorazio/Royle (DR) community occupancy model with data-augmentation (Dorazio and Royle, 2005; Dorazio et al., 2006).
6. The final model, 11, is the abundance version of the DR community, community *N*-mixture, or Dorazio/Royle/Yamaura (DRY) model (Yamaura et al., 2012).

In community occupancy models, in the absence of covariates that vary by replicate survey j, it is convenient to aggregate binary detection/nondetection data y_{ijk} into site- and occasion-specific counts by summing over replicates, and then model detection frequency $ysum_{ik}$—i.e., the number of detections of species k at site i—as a binomial random variable with a binomial index given by the number of surveys (which may or may not be the same for all sites and/or species). Binomial versions of the models are computationally much more efficient to analyze in BUGS.

Importantly, neither the community occupancy nor the community *N*-mixture model contains a structural parameter for species richness, since they model the occurrence or abundance of each individual species in the community. Species richness is then a derived quantity based on the occurrence of individual species. One simple way to model species richness as a function of other spatially or temporally indexed covariates is to do a two-step analysis and plug species-richness estimates into a regression analysis to model the effects of those covariates (e.g., see Tingley and Beissinger, 2013), but in the second analysis you must account for the estimation uncertainty coming from the first analysis. Section 11.6.4 gives an example of how to do this in the context of estimating an elevation profile of avian species richness in Switzerland. A more formal way of modeling species

richness is by adopting a logit-linear model for the data augmentation parameter omega in Section 11.7 (Sutherland et al., in review) or else by the construction shown in Section 7.8.4. We may show this in volume 2.

11.5 COMMUNITY MODELS THAT IGNORE SPECIES IDENTITY

We start by fitting a few fairly simplistic models to the Swiss MHB 2014 data set that focus on spatial or spatiotemporal patterns of species richness only. These models directly describe the number of observed species, but do not keep track of *which* species was observed where and when. That is, these models do not keep track of species identity across replicate surveys at each site and especially, they don't keep track of species identity across replicate sites. Thus, any inferences about individual species are impossible, and at the community level, we risk missing relationships with environmental covariates if a similar number of species show a negative and a positive response (see Figure 11.2, left). Finally, accommodation of false-negative measurement error of species richness is only possible in a very rudimentary way (and in practice is hardly ever done in the type of models in this section).

11.5.1 SIMPLE POISSON REGRESSION FOR THE OBSERVED COMMUNITY SIZE

Probably the most common approach to inference about species richness in much of current ecology would be to fit some kind of regression model to the observed numbers of species. Here, we relate this number to elevation (linear and squared) and forest cover (linear). In the simplest approach with this model, we might discard the information coming from repeated measurements of presence/absence and simply model the total number of species detected across all survey occasions combined. If we want to account for any effects of survey date or survey duration, we then have to summarize these over the surveys, typically by taking the mean for each site (see end of this section for a version of this model where we don't need to aggregate the data). We fit the following model to the observed number of species C at site i:

$$C_i \sim Poisson(\lambda_i)$$

$$\log(\lambda_i) = \gamma_0 + \gamma_1 * elev_i + \gamma_2 * elev_i^2 + \gamma_3 * forest_i + \gamma_4 * mean.date_i + \gamma_5 * mean.date_i^2 + \gamma_6 * mean.duration_i$$

Presumably, the first three covariates would be assumed to act on the true species richness at each site, while the last three might be seen to adjust for seasonal and survey-effort-related differences in the detectability of the average species. However, there is no way to explicitly specify such mechanisms, since in this model there is no distinction between a true ecological state (referring to the true number of species at each site) and the measurement process that relates such a state to our observed data which is only an index to true species richness. To emphasize that we don't know the nature of these effects in this model, we call the coefficients `gamma` rather than `beta` or `alpha` as we do elsewhere throughout the book to denote covariate effects for the true state and its measurement, respectively.

```
# Get covariates and standardize them
# Quadrat elevation and forest cover
orig.ele <- data$elev[1:nsite]
(mean.ele <- mean(orig.ele, na.rm = TRUE))
(sd.ele <- sd(orig.ele, na.rm = TRUE))
ele <- (orig.ele - mean.ele) / sd.ele
orig.forest <- data$forest[1:nsite]
```

```
(mean.forest <- mean(orig.forest, na.rm = TRUE))
(sd.forest <- sd(orig.forest, na.rm = TRUE))
forest <- (orig.forest - mean.forest) / sd.forest

# Average date and duration of survey
tmp <- cbind(data$date141, data$date142, data$date143)[1:nsite,]
orig.mdate <- apply(tmp, 1, mean, na.rm = TRUE)
(mean.mdate <- mean(orig.mdate[-NAsites])) # drop unsurveyed site
(sd.mdate <- sd(orig.mdate[-NAsites]))
mdate <- (orig.mdate - mean.mdate) / sd.mdate
mdate[NAsites] <- 0                # impute mean for missing

tmp <- cbind(data$dur141, data$dur142, data$dur143)[1:nsite,]
orig.mdur <- apply(tmp, 1, mean, na.rm = TRUE)
(mean.mdur <- mean(orig.mdur[-NAsites]))
(sd.mdur <- sd(orig.mdur[-NAsites]))
mdur <- (orig.mdur - mean.mdur) / sd.mdur
mdur[NAsites] <- 0                 # impute mean for missing

# Bundle data and summarize input data for BUGS
str( win.data <- list(C = C, nsite = length(C), ele = ele, forest = forest, mdate = mdate,
mdur = mdur) )
List of 6
 $ C     : num [1:267] 30 32 51 40 44 41 34 35 42 35 ...
 $ nsite : int 267
 $ ele   : num [1:267] -1.1539 -1.1539 -0.2175 -0.3735 -0.0614 ...
 $ forest: num [1:267] -1.1471 -0.4967 -0.0992 -0.9303 0.0092 ...
 $ mdate : num [1:267] -0.3814 -0.6011 -0.2415 -0.0418 -0.9207 ...
 $ mdur  : num [1:267] -0.274 -0.994 -0.184 1.047 -0.184 ...

# Specify model in BUGS language
sink("model1.txt")
cat("
model {

# Priors
gamma0 ~ dnorm(0, 0.001)           # Regression intercept
for(v in 1:6){                     # Loop over regression coef's
  gamma[v] ~ dnorm(0, 0.001)
}

# Likelihood for Poisson GLM
for (i in 1:nsite){
  C[i] ~ dpois(lambda[i])
  log(lambda[i]) <- gamma0 + gamma[1] * ele[i] + gamma[2] * pow(ele[i],2) +
gamma[3] * forest[i] + gamma[4] * mdate[i] + gamma[5] * pow(mdate[i],2) +
gamma[6] * mdur[i]
}
}
",fill = TRUE)
sink()
```

```
# Initial values
inits <- function() list(gamma0 = rnorm(1), gamma = rnorm(6))

# Parameters monitored
params <- c("gamma0", "gamma")

# MCMC settings
ni <- 6000 ; nt <- 4 ; nb <- 2000 ; nc <- 3

# Call WinBUGS from R (ART <1 min)
out1 <- bugs(win.data, inits, params, "model1.txt", n.chains = nc,
n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory = bugs.dir,
working.directory = getwd())

# Call JAGS from R (ART <1 min), check convergence and summarize posteriors
library(jagsUI)
out1J <- jags(win.data, inits, params, "model1.txt", n.chains = nc, n.thin = nt,
n.iter = ni, n.burnin = nb)
traceplot(out1J)    ;    print(out1J, dig = 3)
```

```
            mean     sd    2.5%     50%   97.5% overlap0     f    Rhat n.eff
gamma0     3.701  0.020   3.662   3.701   3.739    FALSE 1.000   1.001  1790
gamma[1]  -0.169  0.021  -0.209  -0.169  -0.130    FALSE 1.000   1.000  2738
gamma[2]  -0.152  0.018  -0.187  -0.152  -0.118    FALSE 1.000   1.002  1734
gamma[3]  -0.015  0.012  -0.038  -0.015   0.010     TRUE 0.882   1.000  3000
gamma[4]  -0.003  0.024  -0.049  -0.003   0.044     TRUE 0.543   1.001  1901
gamma[5]  -0.071  0.014  -0.099  -0.071  -0.043    FALSE 1.000   1.000  3000
gamma[6]   0.090  0.012   0.067   0.090   0.113    FALSE 1.000   1.000  3000
```

```
# Plot posterior distributions for potentially 'ecological' parameters
par(mfrow = c(1,3))
hist(out1J$sims.list$gamma[,1], breaks = 50, col = "grey", main = "", xlab =
"Slope of elevation (linear)")
hist(out1J$sims.list$gamma[,2], breaks = 50, col = "grey", main = "", xlab =
"Slope of elevation (squared)")
hist(out1J$sims.list$gamma[,3], breaks = 50, col = "grey", main = "", xlab =
"Slope of forest")
```

The evidence for linear and quadratic effects of elevation is strong, while that for an effect of forest cover is much less so. Let's inspect these relationships by plotting them (Figure 11.6, left and middle).

```
# Get covariate values for prediction
orig.pred.ele <- seq(250, 2750,, 500) # 500 vals spread between 250 and 2750
p.ele <- (orig.pred.ele - mean.ele) / sd.ele
orig.pred.forest <- seq(1, 100,, 500)
p.forest <- (orig.pred.forest - mean.forest) / sd.forest

# Compute predictions
nsamp <- out1J$mcmc.info$n.samples
pred.ele <- pred.forest <- array(NA, dim = c(500, nsamp))
for(i in 1:nsamp){
   pred.ele[,i] <- exp(out1J$sims.list$gamma0[i] + out1J$sims.list$gamma[i,1] *
p.ele + out1J$sims.list$gamma[i,2]* p.ele^2)
```

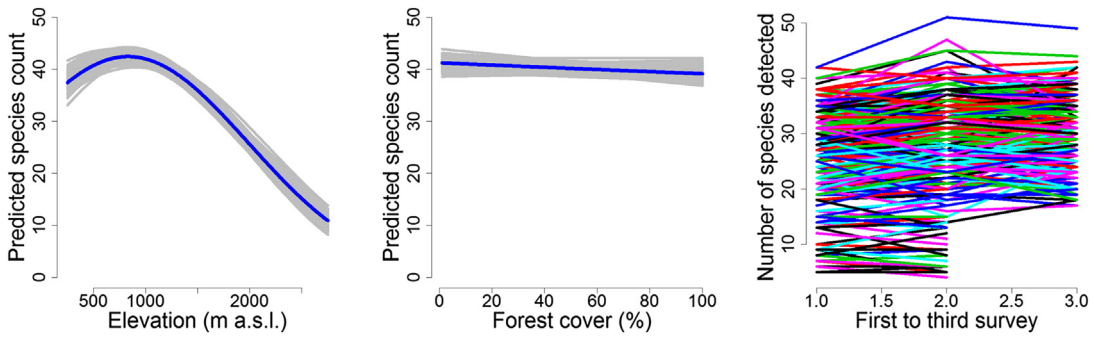

FIGURE 11.6

Left and middle: Inferences about the relationship between species richness and elevation and forest cover, respectively, in Switzerland in 2014 based on a simple Poisson GLM that does not retain species identity and ignores imperfect detection (blue line shows posterior mean and gray lines a random sample of size 100 of the posterior distribution of the regression equations). Right: Number of detected bird species per site and survey during the 2014 Swiss MHB survey.

```
 pred.forest[,i] <- exp(out1J$sims.list$gamma0[i] + out1J$sims.list$gamma[i,3]
* p.forest)
}

# Plot posterior mean and a random sample of 100 from posterior of regression
selection <- sample(1:nsamp, 100)
par(mfrow = c(1,3))
matplot(orig.pred.ele, pred.ele[,selection], ylab = "Predicted species count", xlab =
"Elevation (m a.s.l.)", type = "l", lty = 1, lwd = 1, col = "grey", ylim = c(0, 50), frame = F)
lines(orig.pred.ele, apply(pred.ele, 1, mean), lwd = 3, col = "blue")
matplot(orig.pred.forest, pred.forest[,selection], ylab = "Predicted species count",
xlab = "Forest cover (%)", type = "l", lty = 1, lwd = 1, col = "grey",
ylim = c(0, 50), frame = F)
lines(orig.pred.forest, apply(pred.forest, 1, mean), lwd = 3, col = "blue")
```

So, the response of the community to elevation seems clear: the largest communities are observed at low to medium elevations and the smallest at high elevations (Figure 11.6, left). There is only a very weak negative response of the community as a whole to forest cover in terms of the observed species richness (Figure 11.6, middle). Hence, we *know* that forest cover must be hugely significant on the occurrence patterns of many species and, yet, when we model the aggregate total we see nothing. That's all we have learned about the metacommunity of Swiss breeding birds, using a variant of what may be the most traditional approach to inference about the ecological determinants of the size of a community.

By simply taking the maximum species count over surveys, we obviously lost some information, especially about the relationship between counts and the two covariates that presumably may also affect the observed species richness via detection probability. So as an improvement over the current model, let's fit a similar model to the individual counts for each survey (Figure 11.6, right). This model is again a classical analysis that attempts to correct for measurement error (here associated with species richness N) by simply adding covariates that are believed, hoped, or claimed to be correlated with detection probability. We think that with longer survey duration, we see a larger proportion of the

species present. Survey date is also expected to be related to detection probability (see, for instance, our earlier analyses of abundance and distribution of Swiss MHB data in this book). In principle, in the case of an open community, *N* could also be related to survey date; i.e., species might move in or out during the entire sampling period, and so covariate *date* might capture part of the resulting variation. However, in accordance with the usual closure assumption that we are willing to make for MHB data within a single two-to-three-month breeding season, we will interpret effects of *date* in terms of variation in detection probability rather than in terms of variation in the true community size. We will tally up the number of species detected per site and survey and then prepare the observational covariates for survey date (*DAT*) and survey duration (*DUR*), which vary by site and survey only (but not by species); we scale both.

```
# Get observed species richness per site and rep and plot
CC <- apply(y, c(1,2), sum, na.rm = TRUE)
CC[CC == 0] <- NA              # 0 means not surveyed
 matplot(t(CC), type = 'l', lty = 1, lwd = 2, xlab = "First to third survey",
 ylab = "Number of species detected", frame = F)# Fig. 11-6 right

# Get survey date and survey duration and standardise both
# Survey date (this is Julian date, with day 1 being April 1)
orig.DAT <- cbind(data$date141, data$date142, data$date143)[1:nsite,]
(mean.date <- mean(orig.DAT, na.rm = TRUE))
(sd.date <- sd(c(orig.DAT), na.rm = TRUE))
DAT <- (orig.DAT - mean.date) / sd.date          # scale
DAT[is.na(DAT)] <- 0                             # mean-impute missings
# Survey duration (in minutes)
orig.DUR <- cbind(data$dur141, data$dur142, data$dur143)[1:nsite,]
(mean.dur <- mean(orig.DUR, na.rm = TRUE))
(sd.dur <- sd(c(orig.DUR), na.rm = TRUE))
DUR <- (orig.DUR - mean.dur) / sd.dur            # scale
DUR[is.na(DUR)] <- 0                             # mean-impute missings

# Bundle data and summarize
str( win.data <- list(CC = CC, M = nrow(CC), J = ncol(CC), ele = ele, forest =
forest, DAT = DAT, DUR = DUR) )
List of 7
 $ CC    : num [1:267, 1:3] 21 28 32 29 29 31 24 32 30 30 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : NULL
  .. ..$ : NULL
 $ M     : int 267
 $ J     : int 3
 $ ele   : num [1:267] -1.1539 -1.1539 -0.2175 -0.3735 -0.0614 ...
 $ forest: num [1:267] -1.1471 -0.4967 -0.0992 -0.9303  0.0092 ...
 $ DAT   : num [1:267, 1:3] -1.415 -1.19 -1.235 -0.559  -1.64 ...
 $ DUR   : num [1:267, 1:3] -0.43511 -0.78764 -0.52324 1.23944 0.00556 ...

# Specify model in BUGS language
sink("model2.txt")
cat("
model {
```

```
# Priors
gamma0 ~ dnorm(0, 0.001)
for(v in 1:6){
  gamma[v] ~ dnorm(0, 0.001)
}

# Likelihood for Poisson GLM
for (i in 1:M){                          # Loop over sites
  for(j in 1:J){                         # Loop over occasions
    CC[i,j] ~ dpois(lambda[i,j])
    log(lambda[i,j]) <- gamma0 + gamma[1] * ele[i] + gamma[2] * pow(ele[i],2) + gamma[3] *
    forest[i] + gamma[4] * DAT[i,j] + gamma[5] * pow(DAT[i,j],2) + gamma[6] * DUR[i,j]
  }
}
}
",fill = TRUE)
sink()

# Initial values
inits <- function() list(gamma0 = rnorm(1), gamma = rnorm(6))

# Parameters monitored
params <- c("gamma0", "gamma")

# MCMC settings
ni <- 6000 ; nt <- 4 ; nb <- 2000 ; nc <- 3

# Call WinBUGS from R (ART 1.7 min)
out2 <- bugs(win.data, inits, params, "model2.txt", n.chains = nc,
n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
bugs.dir, working.directory = getwd())

# Call JAGS from R (ART 0.6 min)
out2J <- jags(win.data, inits, params, "model2.txt", n.chains = nc, n.thin = nt,
n.iter = ni, n.burnin = nb)
traceplot(out2J) ; print(out2J, dig = 2)
```

	mean	sd	2.5%	50%	97.5%	overlap0	f	Rhat	n.eff
gamma0	3.44	0.01	3.42	3.44	3.47	FALSE	1.00	1	3000
gamma[1]	-0.19	0.01	-0.21	-0.19	-0.17	FALSE	1.00	1	1336
gamma[2]	-0.15	0.01	-0.17	-0.15	-0.13	FALSE	1.00	1	3000
gamma[3]	-0.01	0.01	-0.03	-0.01	0.00	TRUE	0.93	1	3000
gamma[4]	0.00	0.01	-0.01	0.00	0.02	TRUE	0.67	1	2352
gamma[5]	-0.03	0.01	-0.05	-0.03	-0.02	FALSE	1.00	1	3000
gamma[6]	0.10	0.01	0.09	0.10	0.12	FALSE	1.00	1	856

Now we see "significant" effects of elevation (linear and squared—`gamma[1]` and `gamma[2]`), but the effect of forest (`gamma[3]`) is still not "significant." Two of the three effects that we think are related with the measurement error of species richness rather than with species richness itself (i.e., date squared and survey duration—`gamma[5]` and `gamma[6]`) are "significant"—i.e., have 95% CRIs that do not overlap zero.

11.5.2 POISSON RANDOM-EFFECTS MODEL FOR THE OBSERVED COMMUNITY SIZE

The next model in our progression is a random-effects Poisson (REP) model that accounts for possible dependence of repeated counts at the same site by adopting normal random site effects on the log scale. Thus, while the previous model treated repeated species counts at the same site as independent (given the covariate values), the improvement made by the REP model is that it accommodates a possible correlation of counts made at the same site.

$$C_{ij} \sim Poisson(\lambda_{ij})$$
$$\log(\lambda_{ij}) = \gamma_{0,i} + \text{covariates}_{ij}$$
$$\gamma_{0,i} \sim Normal(mu.gamma0, sd.gamma0^2)$$

This is a simplified version of the REP models developed by Bill Link and John Sauer (e.g., Link and Sauer, 2002) for counts of individual species. Note also that there is a relationship between this Poisson/Normal mixture model and the Binomial/Poisson *N*-mixture model (Chapter 6; see also Dennis et al., 2015a).

```
# Bundle and summarize data set
str( win.data <- list(CC = CC, M = nrow(CC), J = ncol(CC), ele = ele, forest =
forest, DAT = DAT, DUR = DUR) )

# Specify model in BUGS language
sink("model3.txt")
cat("
model {

# Priors
mugamma0 ~ dnorm(0, 0.001)                # Hyperparameters
taugamma0 <- pow(sd.gamma0,-2)
sd.gamma0 ~ dunif(0, 10)
for(v in 1:6){                            # Parameters
  gamma[v] ~ dnorm(0, 0.001)
}

# Likelihood for Poisson GLMM
for (i in 1:M){                           # Loop over sites
  gamma0[i] ~ dnorm(mugamma0, taugamma0) # Site intercepts random now
  for(j in 1:J){                          # Loop over repeated measurements
    CC[i,j] ~ dpois(lambda[i,j])
    log(lambda[i,j]) <- gamma0[i] + gamma[1]*ele[i] + gamma[2] * pow(ele[i],2) +gamma[3] *
    forest[i] + gamma[4] * DAT[i,j] + gamma[5] * pow(DAT[i,j],2) + gamma[6] * DUR[i,j]
  }
}
}
",fill = TRUE)
sink()

# Initial values
inits <- function() list(gamma0 = rnorm(nrow(CC)), gamma = rnorm(6))
```

```
# Parameters monitored
params <- c("mugamma0", "sd.gamma0", "gamma0", "gamma")

# MCMC settings
ni <- 6000 ; nt <- 4 ; nb <- 2000 ; nc <- 3

# Call WinBUGS from R (ART 2.9 min)
out3 <- bugs(win.data, inits, params, "model3.txt", n.chains = nc,
n.thin = nt, n.iter = ni, n.burnin = nb, debug = TRUE, bugs.directory =
bugs.dir, working.directory = getwd())

# Call JAGS from R (ART 0.7 min)
out3J <- jags(win.data, inits, params, "model3.txt", n.chains = nc, n.thin = nt,
n.iter = ni, n.burnin = nb)
traceplot(out3J, c('mugamma0', 'sd.gamma0', 'gamma')) ;      print(out3J, dig = 2)
              mean    sd    2.5%    50%   97.5%  overlap0     f   Rhat   n.eff
mugamma0      3.43  0.02   3.38   3.43    3.47      FALSE  1.00   1.01     189
sd.gamma0     0.18  0.01   0.16   0.18    0.21      FALSE  1.00   1.00    3000
gamma0[1]     3.31  0.10   3.10   3.31    3.51      FALSE  1.00   1.00    3000
gamma0[2]     3.44  0.10   3.25   3.44    3.62      FALSE  1.00   1.00    1021
gamma0[3]     3.58  0.08   3.41   3.58    3.75      FALSE  1.00   1.00    3000
  [ ... output truncated .... ]
gamma0[265]  3.39  0.14  3.10   3.39    3.67      FALSE   1.00   1.00    3000
gamma0[266]  3.81  0.10  3.61   3.81    3.99      FALSE   1.00   1.00    3000
gamma0[267]  3.88  0.09  3.70   3.88    4.04      FALSE   1.00   1.00     608
gamma[1]    -0.21  0.02 -0.24  -0.21   -0.17      FALSE   1.00   1.00     415
gamma[2]    -0.18  0.02 -0.22  -0.18   -0.14      FALSE   1.00   1.01     278
gamma[3]     0.00  0.02 -0.03   0.00    0.03       TRUE   0.54   1.00    3000
gamma[4]     0.01  0.01 -0.01   0.01    0.03       TRUE   0.84   1.00    3000
gamma[5]    -0.02  0.01 -0.04  -0.02   -0.01      FALSE   1.00   1.00    3000
gamma[6]     0.10  0.01  0.08   0.10    0.13      FALSE   1.00   1.00    1454
```

We mostly present this model as one element in our progression to models that are more sophisticated in the context of metacommunities, so we don't say much about it here. But we note than when we account for the nonindependence of species counts from the same sites, the effect of forest (`gamma[3]`) is again clearly not "significant." We also note that JAGS is much faster than WinBUGS (0.7 vs. 2.9 minutes), so for the rest of the chapter we will fit models only in JAGS.

11.5.3 *N*-MIXTURE MODEL FOR OBSERVED COMMUNITY SIZE

A next step in our progression of increasingly sophisticated community models is a simple binomial *N*-mixture model. This model also accounts for the correlation of repeated measurements of species richness at a site. Moreover, it obtains an estimate of the average detection probability of a species from this correlation and thereby also produces an estimate of local community size: the number of species at each site. However, it does not allow for species heterogeneity in detection probability, nor for inference about individual species, because species identities are lost in this approach; we use only counts of "unmarked" species. As a consequence, we cannot estimate metacommunity size. Nevertheless, this model achieves an improved conceptual clarity by separating covariate effects on the community from covariates that affect its measurement only, and so we will return to our usual `alpha-beta` notation for the coefficients. We fit the following model to C_{ij}, the number of species recorded at site i during occasion j, where N_i is the community size (the number of species occurring) at site i.

$$
\begin{aligned}
C_{ij} &\sim Binomial(N_i, p_{ij}) \\
\text{logit}(p_{ij}) &= \alpha_0 + \text{covariates}_{ij} \\
N_i &\sim Poisson(\lambda_i) \\
\log(\lambda_i) &= \beta_0 + \text{covariates}_i
\end{aligned}
$$

Here, p_{ij} is the average detection probability of a species at site i during occasion j, and λ_i is the expected community size (i.e., local species richness) at site i. Under the closure assumption, the number of species occurring at a site does not change among repeated measurements of community size at site i. We index the detection and community richness covariates by ij and by i to say that they can be observation or site covariates for the former, but only site covariates for the latter.

```
# Bundle and summarize data set
str( win.data <- list(CC = CC, M = nrow(CC), J = ncol(CC), ele = ele, forest =
forest, DAT = DAT, DUR = DUR) )

# Specify model in BUGS language
sink("model4.txt")
cat("
model {

# Priors
alpha0 ~ dnorm(0, 0.01)        # Base-line community detection probability
beta0 ~ dnorm(0, 0.01)         # Base-line community size (number of species)
for(v in 1:3){
    alpha[v] ~ dnorm(0, 0.01)  # Covariate effects on detection
    beta[v] ~ dnorm(0, 0.01)   # Covariate effects on community size
}

# Likelihood
# Ecological model for true community size
for (i in 1:M){                # Loop over sites
  N[i] ~ dpois(lambda[i])      # Community size
   lambda[i] <- exp(beta0 + beta[1] * ele[i] + beta[2] * pow(ele[i],2) +
   beta[3] * forest[i])

  # Observation model for repeated measurements
  for (j in 1:J){                # Loop over occasions
     CC[i,j] ~ dbin(p[i,j], N[i])
     p[i,j] <- 1 / (1 + exp(-lp[i,j]))
     lp[i,j] <- alpha0 + alpha[1] * DAT[i,j] + alpha[2] * pow(DAT[i,j],2) +
     alpha[3] * DUR[i,j]
  # logit(p) = ... causes undefined real result in WinBUGS (but not JAGS)
  }
}
}
",fill = TRUE)
sink()
```

The total over all sites, `Ntotal` in our previous applications of this model in Chapter 6, is no longer a meaningful quantity. We do not keep track of which species contribute to the species total at each site, and this prevents us from adding up the species at all sites for an estimate of the total number of species in the metacommunity.

```
# Define function to generate random initial values
Nst <- apply(CC, 1, max, na.rm = TRUE) + 1
Nst[Nst == -Inf] <- max(Nst, na.rm = T) # Some nonzero val. for unsurv. sites
inits <- function() list(N = Nst, alpha0 = rnorm(1), alpha = rnorm(3), beta0 =
rnorm(1), beta = rnorm(3))

# Parameters monitored
params <- c("alpha0", "alpha", "beta0", "beta","N")

# MCMC settings
ni <- 6000 ; nt <- 4 ; nb <- 2000 ; nc <- 3

# Run JAGS from R (ART 1.5 min) in parallel, look at traceplots
#    and summarize posteriors
out4 <- jags(win.data, inits, params, "model4.txt",
   n.chains = nc, n.thin = nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
traceplot(out4, c('alpha0', 'alpha', 'beta0', 'beta')); print(out4, 3)
```

```
            mean     sd     2.5%      50%   97.5%  overlap0      f   Rhat  n.eff
alpha0     1.180  0.075    1.022    1.182   1.319     FALSE  1.000  1.006    503
alpha[1]   0.052  0.020    0.013    0.052   0.092     FALSE  0.994  1.000   3000
alpha[2]  -0.051  0.017   -0.083   -0.051  -0.017     FALSE  0.998  1.001   1565
alpha[3]   0.334  0.033    0.270    0.333   0.400     FALSE  1.000  1.000   3000
beta0      3.713  0.025    3.667    3.713   3.763     FALSE  1.000  1.000   3000
beta[1]   -0.200  0.013   -0.225   -0.199  -0.175     FALSE  1.000  1.004    491
beta[2]   -0.176  0.017   -0.208   -0.176  -0.143     FALSE  1.000  1.001   1413
beta[3]   -0.007  0.013   -0.032   -0.007   0.018      TRUE  0.720  1.001   1581
N[1]      34.904  2.072  31.000   35.000  39.000     FALSE  1.000  1.000   3000
N[2]      40.330  2.510  36.000   40.000  45.000     FALSE  1.000  1.000   3000
N[3]      50.092  2.247  46.000   50.000  55.000     FALSE  1.000  1.001   2803
[ ... output truncated ... ]
```

We can compare the main abundance parameter estimates from the REP regression with the *N*-mixture model.

```
print(tmp <- cbind(out3$summary[c(1, 270:272),c(1:3,7)], out4$summary[5:8,
c(1:3, 7)]), 3) # REP model: cols. 1-4; Nmix model: cols. 5-8
              mean      sd     2.5%   97.5%      mean      sd     2.5%    97.5%
mugamma0   3.43072  0.0234   3.3850   3.4770   3.71304  0.0249   3.6665   3.7633
gamma[1]  -0.21030  0.0165  -0.2431  -0.1780  -0.19952  0.0129  -0.2252  -0.1746
gamma[2]  -0.18101  0.0196  -0.2208  -0.1440  -0.17632  0.0167  -0.2078  -0.1430
gamma[3]   0.00196  0.0159  -0.0292   0.0328  -0.00709  0.0129  -0.0323   0.0184
```

The species richness intercept under the REP model is `exp(3.43) = 30.88` species, while that under the *N*-mixture model is exactly 10 more—i.e., `exp(3.71) = 40.85` species. This is surely not so surprising, since the REP model does not account for imperfect detection. Otherwise, we see qualitatively similar inferences about the abundance covariates. In particular, we don't see any effect of forest cover on estimated true species richness. We finish by plotting the estimates of local species richness and the observed species counts against elevation (Figure 11.7). Although we can't easily tell which estimate under the *N*-mixture model belongs to which observed value of species richness, we get the impression that the *N*-mixture model must underestimate species richness, since the observed number of species at a site is on average very similar to that estimated under the *N*-mixture model. We think it unlikely that we see all occurring species with only three surveys. Rather, it appears more likely that among-species heterogeneity in detection probability leads to an underestimation of the site-specific *N* in the *N*-mixture

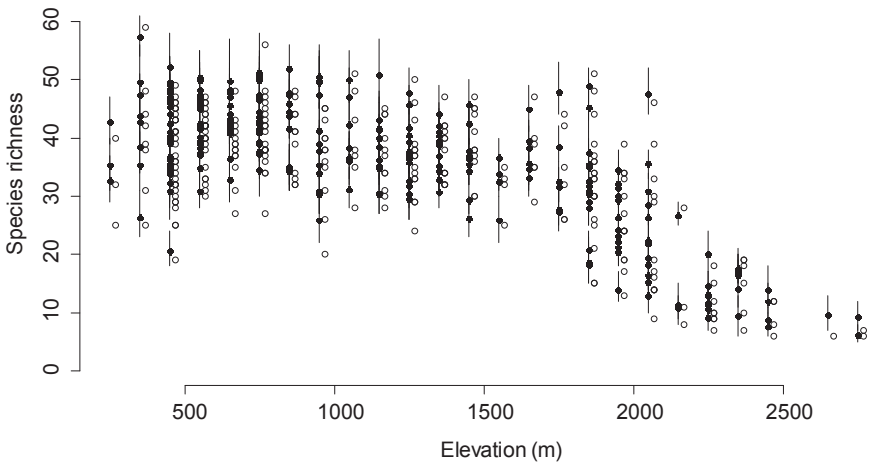

FIGURE 11.7

Estimated number of species at each of 267 sites in the Swiss breeding bird survey MHB under the binomial *N*-mixture model (filled circles, with 95% CRI) along the elevation profile from 200–2750 m a.s.l. The open circles show the *observed* number of species over the two or three surveys combined. Note counts are slightly offset for improved visibility.

model (per the second law of capture-recapture). In the *N*-mixture model, we can't accommodate such heterogeneity among species, and this is one of the main drawbacks of this approach.

```
plot(orig.ele, out4$summary[9:275, 1], pch = 16, xlab = "Elevation (m)", ylab =
"Species richness", ylim = c(0, 60), frame = F)
segments(orig.ele, out4$summary[9:275, 3], orig.ele, out4$summary[9:275, 7])
points(orig.ele+20, C)       # elevation slightly offset
```

Remember that instead of a binomial *N*-mixture model for site-specific species richness estimation, you could also adopt a multinomial *N*-mixture model (Chapter 7) for the *species* detection/nondetection data and model local species richness instead of local population abundance as in the more typical applications of this model in Chapter 7. In contrast to the binomial *N*-mixture model, species identity would only be lost across sites, but not across replicate surveys at each site. Hence, more information would be used, plus we could model individual detection heterogeneity, which is compulsory for species richness applications of closed population abundance estimators (Kéry 2011a). Importantly, we could then also model local species richness directly as a function of site covariates. A pure M_h type of such a model (i.e., with only species heterogeneity, but no other effect in detection probability) may even be fit in `unmarked` (see Section 7.8.3), where we could then test hypotheses about drivers of species richness using formal significance tests or AIC. When fit in BUGS, we could also model other structure into detection probability beyond individual (i.e., species-specific) heterogeneity, e.g., any of the detection covariates that we consider in this chapter. See Exercise 2 for an example of this.

11.6 COMMUNITY MODELS THAT FULLY RETAIN SPECIES IDENTITY

In the next set of models in our progression toward a powerful and satisfactory community modeling framework, we adopt a totally different approach and model the specific response to covariates of each member of the community. From now on, our community model will be a

collection of component models for the presence and absence of individual species. This will allow us to get much deeper insights into the metacommunity as well as into the local communities, and the ways in which they respond to environmental and other covariates as a function of individual species' responses. The species-specific approach to modeling the metacommunity will permit us to look at the average response of both the (meta)community and every one of its members. Remember our binary detection/nondetection indicators y_{ijk} for species k at site i during survey j; how can we retain species identities in an analysis of these data?

11.6.1 SIMPLEST COMMUNITY OCCUPANCY MODEL: *N*-FOLD SINGLE SPECIES OCCUPANCY MODEL WITH SPECIES TREATED AS FIXED EFFECTS

The first and simplest, but already very powerful, approach is to simply fit a separate species distribution model (SDM) to every species. Of course, we will use our favorite SDM, the occupancy model from Chapter 10. We could do this in a loop by fitting a separate occupancy model to the data from each of the 145 species in turn. However, it is *much more efficient* to fit these 145 models to the data of *all* species at once. This can easily be done in BUGS by organizing the data as a three-dimensional array (which we have already done). For illustration, we simply fit a null model to each species, with a species-specific constant intercept for both the occupancy and the detection probability, though we could readily add covariates. One big advantage of this approach over one that fits 145 separate models is that in the combined model, we can easily sum or compare quantities across species. For instance, it is trivial to add up the estimated number of occurring species for each site with a full propagation of all uncertainties involved in producing such a sum. As usual, the ease with which we can do computations on latent variables (here, the indicators of occurrence z) is one of the great practical benefits of a Bayesian analysis using MCMC. Similarly, we could constrain parameters among species, for instance by making them constant for some or all species or modeling them as a function of species-specific covariates (as we will see in Section 11.6.3).

We start the series of multispecies or community occupancy models without accounting for any structure among the three replicate surveys, so that we can collapse the detection/nondetection data to detection frequencies (which tell us whether a species was detected 0, 1, 2, or 3 times). We fit the following two-level HM to our detection frequency data $ysum_{ik}$ for species k ($k = 1\ldots145$) and site i ($i = 1\ldots267$).

$$\begin{aligned} &\text{Process model :} && z_{ik} \sim Bernoulli(\psi_k) \\ &\text{Observation model :} && ysum_{ik}|z_{ik} \sim Binomial(J_i, z_{ik}p_k) \end{aligned}$$

Here, J_i is the number of surveys and it is indexed by site, because at 47 high-elevation sites, there are two instead of the usual three surveys per breeding season. This corresponds *exactly* to the binomial version of the simplest possible occupancy model (see Section 10.3), except that we now stratify by species: every quantity is now also indexed by k, allowing it to be different for every species. No relationship among species is imposed on the species-specific parameters of occupancy (ψ_k) and detection probability (p_k), so their estimates will be identical to what you would get if you looped over all species and fitted a separate model to each.

We fit the binomial model to the aggregated data for illustrative purposes, mostly because this makes it easier to understand the basic structure of this model and the next few models in our progression of community models. It would be easy to fit the analogous models with a Bernoulli response for the disaggregated data y_{ijk}; see Exercise 3. You will see that the Bernoulli model takes about 50% longer to run for this data set.

```
# Collapse 3D detection/nondetection data to 2D detection frequencies
ysum <- apply(y, c(1,3), sum, na.rm = T)     # Collapse to detection frequency
ysum[NAsites,] <- NA                         # Have to NA out sites with NA data

# Bundle and summarize data set
str( win.data <- list(ysum = ysum, M = nrow(ysum), J = data$nsurvey[1:nsite],
nspec = dim(ysum)[2]) )

# Specify model in BUGS language
sink("model5.txt")
cat("
model {

# Priors
for(k in 1:nspec){                  # Loop over species
   psi[k] ~ dunif(0, 1)
   p[k] ~ dunif(0, 1)
}

# Ecological model for latent occurrence z (process model)
for(k in 1:nspec){                  # Loop over species
  for (i in 1:M) {                  # Loop over sites
      z[i,k] ~ dbern(psi[k])
  }
}

# Observation model for observed data ysum
for(k in 1:nspec){                  # Loop over species
   for (i in 1:M) {
      mup[i,k] <- z[i,k] * p[k]
      ysum[i,k] ~ dbin(mup[i,k], J[i])
   }
}

# Derived quantities
for(k in 1:nspec){                  # Loop over species
   Nocc.fs[k] <- sum(z[,k])         # Add up number of occupied sites among the 267
}
for (i in 1:M) {                    # Loop over sites
    Nsite[i] <- sum(z[i,])          # Add up number of occurring species at each site
}
}
",fill = TRUE)
sink()

# Initial values
zst <- apply(y, c(1,3), max)        # Observed occurrence as inits for z
zst[is.na(zst)] <- 1
inits <- function() list(z = zst, psi = rep(0.4, nspec), p = rep(0.4, nspec))

# Parameters monitored
params <- c("psi", "p", "Nsite", "Nocc.fs")
```

```
# MCMC settings
ni <- 2500 ; nt <- 2 ; nb <- 500 ; nc <- 3

# Call JAGS from R (ART 2.1 min)
out5 <- jags(win.data, inits, params, "model5.txt", n.chains = nc, n.thin = nt,
n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(4,4)) ; traceplot(out5) ; print(out5, dig = 3)
```

```
                mean     sd    2.5%     50%   97.5% overlap0 f   Rhat  n.eff
psi[1]         0.025  0.011   0.009   0.024   0.051    FALSE 1  1.000   3000
psi[2]         0.025  0.031   0.005   0.019   0.067    FALSE 1  1.211    159
psi[3]         0.022  0.011   0.007   0.020   0.047    FALSE 1  1.009   2555
psi[4]         0.008  0.007   0.001   0.007   0.025    FALSE 1  1.054    658
psi[5]         0.023  0.009   0.009   0.022   0.043    FALSE 1  1.000   3000
psi[6]         0.170  0.191   0.003   0.090   0.657    FALSE 1  1.059     69
  [ ... ]
psi[143]       0.035  0.021   0.011   0.031   0.086    FALSE 1  1.012   3000
psi[144]       0.124  0.020   0.087   0.123   0.167    FALSE 1  1.000   3000
psi[145]       0.026  0.017   0.007   0.023   0.070    FALSE 1  1.005   3000
p[1]           0.592  0.134   0.317   0.597   0.840    FALSE 1  1.001   2551
p[2]           0.417  0.181   0.097   0.418   0.760    FALSE 1  1.002   1007
p[3]           0.578  0.152   0.268   0.583   0.840    FALSE 1  1.000   3000
[ ... ]
p[143]         0.392  0.149   0.115   0.386   0.687    FALSE 1  1.001   1353
p[144]         0.734  0.049   0.633   0.737   0.822    FALSE 1  1.001   2022
p[145]         0.463  0.167   0.142   0.469   0.767    FALSE 1  1.000   3000
Nsite[1]      35.684  2.189  32.000  35.000  40.000    FALSE 1  1.050     45
Nsite[2]      37.697  2.201  34.000  38.000  42.000    FALSE 1  1.041     53
Nsite[3]      56.031  1.972  52.000  56.000  60.000    FALSE 1  1.027     78
  [ ... ]
Nsite[265]    20.881  2.741  16.000  21.000  27.000    FALSE 1  1.022     97
Nsite[266]    52.958  1.997  49.000  53.000  57.000    FALSE 1  1.066     35
Nsite[267]    55.543  1.939  52.000  55.000  60.000    FALSE 1  1.056     41
Nocc.fs[1]     5.851  1.363   5.000   5.000  10.000    FALSE 1  1.000    899
Nocc.fs[2]     5.710  7.773   3.000   4.000  16.000    FALSE 1  1.230    160
Nocc.fs[3]     4.898  1.782   4.000   4.000   9.000    FALSE 1  1.044   1032
[ ... ]
Nocc.fs[143]   8.249  4.749   5.000   7.000  21.000    FALSE 1  1.029    809
Nocc.fs[144]  32.172  1.212  31.000  32.000  35.000    FALSE 1  1.003    327
Nocc.fs[145]   6.038  3.510   4.000   5.000  15.000    FALSE 1  1.016   3000
```

In this analysis, species can be said to correspond to "fixed effects," so the parameter estimates for each species are exclusively determined by the data for that species. This means that the quality of estimates of species with very few observed presences will be pretty mediocre or even downright terrible; you can see this by the large posterior standard deviations and also the bad mixing of the chains for `psi` and `p` for some species. Nevertheless, one of the advantages of fitting this model with species strata is that we can compute derived quantities that are functions of the estimates for more than one species. One such example is the estimated number of species (among the list of 145 that were detected anywhere in 2014) occurring at a site, `Nsite`. Hence, as for the *N*-mixture model approach,

this analysis accommodates imperfect detection, and yields an estimate of site-specific species richness conditional on the list of species that were detected at least once. But in contrast to the *N*-mixture model, the community occupancy approach does *not* assume all species are identical. Rather, every species is allowed to differ in terms of *both* occupancy *and* detection probability. Thus, we would expect the resulting estimates of *N* to be greater than those under the binomial mixture model, which we argued were probably underestimates due to unmodeled individual detection heterogeneity (Boulinier et al., 1998; Cam et al., 2002b,c). We check this in a plot (Figure 11.8), where we moreover distinguish between sites surveyed twice versus three times. We see that indeed, all estimates of *N* are now clearly greater than the observed number of species (they lie above the 1:1 line), presumably because our new model better accounts for species-specific detection heterogeneity than did the *N*-mixture model. The second striking observation is that there are two sets of sites, and in one of them, the underestimation of species richness by the raw data is greater than in the other. This is of course the distinction of sites surveyed twice only: it is expected that a larger proportion of occurring species is missed.

```
# Compare observed and estimated site species richness
par(cex = 1.3)
twice <- which(data$nsurvey[1:267] == 2)
plot(C[twice], out5$summary[291:557,1][twice], xlab = "Observed number of
species", ylab = "Estimated number of species", frame = F, xlim = c(0, 60), ylim = c(0, 70),
col = "red", pch = 16)
segments(C[twice], out5$summary[291:557,3][twice], C[twice],
out5$summary[291:557,7][twice], col = "red")
points(C[-twice], out5$summary[291:557,1][-twice], col = "blue", pch = 16)
segments(C[-twice], out5$summary[291:557,3][-twice],
C[-twice], out5$summary[291:557,7][-twice], col = "blue")
abline(0,1)
```

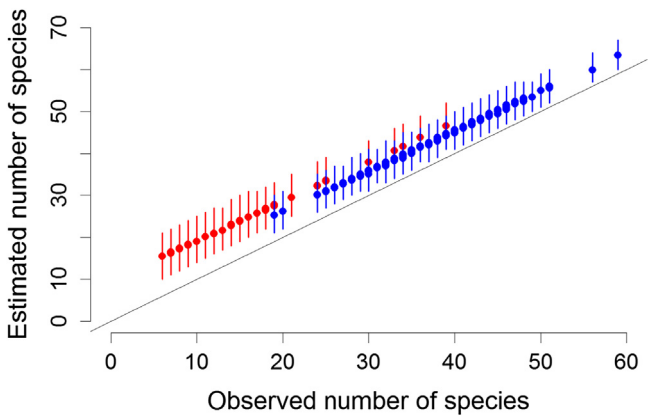

FIGURE 11.8

Comparison of observed and estimated number of species in 2014 at 267 sites under a community occupancy model with species treated as fixed effects (with 95% CRIs). The species richness estimate is conditional upon the list of 145 species that were detected at least once anywhere during the breeding season 2014 among the MHB sample quadrats. Blue: sites with three visits, red: sites with only two visits.

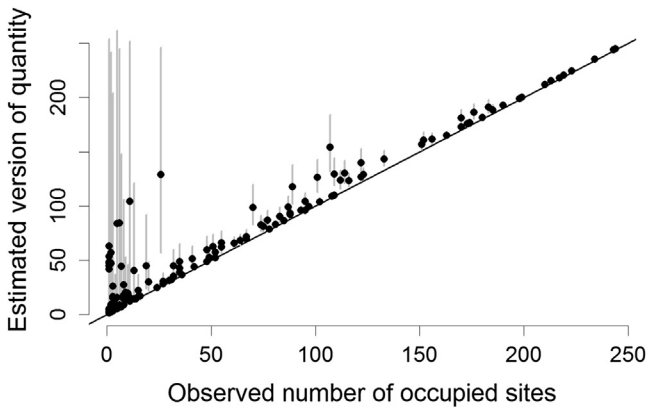

FIGURE 11.9

Observed and estimated number of sites (out of 267) where a species occurred in 2014 (with 95% CRIs).

We can also compare the observed and estimated number of sites where each species occurs (Figure 11.9): not surprisingly, when a species is detected at only few sites, the estimates become much more imprecise.

```
# Observed and estimated number of occupied sites for each species
# in a table
cbind(obs.occu = obs.occ, out5$summary[558:702, c(1,3,7)])
                              obs.occu        mean    2.5%    97.5%
Little Grebe                         5    5.850667   5.000   10.000
Great Crested Grebe                  3    5.709667   3.000   16.000
Grey Heron                           4    4.898333   4.000    9.000
White Stork                          1    1.277667   1.000    3.025
Mute Swan                            5    5.145000   5.000    6.000
Greylag Goose                        1   44.787667   1.000  176.000
Mallard                             55   61.984000  56.975   70.000
Common Merganser                     1    1.188333   1.000    3.000
European Honey Buzzard               5   83.990667   8.000  262.000
Red Kite                            95  103.973000  98.000  112.000
[ ... output truncated ... ]
```

```
# and in a plot
plot(obs.occ, out5$summary[558:702, 1], xlab = "Observed number of occupied sites",
ylab = "Estimated version of quantity", ylim = c(0, 267), frame = F, pch = 16)
abline(0,1)
segments(obs.occ, out5$summary[558:702,3], obs.occ, out5$summary[558:702,7], col
= "grey", lwd = 2)
```

Finally, we want to compare the estimates of detection probability with those for occupancy probability for all 145 species under model 5 (Figure 11.10).

```
# Estimated occupancy and detection probability for each species (model 5)
plot(out5$summary[1:145,1], out5$summary[146:290,1], xlab = "Occupancy estimate",
ylab = "Detection estimate", xlim = c(0,1), ylim = c(0,1), frame = F, pch = 16)
```

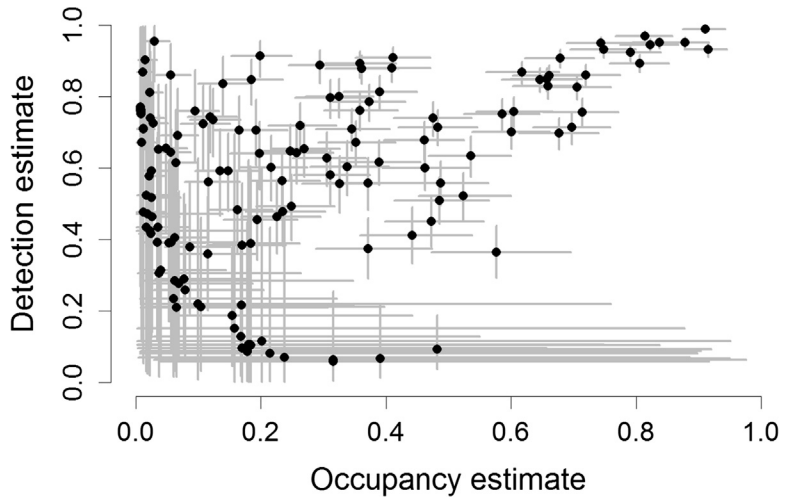

FIGURE 11.10

Estimates of detection and occupancy probabilities for the 145 species detected in the MHB in 2014 (with 95% CRIs).

```
segments(out5$summary[1:145,3], out5$summary[146:290,1], out5$summary[1:145,7],
out5$summary[146:290,1], col = "grey", lwd = 2)
segments(out5$summary[1:145,1], out5$summary[146:290,3], out5$summary[1:145,1],
out5$summary[146:290,7], col = "grey", lwd = 2)
```

Naturally, with small sample sizes, estimates become worse in the sense of having (much) wider 95% CRIs. Here, in what is exactly equivalent to independent single-species occupancy models fitted to 145 species, we see that when occupancy is smaller than about 0.1 and detection smaller than about 0.2, estimates become very imprecise and likely also biased, as is well known in the occupancy literature (MacKenzie et al., 2002; Guillera-Arroita et al., 2010). We will next see how we can improve by not estimating the parameters for all species independently (by treating species as a fixed effects factor), but rather treating the observed species as a sample from a larger statistical population of species—i.e., by treating species as random effects sharing hyperparameters that we estimate.

11.6.2 COMMUNITY OCCUPANCY MODEL WITH BIVARIATE SPECIES-SPECIFIC RANDOM EFFECTS

In this analysis, we treat the parameters of each species as random effects. This means that we assume that species-specific effects are drawn from a common distribution, called a prior distribution, with hyperparameters that we estimate. Under the assumption that the species are exchangeable ("similar but not identical"), the random-effects assumption will typically lead to improved estimates for individual species, in the sense of reducing prediction error or uncertainty intervals. The former is not something that we can prove or observe for real data because we don't know the truth; instead, we would have to check that with simulated data. However, the latter is easily shown (Kéry and Royle, 2008; Zipkin et al., 2009).

Our new model is almost exactly the same as before, except that now the species-specific parameters will be constrained by a common prior distribution. All other things, for instance the

finite-sample number of occupied quadrats or species richness (among our list of 145) can be computed in exactly the same way as in the previous analysis where species were treated as fixed effects. In the previous analysis, there was no need to apply a link function to the parameters, because the *Uniform*(0,1) priors for each parameter naturally enforced the usual range constraint for probability parameters. However, now that we make the random-effects assumption, we make the conventional assumption of normal priors for the parameters on the logit scale.

We could fit the following three-level HM, where the species random effects add another level to our HM.

$$\begin{aligned}
&\text{Process model :} && z_{ik} \sim Bernoulli(\psi_k)\\
&\text{Observation model :} && ysum_{ik}|z_{ik} \sim Binomial(J_i, z_{ik}p_k)\\
&\text{Models of species heterogeneity :} && \text{logit}(\psi_k) \sim Normal\left(\mu_{lpsi}, \sigma^2_{lpsi}\right)\\
&&& \text{logit}(p_k) \sim Normal\left(\mu_{lp}, \sigma^2_{lp}\right)
\end{aligned}$$

The species-specific parameters are no longer estimated independently; rather, they are constrained by the assumption of a common normal prior distribution for the logits of the occupancy and detection probabilities. It is the last two equations that declare species effects to be random. Also note that the species index k still runs from 1 to 145, which is the number of observed species. In this model, the pairs of values $[\text{logit}(\psi_k), \text{logit}(p_k)]$ for a species are assumed to be independent; i.e., there is no *a priori* assumption of any relationship between the two. However, as Dorazio et al. (2006) have argued, we may well assume that there is such a relationship induced by the underlying abundance of a species: more-common species are expected to be both more widespread and more detectable. This would lead to a positive correlation between occupancy and detection probabilities. We can actually see something of this in Figure 11.10 (though perhaps a triangular relationship better describes the results): very widespread species always seem to have high values of p, while many species that are not widespread exhibit small and in other cases large values of detection probability.

If we want to quantify an association between two parameters, we can assume a multivariate normal distribution for them instead of two independent normal distributions, and this has been done in many applications of multispecies occupancy models (e.g., Dorazio et al., 2006; Kéry and Royle, 2008, 2009). The final two lines in the above model then become this:

$$\text{Models of species heterogeneity :} \quad (lpsi_k, lp_k) \sim MVN\left(\begin{pmatrix}\mu_{lpsi}\\ \mu_{lp}\end{pmatrix}, \begin{pmatrix}\sigma^2_{lpsi} & \sigma_{lp*lpsi}\\ \sigma_{lp*lpsi} & \sigma^2_{lp}\end{pmatrix}\right)$$

This model has one additional parameter, the covariance $\sigma_{lp*lpsi}$. The covariance can be expressed as $\rho\sigma_{lpsi}\sigma_{lp}$—i.e., as the product of the correlation coefficient ρ and the standard deviations of the logit-scale parameters *lpsi* and *lp*. This is the parameterization that we had chosen in Kéry and Royle (2008); see also Chapter 12 in Kéry (2010). However, here we adapt code for covariance parameters from Chapter 7 in Kéry and Schaub (2012); also see Cam et al. (2002a). For a version of the model with two univariate normals and therefore a zero covariance, see Section 11.6.3 (where we add "species group" effects).

```
# Bundle and summarize data set
str( win.data <- list(ysum = ysum, M = nrow(ysum), J = data$nsurvey[1:nsite],
nspec = dim(ysum)[2], R = matrix(c(5,0,0,1), ncol = 2), df = 3) )

# Specify model in BUGS language
sink("model6.txt")
cat("
model {
```

```
# Priors
for(k in 1:nspec){    # Group lpsi and lp together in array eta
   lpsi[k] <- eta[k,1]
   lp[k] <- eta[k,2]
   eta[k, 1:2] ~ dmnorm(mu.eta[], Omega[,])
}
# Hyperpriors
# Priors for mu.lpsi=mu.eta[1] and mu.lp=mu.eta[2]
# probs = community means of occupancy and detection probability
for(v in 1:2){
   mu.eta[v] <- log(probs[v] / (1-probs[v]))
   probs[v] ~ dunif(0,1)
}
# Prior for variance-covariance matrix
Omega[1:2, 1:2] ~ dwish(R[,], df)
Sigma[1:2, 1:2] <- inverse(Omega[,])

# Ecological model for latent occurrence z (process model)
for(k in 1:nspec){
   logit(psi[k]) <- lpsi[k]      # Must take outside of i loop (b/c only indexed k)
   for (i in 1:M) {
      z[i,k] ~ dbern(psi[k])
   }
}

# Observation model for observed data ysum
for(k in 1:nspec){
   logit(p[k]) <- lp[k]          # Needs to be outside of i loop
   for (i in 1:M){
      mu.p[i,k] <- z[i,k] * p[k]
      ysum[i,k] ~ dbin(mu.p[i,k], J[i])
   }
}

# Derived quantities
rho <- Sigma[1,2] / sqrt(Sigma[1,1] * Sigma[2,2])     # Correlation coefficient
for(k in 1:nspec){
   Nocc.fs[k] <- sum(z[,k])      # Number of occupied sites among the 267
}
for (i in 1:M) {
   Nsite[i] <- sum(z[i,])        # Number of occurring species
}
}
",fill = TRUE)
sink()

# Initial values
zst <- apply(y, c(1,3), max) # Observed occurrence as starting values for z
zst[is.na(zst)] <- 1
inits <- function() list(z = zst, Omega = matrix(c(1,0,0,1), ncol = 2), eta =
matrix(0, nrow = nspec, ncol = 2))
```

```
# Parameters monitored
params <- c("mu.eta", "probs", "psi", "p", "Nsite", "Nocc.fs", "Sigma", "rho")

# MCMC settings
ni <- 20000 ; nt <- 15 ; nb <- 5000 ; nc <- 3

# Call JAGS from R (ART 12 min), check traceplots and summarize posteriors
out6 <- jags(win.data, inits, params, "model6.txt", n.chains = nc, n.thin = nt, n.iter = ni,
n.burnin = nb, parallel = TRUE)
par(mfrow = c(3,3)) ; traceplot(out6, c('mu.eta', 'probs', 'Sigma', 'rho'))
print(out6, 3)
```

	mean	sd	2.5%	50%	97.5%	overlap0	f	Rhat	n.eff
mu.eta[1]	-1.889	0.176	-2.230	-1.891	-1.535	FALSE	1	1.000	3000
mu.eta[2]	0.536	0.116	0.305	0.536	0.760	FALSE	1	1.000	3000
probs[1]	0.133	0.020	0.097	0.131	0.177	FALSE	1	1.000	3000
probs[2]	0.630	0.027	0.576	0.631	0.681	FALSE	1	1.000	3000
psi[1]	0.024	0.010	0.009	0.023	0.047	FALSE	1	1.001	3000
psi[2]	0.018	0.010	0.005	0.016	0.045	FALSE	1	1.009	2854
psi[3]	0.020	0.009	0.007	0.018	0.042	FALSE	1	1.001	2228
[...]									
psi[143]	0.027	0.013	0.010	0.025	0.056	FALSE	1	1.009	1138
psi[144]	0.121	0.020	0.084	0.120	0.162	FALSE	1	1.000	3000
psi[145]	0.022	0.011	0.007	0.020	0.045	FALSE	1	1.001	3000
p[1]	0.570	0.125	0.320	0.574	0.802	FALSE	1	1.000	3000
p[2]	0.437	0.148	0.162	0.431	0.744	FALSE	1	1.001	1681
p[3]	0.562	0.132	0.305	0.568	0.806	FALSE	1	1.002	1012
[...]									
p[143]	0.431	0.128	0.188	0.429	0.683	FALSE	1	1.000	3000
p[144]	0.735	0.049	0.636	0.737	0.827	FALSE	1	1.001	1581
p[145]	0.474	0.137	0.214	0.472	0.737	FALSE	1	1.001	2019
Nsite[1]	32.007	1.393	30.000	32.000	35.000	FALSE	1	1.001	1705
Nsite[2]	34.023	1.359	32.000	34.000	37.000	FALSE	1	1.001	1064
Nsite[3]	52.689	1.229	51.000	53.000	55.000	FALSE	1	1.001	3000
[...]									
Nsite[265]	17.135	2.181	13.000	17.000	22.000	FALSE	1	1.000	3000
Nsite[266]	49.537	1.211	48.000	49.000	52.000	FALSE	1	1.000	3000
Nsite[267]	52.223	1.086	51.000	52.000	55.000	FALSE	1	1.001	3000
Nocc.fs[1]	5.829	1.230	5.000	5.000	9.000	FALSE	1	1.003	2650
Nocc.fs[2]	4.331	1.986	3.000	4.000	10.000	FALSE	1	1.007	3000
Nocc.fs[3]	4.759	1.222	4.000	4.000	8.000	FALSE	1	1.007	1294
[...]									
Nocc.fs[143]	6.894	2.450	5.000	6.000	13.000	FALSE	1	1.017	3000
Nocc.fs[144]	32.102	1.181	31.000	32.000	35.000	FALSE	1	1.000	3000
Nocc.fs[145]	5.292	1.870	4.000	5.000	10.000	FALSE	1	1.003	3000
Sigma[1,1]	4.309	0.550	3.350	4.258	5.548	FALSE	1	1.000	3000
Sigma[2,1]	1.462	0.285	0.937	1.450	2.056	FALSE	1	1.000	3000
Sigma[1,2]	1.462	0.285	0.937	1.450	2.056	FALSE	1	1.000	3000
Sigma[2,2]	1.522	0.224	1.143	1.499	2.021	FALSE	1	1.000	3000
rho	0.571	0.078	0.408	0.576	0.707	FALSE	1	1.000	3000

We see that the species heterogeneity in the logit transform of detection probability (`sd.lp`) is estimated at 1.522, while for occupancy probability (`sd.lpsi`) it is estimated at a much larger value of 4.309, and the correlation between the two logit-scale parameters is estimated at 0.57.

```
# Graphically compare some estimates between fixed- and random-effects model
par(mfrow = c(2,2))       # not shown
# Species-specific occupancy (probability scale)
plot(out5$summary[1:145,1], out6$summary[5:149,1], main = "Species-specific occupancy
probability") ;                abline(0,1)
# Species-specific detection (probability scale)
plot(out5$summary[146:290,1], out6$summary[150:294,1], main = "Species-specific
detection probability") ;      abline(0,1)
# Site-specific species richness
plot(out5$summary[291:557,1], out6$summary[295:561,1], main = "Site-specific
species richness (conditional on list of 145 detected)") ;        abline(0,1)
# Species-specific number of presences
plot(out5$summary[558:702,1], out6$summary[562:706,1], main = "Species-specific number
of presences (in 267 sites)") ;              abline(0,1)
```

For the most part (i.e., for species with at least moderate sample size), point estimates are fairly similar under the two models. However, for species with relatively sparse data, the estimates are substantially more precise, and there can be considerable shifts in the posterior means; this is the "shrinkage" effect of Bayesian estimators at work. Finally, we want to compare the estimates of detection probability with those of occupancy probability for all 145 species under model 6 (Figure 11.11). The length of the 95% CRI appears reduced overall compared with those under model 5, which did not share information among the species via the random-effects assumption.

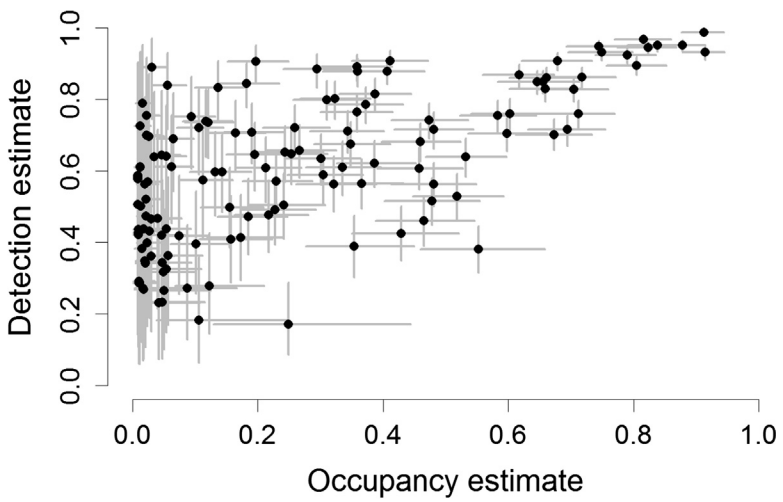

FIGURE 11.11

Estimates of detection probability and occupancy probability for the 145 species detected in the MHB in 2014 (with 95% CRI) under random-effects model 6.

```
# Estimated occupancy and detection probability for each species
plot(out6$summary[5:149,1], out6$summary[150:294,1], xlab = "Occupancy estimate",
ylab = "Detection estimate", xlim = c(0,1), ylim = c(0,1), frame = F, pch = 16)
segments(out6$summary[5:149,3], out6$summary[150:294,1], out6$summary[5:149,7],
out6$summary[150:294,1], col = "grey", lwd = 2)
segments(out6$summary[5:149,1], out6$summary[150:294,3], out6$summary[5:149,1],
out6$summary[150:294,7], col = "grey", lwd = 2)
```

11.6.3 MODELING SPECIES-SPECIFIC EFFECTS IN COMMUNITY OCCUPANCY MODELS

Frequently, we want to compare groups of species—for instance, guilds (e.g., herbivores vs carnivores), life-history categories (slow vs fast species), migration modes (resident vs migratory species), or other distinctions between species. We then treat species again as random effects, but specify a linear model for the hyperparameters that govern the species-specific occupancy and detection parameters. We illustrate this here. In our data file, there are data on three traits that characterize species: body length (cm), body mass (g), and wingspan (cm). As a first example for the modeling of species-specific effects, we will fit a community occupancy model with separate parameters for three species size groups that we define by body mass. We try to obtain three groups with roughly similar numbers of species, and take as cut points 1.3 and 2.6 for the log10 of body mass; this corresponds to about 20 and 398 g. One species with 11-kg body mass (the mute swan) is assigned to group 3, too.

Modeling species effects in such groups corresponds to an ANOVA-type of linear model for species-specific effects in the community model. In our second example below, we will illustrate the corresponding "regression-type" of linear model—i.e., one that models the effects of a continuous covariate. Clearly, when you understand how to fit linear models in the BUGS language, you could easily combine the two types of effects (see Chapters 3 and 5)—for instance, if you had effects of body size (continuous) and a binary factor such as whether species are sexually dimorphic (Doherty et al., 2003).

```
# Look at distribution of body mass among 136 observed species
mass <- tapply(data$body.mass, data$specid, mean) # Get mean species mass
hist(log10(mass), breaks = 40, col = "grey")      # Look at log10
gmass <- as.numeric(log10(mass) %/% 1.3 + 1)      # Size groups 1, 2 and 3
gmass[gmass == 4] <- 3                            # Mute swan is group 3, too
```

We will index the three size groups as $g = 1, 2, 3$ and could adopt any one of the following three models, which differ only in the structure assumed for species heterogeneity:

Interspecific Comparison 1: Group Effects for the Means Only

$$
\begin{aligned}
&\text{Process model :} && z_{ik} \sim Bernoulli(\psi_k) \\
&\text{Observation model :} && ysum_{ik} | z_{ik} \sim Binomial(J_i, z_{ik} p_k) \\
&\text{Models of species heterogeneity :} && \text{logit}(\psi_k) \sim Normal\left(\mu_{lpsi}[g], \sigma^2_{lpsi}\right) \\
& && \text{logit}(p_k) \sim Normal\left(\mu_{lp}[g], \sigma^2_{lp}\right)
\end{aligned}
$$

Interspecific Comparison 2: Group Effects for Both Means and Variances
(same process and observation models)

$$\text{Models of species heterogeneity}: \quad \text{logit}(\psi_k) \sim Normal\left(\mu_{lpsi}[g], \sigma^2_{lpsi}[g]\right)$$
$$\text{logit}(p_k) \sim Normal\left(\mu_{lp}[g], \sigma^2_{lp}[g]\right)$$

By the *g* within square brackets, we are saying that we estimate a different parameter for every group. Model 1 only distinguishes between the groups in terms of the mean of the intercepts for occupancy and detection probability, while model 2 also allows the *variances* to differ between the groups. So, this is another rare example where we explicitly model a variance parameter as a function of a covariate (see also Section 6.11.2.2 for an example in the context of a binomial mixture model). For illustration, we here fit model 2 (model 1 is a simple restriction of model 2). In this model, the only changes to a model without group effects (e.g., model 6) happen in the prior and hyperprior sections of the BUGS model description.

```
# Bundle and summarize data set
str( win.data <- list(ysum = ysum, g = gmass, M = nrow(ysum), J =
data$nsurvey[1:nsite], nspec = dim(ysum)[2]) )

# Specify model in BUGS language
sink("model7.txt")
cat("
model {

# Priors: note group effects specified in this section
for(k in 1:nspec){       # loop over species
   lpsi[k] ~ dnorm(mu.lpsi[g[k]], tau.lpsi[g[k]])   # note g-dependence now
   lp[k] ~ dnorm(mu.lp[g[k]], tau.lp[g[k]])
}

# Hyperpriors
for(g in 1:3){           # loop over 3 groups (g)
  mu.lpsi[g] <- logit(mu.psi[g])       # everything is indexed g now
  mu.lp[g] <- logit(mu.p[g])
  mu.psi[g] ~ dunif(0,1)
  mu.p[g] ~ dunif(0,1)
  tau.lpsi[g] <- pow(sd.lpsi[g], -2)
  sd.lpsi[g] ~ dunif(0,5)
  tau.lp[g] <- pow(sd.lp[g], -2)
  sd.lp[g] ~ dunif(0,5)
}

# Ecological model for latent occurrence z (process model)
for(k in 1:nspec){                 # no change at all down here in model
   logit(psi[k]) <- lpsi[k]
   for (i in 1:M) {
      z[i,k] ~ dbern(psi[k])
   }
}
```

```
# Observation model for observed data ysum
for(k in 1:nspec){      # Loop over species
   logit(p[k]) <- lp[k]
   for (i in 1:M) {
      mu.px[i,k] <- z[i,k] * p[k]  # call mu.px to avoid conflict with above
      ysum[i,k] ~ dbin(mu.px[i,k], J[i])
   }
}

# Derived quantities
for(k in 1:nspec){             # Loop over species
   Nocc.fs[k] <- sum(z[,k])  # Number of occupied sites among the 267
}
for (i in 1:M) {               # Loop over sites
    Nsite[i] <- sum(z[i,])    # Number of occurring species at each site
}
}
",fill = TRUE)
sink()

# Initial values
zst <- apply(y, c(1,3), max)
zst[is.na(zst)] <- 1
inits <- function() list(z = zst)

# Parameters monitored
params <- c("mu.psi", "mu.lpsi", "sd.lpsi", "mu.p", "mu.lp", "sd.lp")

# MCMC settings
ni <- 6000 ; nt <- 2 ; nb <- 2000 ; nc <- 3

# Call JAGS from R (ART 6 min), look at convergence and summarize posteriors
out7 <- jags(win.data, inits, params, "model7.txt", n.chains = nc, n.thin = nt,
n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(3,3)) ; traceplot(out7)
print(out7, dig = 3)
```

	mean	sd	2.5%	50%	97.5%	overlap0	f	Rhat	n.eff
mu.psi[1]	0.240	0.063	0.134	0.235	0.375	FALSE	1.000	1.001	2417
mu.psi[2]	0.132	0.028	0.082	0.129	0.192	FALSE	1.000	1.000	6000
mu.psi[3]	0.080	0.031	0.035	0.074	0.151	FALSE	1.000	1.006	347
mu.lpsi[1]	-1.183	0.349	-1.865	-1.181	-0.511	FALSE	0.999	1.001	2291
mu.lpsi[2]	-1.909	0.246	-2.413	-1.905	-1.439	FALSE	1.000	1.000	6000
mu.lpsi[3]	-2.517	0.407	-3.305	-2.523	-1.725	FALSE	1.000	1.006	316
sd.lpsi[1]	2.200	0.275	1.734	2.174	2.820	FALSE	1.000	1.001	2656
sd.lpsi[2]	2.133	0.195	1.794	2.118	2.553	FALSE	1.000	1.000	6000
sd.lpsi[3]	1.871	0.323	1.350	1.835	2.613	FALSE	1.000	1.000	6000
mu.p[1]	0.750	0.033	0.680	0.752	0.810	FALSE	1.000	1.000	6000
mu.p[2]	0.666	0.036	0.591	0.667	0.735	FALSE	1.000	1.000	6000
mu.p[3]	0.412	0.095	0.237	0.410	0.598	FALSE	1.000	1.003	759

```
mu.lp[1]      1.108   0.178   0.753   1.110   1.453    FALSE 1.000 1.000 6000
mu.lp[2]      0.693   0.165   0.369   0.693   1.019    FALSE 1.000 1.000 6000
mu.lp[3]     -0.369   0.407  -1.169  -0.364   0.396     TRUE 0.822 1.003  723
sd.lp[1]      1.031   0.136   0.801   1.018   1.329    FALSE 1.000 1.001 2990
sd.lp[2]      1.292   0.147   1.041   1.280   1.617    FALSE 1.000 1.001 3446
sd.lp[3]      1.903   0.409   1.262   1.851   2.835    FALSE 1.000 1.004  471
```

We note some nice patterns along the body mass gradient in every hyperparameter. The average occupancy probability of small birds (in size class 1) is 0.24, and then drops to 0.13 and 0.08 for species in size classes 2 and 3, respectively. The interspecific variability in occupancy probability follows a similar pattern, as suggested by the estimates for `sd.lpsi`. The average detection probability decreases with increasing average body mass of a species (you see this in `mu.p`), while the interspecific variability in detection probability seems to increase with mean body mass (look at `sd.lp`).

Perhaps you feel that binning body mass is artificial. Of course, you can easily fit a model where body mass is treated as a continuous covariate, thus avoiding the categorizing of species. We next fit a linear regression of the hyperparameters governing species-specific values of occupancy and detection probability on the *continuous* mass covariate (which we log-transform). This is model 8, and it looks like this (the only change to the previous model is again in the part specifying the nature of species heterogeneity).

Interspecific Comparison 3: Linear Regressions for Means and Variances

$$
\begin{aligned}
\text{Models of species heterogeneity :} \quad & \text{logit}(\psi_k) \sim Normal\left(\mu_{lpsi,k}, \sigma^2_{lpsi,k}\right) \\
& \text{logit}(p_k) \sim Normal\left(\mu_{lp,k}, \sigma^2_{lp,k}\right) \\
& \mu_{lpsi,k} = delta0.lpsi + delta1.lpsi * \log(mass_k) \\
& \mu_{lp,k} = delta0.lp + delta1.lp * \log(mass_k) \\
& \log\left(\sigma^2_{lpsi,k}\right) = phi0.lpsi + phi1.lpsi * \log(mass_k) \\
& \log\left(\sigma^2_{lp,k}\right) = phi0.lp + phi1.lp * \log(mass_k)
\end{aligned}
$$

Now, each of the four hyperparameters that govern the two species traits (occupancy and detection probability) is indexed by species (k), and we simply add to our model four linear regressions on the log10 of body mass. To enforce the range constraint for a variance, we apply the linear model for variances with a log link function. Now, the "top-level" parameters in our HM (the hyper-hyperparameters) are the four intercepts and the four slopes. To keep our notation for covariate effects tidy, we now call these hypercoefficients `delta` and `phi`.

```
# Bundle and summarize data set
logmass <- as.numeric(log10(mass))       # Take log10 of body mass
str( win.data <- list(ysum = ysum, logmass = logmass, M = nrow(ysum),
J = data$nsurvey[1:nsite], nspec = dim(ysum)[2]) )

# Specify model in BUGS language
sink("model8.txt")
cat("
model {
```

```
# Priors
for(k in 1:nspec){                  # loop over species
   lpsi[k] ~ dnorm(mu.lpsi[k], tau.lpsi[k])      # now all indexed by k, not g
   tau.lpsi[k] <- 1/var.lpsi[k]
   lp[k] ~ dnorm(mu.lp[k], tau.lp[k])
   tau.lp[k] <- 1/var.lp[k]
   mu.lpsi[k] <- delta0.lpsi + delta1.lpsi * logmass[k]
   mu.lp[k] <- delta0.lp + delta1.lp * logmass[k]
   log(var.lpsi[k]) <- phi0.lpsi + phi1.lpsi * logmass[k]
   log(var.lp[k]) <- phi0.lp + phi1.lp * logmass[k]
}
# Priors for regression params for means
delta0.lpsi ~ dnorm(0, 0.01)
delta1.lpsi ~ dnorm(0, 0.01)
delta0.lp ~ dnorm(0, 0.01)
delta1.lp ~ dnorm(0, 0.01)
# Priors for regression params for variances
phi0.lpsi ~ dnorm(0, 0.01)
phi1.lpsi ~ dnorm(0, 0.01)
phi0.lp ~ dnorm(0, 0.01)
phi1.lp ~ dnorm(0, 0.01)

# Ecological model for latent occurrence z (process model)
for(k in 1:nspec){
   logit(psi[k]) <- lpsi[k]
   for (i in 1:M) {
      z[i,k] ~ dbern(psi[k])
   }
}

# Observation model for observed data ysum
for(k in 1:nspec){                  # Loop over species
   logit(p[k]) <- lp[k]
   for (i in 1:M) {
      mu.p[i,k] <- z[i,k] * p[k]
      ysum[i,k] ~ dbin(mu.p[i,k], J[i])
   }
}

# Derived quantities
for(k in 1:nspec){             # Loop over species
   Nocc.fs[k] <- sum(z[,k])    # Number of occupied sites among the 267
}
for (i in 1:nsite) {           # Loop over sites
   Nsite[i] <- sum(z[i,])      # Number of occurring species at each site
}
}
",fill = TRUE)
sink()
```

```
# Initial values
zst <- apply(y, c(1,3), max)
zst[is.na(zst)] <- 1
inits <- function() list(z = zst, delta0.lpsi = rnorm(1), delta1.lpsi = rnorm(1),
   delta0.lp = rnorm(1), delta1.lp = rnorm(1), phi0.lpsi = rnorm(1),
   phi1.lpsi = rnorm(1), phi0.lp = rnorm(1), phi1.lp = rnorm(1))

# Parameters monitored
params <- c("delta0.lpsi", "delta1.lpsi", "delta0.lp", "delta1.lp", "phi0.lpsi",
"phi1.lpsi", "phi0.lp", "phi1.lp", "psi", "p", "Nocc.fs", "Nsite")

# MCMC settings
ni <- 12000 ; nt <- 2 ; nb <- 2000 ; nc <- 3

# Call JAGS from R (ART 12 min), look at convergence and summarize posteriors
out8 <- jags(win.data, inits, params, "model8.txt", n.chains = nc, n.thin = nt,
n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(3,3))
traceplot(out8, c('delta0.lpsi', 'delta1.lpsi', 'delta0.lp', 'delta1.lp', 'phi0.lpsi',
'phi1.lpsi', 'phi0.lp', 'phi1.lp'))
print(out8, dig = 3)
```

	mean	sd	2.5%	50%	97.5%	overlap0	f	Rhat	n.eff
delta0.lpsi	-0.553	0.490	-1.530	-0.552	0.414	TRUE	0.871	1.001	3854
delta1.lpsi	-0.705	0.239	-1.177	-0.702	-0.238	FALSE	0.998	1.001	2538
delta0.lp	2.038	0.300	1.449	2.037	2.633	FALSE	1.000	1.000	15000
delta1.lp	-0.758	0.175	-1.105	-0.756	-0.424	FALSE	1.000	1.000	15000
phi0.lpsi	1.716	0.376	0.995	1.711	2.467	FALSE	1.000	1.001	1751
phi1.lpsi	-0.159	0.192	-0.533	-0.158	0.221	TRUE	0.799	1.001	1743
phi0.lp	-0.477	0.414	-1.260	-0.491	0.369	TRUE	0.873	1.002	1157
phi1.lp	0.480	0.219	0.044	0.484	0.897	FALSE	0.985	1.002	1936

This example highlights the considerable power that BUGS gives you for testing biological hypotheses even inside a fairly complex HM! We finish up this section by plotting the body mass relationships of parameters for the community mean and community heterogeneity (=variance) of the logits of occupancy and detection probabilities in Swiss birds (Figure 11.12). On average, larger species are less widespread and less detectable than smaller species. There is less unexplained variability among species in this relationship for occupancy in larger species, while for detection, larger species are more variable around this relationship than are smaller species. We note, though, that these relationships are less clear for the two variances, and the 95% CRI of `phi1.lpsi` covers 0.

```
# Get covariate values for prediction
predm <- seq(10, 10000,,500)              # Predict for mass of 10g to 10 kg
pred.logm <- log10(predm)

# Compute predictions (all in one array)
tmp <- out8$sims.list                     # Grab simulation list
nsamp <- out8$mcmc.info$n.samples         # Number of MCMC samples
pred <- array(NA, dim = c(500, nsamp, 4))  # Array for predictions
```

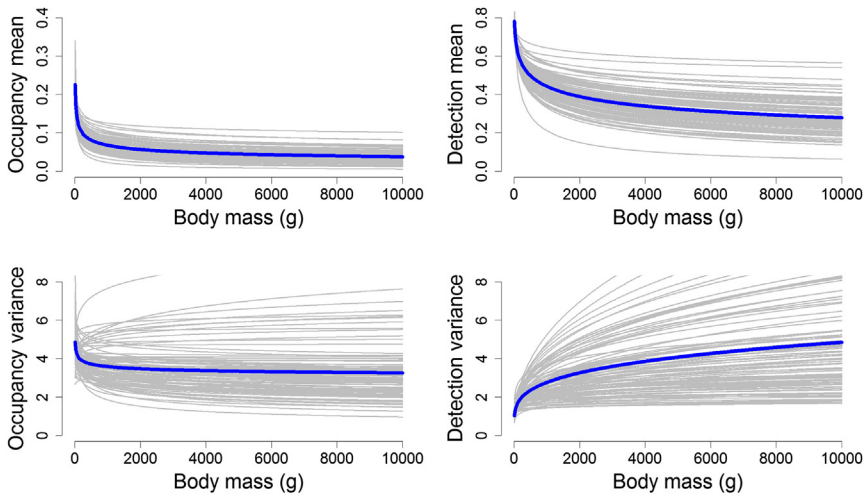

FIGURE 11.12

Estimates of the community relationships between occupancy and detection probability and body size in Swiss birds (top: community means, bottom: community variance, or interspecific variability). Gray lines show a random sample of size 100 of the posterior distribution of these relationships and the blue lines show the posterior mean of these relationships.

```
# [,,1] mu.psi, [,,2] mu.p, [,,3] var.lpsi, [,,4] var.lp
for(i in 1:nsamp){                    # Fill array
  pred[,i,1] <- plogis(tmp$delta0.lpsi[i] + tmp$delta1.lpsi[i] * pred.logm)
  pred[,i,2] <- plogis(tmp$delta0.lp[i] + tmp$delta1.lp[i] * pred.logm)
  pred[,i,3] <- exp(tmp$phi0.lpsi[i] + tmp$phi1.lpsi[i] * pred.logm)
  pred[,i,4] <- exp(tmp$phi0.lp[i] + tmp$phi1.lp[i] * pred.logm)
}

# Plot posterior mean and a random sample of 100 from posterior of regression
selection <- sample(1:nsamp, 100)       # Choose random sample of MCMC output
par(mfrow = c(2,2), mar = c(5,5,2,2))
matplot(predm, pred[,selection,1], ylab = "Occupancy mean", xlab = "Body mass (g)",
type = "l", lty = 1, lwd = 1, col = "grey", ylim = c(0, 0.4), frame = F)
lines(predm, apply(pred[,,1], 1, mean), lwd = 3, col = "blue")
matplot(predm, pred[,selection,2], ylab = "Detection mean", xlab = "Body mass (g)",
type = "l", lty = 1, lwd = 1, col = "grey", ylim = c(0, 0.8), frame = F)
lines(predm, apply(pred[,,2], 1, mean), lwd = 3, col = "blue")
matplot(predm, pred[,selection,3], ylab = "Occupancy variance", xlab = "Body mass (g)",
type = "l", lty = 1, lwd = 1, col = "grey", ylim = c(0, 8), frame = F)
lines(predm, apply(pred[,,3], 1, mean), lwd = 3, col = "blue")
matplot(predm, pred[,selection,4], ylab = "Detection variance", xlab = "Body mass (g)",
type = "l", lty = 1, lwd = 1, col = "grey", ylim = c(0, 8), frame = F)
lines(predm, apply(pred[,,4], 1, mean), lwd = 3, col = "blue")
```

We make two final comments, one on "phylogenetic independence," and the other on the inference about unseen species. First, our analysis treats every species as independent, but in an evolutionary sense, they are not: two closely related species may be similar purely because of common ancestry, and you may want to adjust for such phylogenetic nonindependence (Dorazio and Connor, 2014). If you want to weigh species by their phylogenetic "similarity," such that two closely related species contribute less information to the among-species model than do two species that are more distantly related, you could specify a multivariate normal distribution for the species effects, and model the effects of a relationship matrix (which expresses how closely every pair of species are related) into the variance-covariance matrix (Ives and Zhu, 2006; Waldmann, 2009; Papaïx et al., 2010). Second, the extension of this model with species-specific covariates to include unseen species in the metacommunity (see Section 11.7) is straightforward. Except that your model then must include a submodel to estimate the values of such missing "individual covariates" for the unseen species (Royle, 2009; Chapter 6 in Kéry and Schaub, 2012; and also see Chapters 7−8 in this book).

11.6.4 MODELING SPECIES RICHNESS IN A TWO-STEP ANALYSIS

In a community model composed of component models for the presence and absence of individual species, species richness is not a structural parameter that can be modeled (though it is in the multinomial mixture model; see our comments in Sections 11.4 and 11.5.3, as well as Exercise 2, and note that we could model species richness directly also by adoping a logit-linear model for the data-augmentation parameter (Sutherland et al., in review) or using the construction explained in Section 7.8.4.). Instead, species richness is a quantity computed from the matrix of the individual species presence indicators (the presence/absence matrix $\boldsymbol{Z}$). Often, questions in community ecology revolve around relationships between species richness (N) and environmental conditions. How can these be addressed in a community occupancy framework?

Here is a simple two-step analysis, which has been called "doing statistics on statistics" (Link, 1999), but which may be a useful first exploratory step in hierarchical modeling (Murtaugh, 2007): we take estimates from one analysis (here, our community size estimates `Nsite`) and plug them into a second analysis to relate them to the environmental variables of our choice; see Tingley and Beissinger (2013) for an example in the context of a community model. The only problem is that these estimates from the first-step analysis are not independent (they typically have covariances), *and* they come with estimation uncertainty (standard errors). To do a two-step analysis properly, we must propagate the estimation uncertainty from the first analysis into the second analysis; otherwise, we will typically underestimate the uncertainty in the final analysis—i.e., get CIs that are too narrow and with too many significant test results. To do this right is not so trivial, and may often jeopardize the very reason for doing a two-step analysis of an HM, which is simplicity. For instance, it would be *wrong* to conduct a second-step analysis and simply weigh each estimate by the reciprocal of its squared standard error, since this would assume that the residuals in the second analysis are composed exclusively of estimation uncertainty from the first analysis (Jenni and Kéry, 2003).

Here we show how to properly do such a two-step analysis with propagation of the first-step estimation uncertainty into the second-step analysis (we ignore the covariances, but see below that they appear to be negligible). We illustrate by exploring the relationship between richness and elevation using species richness (`Nsite`) estimates from model 5, where we would assume that the lack of random species effects would minimize any covariances between the estimates of the Ns.

This analysis can also be called a meta-analysis, because it synthesizes multiple estimates into a single estimate. In a more typical meta-analysis, the former come from different studies, while here they are simply different estimates from a single model. The code in this section is partly taken from McCarthy and Masters (2005).

```
# Extract estimates of N from model 5
N.pm <- out5$summary[291:557, 1]          # Posterior means of Nsite
N.psd <- out5$summary[291:557, 2]         # ... posterior sd's of Nsite
N.cri <- out5$summary[291:557, c(3,7)]    # ... CRL's of Nsite

# Plot estimates as a function of elevation
elev <- data$elev[1:267]
plot(elev, N.pm, xlab = "Altitude (m a.s.l.)", ylab = "Estimated avian species richness",
ylim = c(0, 70), frame = F)
segments(elev, N.cri[,1], elev, N.cri[,2], col = "grey")
lines(smooth.spline(N.pm ~ elev, w = 1 / N.psd), col = "grey", lwd = 3)
```

We fit a simple regression model with cubic polynomials of elevation, but one that has two residual components. The first is the estimation uncertainty coming from the first-step analysis, the magnitude of which is assumed to be known: this is the posterior standard deviation of the community size estimate `Nsite`. The second component is the usual lack of fit component, which allows the individual data point to lie off the modeled relationship. This component will be estimated from the data. In our model, we will compute predictions of species richness for a range of values for elevation.

```
# Bundle and summarize data set
pred.ele <- (seq(200, 2750,5) - mean.ele) / sd.ele      # elevation standardised
str(win.data <- list(ele = ele, N = N.pm, psd = N.psd, n = length(N.pm),
pred.ele = pred.ele, npred = length(pred.ele)))

# Define model in BUGS language
sink("meta.analysis.txt")
cat("
model{

# Priors
for(v in 1:4){     # Priors for intercept and polynomial coefficients
   beta[v] ~ dnorm(0, 0.0001)
}
tau.site <- pow(sd.site, -2)
sd.site ~ dunif(0,10)

# Likelihood
for (i in 1:n){
   N[i] ~ dnorm(muN[i], tau.psd[i])        # Measurement error model for estimated N
   tau.psd[i] <- pow(psd[i], -2)           # 'Known' part of residual: meas. error
    muN[i] <- beta[1] + beta[2] * ele[i] + beta[3] * pow(ele[i],2) +
    beta[4] * pow(ele[i],3) + eps.site[i]  # add another source of uncertainty
   eps.site[i] ~ dnorm(0, tau.site)        # this is the usual 'residual'
}
```

```
# Get predictions for plot
for(i in 1:npred){
    Npred[i] <- beta[1] + beta[2] * pred.ele[i] + beta[3] * pow(pred.ele[i],2) +
beta[4] * pow(pred.ele[i],3)
}
}  # end model
",fill=TRUE)
sink()

# Initial values, params monitored, and MCMC settings
inits <- function() list(beta = rnorm(4))
params <- c("beta", "sd.site", "Npred")
ni <- 12000 ; nt <- 10 ; nb <- 2000 ; nc <- 3

# Call JAGS and summarize posterior
out <- jags(win.data, inits, params, "meta.analysis.txt", n.chains = nc, n.thin
= nt, n.iter = ni, n.burnin = nb)
print(out, 3)
```

We add to our plot the predictions of species richness from our meta-analysis, with 95% CRI of the polynomial relationship.

```
lines(seq(200, 2750,5), out$mean$Npred, col = "blue", lwd = 3)
matlines(seq(200,2750,5), out$summary[6:516,c(3, 7)], col = "blue", lwd = 2,
lty= "dashed")
```

The analysis represented by the blue lines in Figure 11.13 (though not the gray spline smooth) properly propagates the uncertainty in the estimates of *N* from the first analysis into the estimates of the second analysis, but ignores covariances. We can see how bad this may be by plotting the joint posteriors of *N* for pairs of sites. They seem to be mostly uncorrelated (most clouds are fairly round), and we conclude that the covariances of the *N* estimates are small at best, and hence results from our two-step analysis are likely to be adequate.

```
par(mfrow = c(3, 3), mar = c(5,4,3,2))
for(i in 1:267){
   for(j in 1:267){
      plot(jitter(out5$sims.list$Nsite[,i]), jitter(out5$sims.list$Nsite[,j]),
      main = paste("Joint posterior sites", i, "and", j))
#    browser()
   }
}
```

Such a Bayesian meta-analysis is a very useful way of synthesizing in a single estimate or in several parameters (here, the coefficients of a polynomial regression) a collection of estimates while properly accommodating, and therefore weighing, the uncertainty in these estimates. In addition to formal model parameters, you can easily do this for quantities that are *not* formal parameters in your model and for which you therefore cannot directly specify a submodel, as for *N* here. Finally, this method can be very useful to model parameters in a two-step hierarchical procedure as a substitute for fitting a formal HM all at once. Sometimes, such a two-step analysis (which is a hierarchical procedure, but not

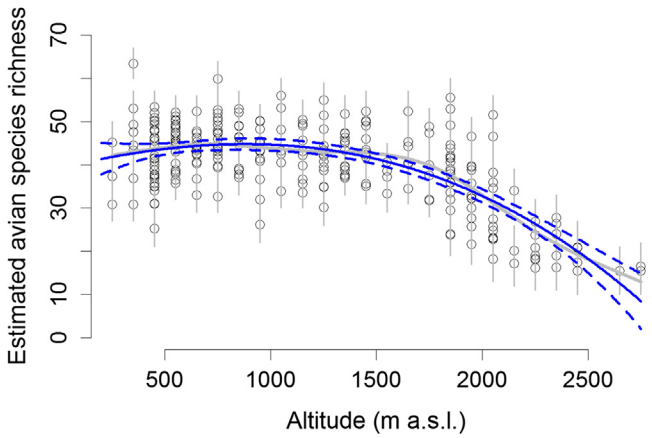

FIGURE 11.13

Relationship between elevation and estimated avian species richness in Switzerland at 1-km^2 sample sites, conditional on the 145 species detected at least once during the 2014 surveys. Symbols denote point estimates with 95% CRIs from community model 5. Gray line is a spline smooth with weights equal to the reciprocal of the squared posterior standard deviations. Blue line is the cubic regression line estimated in the meta-analysis that accounts for both estimation error (posterior standard deviations) and residual variation around the regression line. The dashed blue lines give the 95% CRI of the prediction.

a hierarchical model) may be easier to understand or to explain to somebody else (Murtaugh, 2007), or it may be necessary because your full HM is too big to be fit by your computer all at once (see Sauer and Link, 2002, 2011, for a good example of the latter).

11.7 THE DORAZIO/ROYLE (DR) COMMUNITY OCCUPANCY MODEL WITH DATA AUGMENTATION (DA)

In all previous models, our inferences were restricted to the set of those 145 species that were detected at least once during the MHB surveys in 2014. We did *not* explicitly extend the scope of inference in these models to any species that may have been exposed to sampling, but happened to be missed. Species may be missed because either they did occur in the general area (i.e., were part of the regional species pool), but happened to be absent from the particular 267 sites surveyed; or the species did occur in the particular set of surveyed sites, but simply failed to be detected during the repeated "measurements" of their presence/absence. In models 7 and 8, we already extended the scope of inference from the 145 observed species to some undefined larger statistical "population" or community of species, by treating species-specific parameters as random effects. Thereby, we were able to formally estimate certain quantities of that community: these were the hyperparameters governing species-specific random effects. In the final and most sophisticated set of models in this chapter, we will formally extend the scope of inference to *all* species that constitute the avian metacommunity sampled by the Swiss MHB. This model will allow us to also make inferences about occurring species that were missed by the surveys. That is, to species that were not detected in any survey of any sample quadrat.

By failing to account for such missed species, any estimator of a community quantity such as species richness or the average response to some covariate, under all previous models, was conditional on the list of 145 detected species. If there are any undetected species, the previous estimators also underestimate the true species richness at each site. Furthermore, with our previous models, we were unable to estimate the total size of the metacommunity. Finally, our previous models may give biased results with respect to the total metacommunity, because the 145 species that were detected at least once may be a biased sample from the total metacommunity. For instance, quite likely, the sample of detected species will be biased positively with respect to the average detection probability in the metacommunity. As a consequence, the sample of species detected may also be biased in other traits directly or indirectly related to detection probability of a species; probably, for instance, occupancy (see Section 11.6.2).

Our method of making an inference about species that were never detected is again based on *parameter-expanded data augmentation*, or *data augmentation* for short (PX-DA or DA) (Tanner and Wong, 1987; Liu and Wu, 1999; Royle et al., 2007a; Royle and Dorazio, 2012; see also Sections 7.8.1.4 and 8.3). Dorazio and Royle (2005) developed a multispecies occupancy community model that extends inferences to unseen species, and Dorazio et al. (2006) fit this model using DA with BUGS. In this section, we will fit two such Dorazio/Royle (DR) community occupancy models. We will see that they are conceptually fairly simple extensions of the models in Section 11.6; hence, everything that we said and did there remains relevant for the full DR community models with DA.

11.7.1 THE SIMPLEST DR COMMUNITY MODEL WITH DA

To formally extend our inferences to the unseen species in a metacommunity, we imagine a superpopulation of, say, M species (where M is much larger than the total number of species in the metacommunity N), and then we estimate which members of M are also members of N. That is, which of a number of M potentially occurring species are exposed to our spatial sample comprising our S sites. (In this section, we again deviate slightly from our standard notation, and use S instead of M for the number of study sites, as we did in Chapters 7–9.) We achieve this by PX-DA (Royle et al., 2007a). In a sense, "DA" comes before "PX": it means that we first add all-zero detection histories for a number of $nz = M - n$ additional potential species, where n is the number of observed species (here 145) and nz is the number of all-zero species added. The PX part means that we add another hierarchical layer (represented by another parameter) to our model, one that describes the random sampling of the N occurring "real" species from the total of M potential species. This sampling process is represented by a Bernoulli random variable w, which is an indicator for a species that is part of the studied metacommunity (we also call it the data augmentation variable). Consequently, our hierarchical community model now has three main levels and can be written in following conditional (i.e., linked) probability statements:

1. Superpopulation process : $w_k \sim Bernoulli(\Omega)$
2. State process (occurrence) : $z_{ik}|w_k \sim Bernoulli(w_k \psi_k)$
3. Observation process (detection) : $ysum_{ik}|z_{ik} \sim Binomial(J_i, z_{ik} p_k)$
4. Models of species heterogeneity : $\text{logit}(\psi_k) \sim Normal\left(\mu_{lpsi}, \sigma^2_{lpsi}\right)$

   $\text{logit}(p_k) \sim Normal\left(\mu_{lp}, \sigma^2_{lp}\right)$

The species index k now runs from 1 to M—i.e., up to the number of species in the full augmented data set rather than only up to the 145 observed species. But otherwise, the model of species heterogeneity is *exactly* the same as for model 6 (except for the trivial difference that here we choose not to specify a correlation parameter). In a sense, the "community occupancy parameter" Ω now takes the place of the metacommunity size N, since the expected value of metacommunity size N is $M\Omega$. You can think of this as exactly analogous to an occupancy problem, where the number of occupied sites (corresponding to N) can be estimated as the product of the occupancy parameter (corresponding to Ω) and the total number of surveyed sites (corresponding to M). Note, hence, that Ω has no meaning if you don't know M.

Table 11.2 shows a conceptual outline of the community occupancy model with PX-DA for the Swiss MHB 2014 data set. Purely for layout reasons, we transpose the matrices *ysum* and $\boldsymbol{Z}$, but this doesn't change anything in the model. The observed data (the dark-gray shaded rectangle) is represented by $ysum_{ki}$, i.e., the detection frequencies of the 145 species observed in the 267 quadrats. To these we will add 150 species-detection histories containing only zeros (this is the data augmentation part represented by the medium-gray shaded rectangle below), bringing our total augmented data set to

Table 11.2 Conceptual outline of the DR community occupancy model for the Swiss MHB 2014 data set.

		Observed: *ysum*					Only partially observed: $\boldsymbol{Z}$ and w					
	Quadrat i	1	2	3	...	267	1	2	3	...	267	w_k
Species k	1	3	2	0	...	2	**1**	**1**	**1**	...	**1**	**1**
	2	0	0	1	...	2	**0**	**1**	**1**	...	**1**	**1**
	3	2	0	1	...	0	**1**	**0**	**1**	...	**1**	**1**
	...	...	...	...	...	...	...	...	...	...	...	...
n	145	0	0	1	...	0	**0**	**1**	**1**	...	**1**	**1**
n+1	146	0	0	0	...	0	**0**	**1**	**0**	...	**0**	**1**
	...	...	...	...	...	...	...	...	...	...	...	...
	...	0	0	0	...	0	**0**	**0**	**1**	...	**0**	**1**
N	**?**	0	0	0	...	0	**0**	**0**	**0**	**0**	**0**	**0**
N+1	**?**+1	0	0	0	...	0	**0**	**0**	**0**	...	**0**	**0**
	...	...	...	...	...	...	...	...	...	...	...	...
	...	...	...	...	...	...	...	...	...	...	...	...
M=n+nz	295	0	0	0	...	0	**0**	**0**	**0**	...	**0**	**0**

$M = 295$ "potential" species. The model permits inferences about two latent structures, the true (augmented) presence/absence matrix $\boldsymbol{Z}$ (i.e., the M-by-267 matrix containing the occurrence indicators z_{ki}) and the "metacommunity membership indicators" or DA variables w_k (both represented by yellow shading to indicate the studied metacommunity). Essentially, one main aim of the modeling is to estimate the unobserved values in the arrays $\boldsymbol{Z}$ and $\boldsymbol{w}$, which are shown in red. Note that we show the $\boldsymbol{Z}$ matrix for the studied metacommunity with fewer than N species to emphasize that the "conditional" metacommunity size (the number of species that occur in the actual sample of 267 survey sites) may well be less than the number of species that form the entire metacommunity (N) inhabiting the wider area sampled by these sites. The "unconditional" metacommunity size is the asymptote of the former—i.e., the number of species actually occurring anywhere in your sampled sites as the number of sample sites goes to infinity (Dorazio et al., 2006; Dupuis et al., 2011).

DA in models for abundance or related quantities can always be imagined as turning a sort of logistic regression model into an occupancy model where each "individual" appears on a row in the data set. We do this by adding to the data a large number of potential and undetected individuals with all-zero detection histories and by adding to the logistic regression model one hierarchical layer representing the presence/absence indicator z (which we here denote w). We have used DA extensively in the multinomial abundance models in Chapters 7–9. What perhaps makes DA more challenging to grasp in the context of the DR community model is that even the basic DR models without DA (i.e., the models in Section 11.6) are *already* site-occupancy models (we have one occupancy model for each observed species). So when we do DA to extend our inferences to the entire metacommunity, we add a second level of such an occupancy, or zero-inflated, type of model. If you find this concept challenging to grasp, then it is best to first try to fully understand DA in the simpler context of estimation of abundance in a closed population (e.g., see Chapter 6 in Kéry and Schaub, 2012; Royle et al., 2007a; Royle and Dorazio, 2012, and Chapters 7–9 in this book). Let's now look at PX-DA in the community models in practice.

```
# Augment data set (DA part)
nz <- 150                  # Number of potential species in superpopulation
M <- nspec + nz            # Size of augmented data set ('superpopulation')
yaug <- cbind(ysum, array(0, dim=c(nsite, nz)))     # Add all zero histories

# Bundle and summarize data set
str( win.data <- list(yaug = yaug, nsite = nrow(ysum), nrep =
data$nsurvey[1:nsite], M = M, nspec = nspec, nz = nz) )

# Specify model in BUGS language
sink("model9.txt")
cat("
model {

# Priors to describe heterogeneity among species in community
for(k in 1:M){                  # Loop over all species in augmented list
  lpsi[k] ~ dnorm(mu.lpsi, tau.lpsi)
  lp[k] ~ dnorm(mu.lp, tau.lp)
}

# Hyperpriors to describe full community
omega ~ dunif(0,1)                  # Data augmentation or 'occupancy' parameter
```

```
mu.lpsi ~ dnorm(0,0.001)              # Community mean of occupancy (logit)
mu.lp ~ dnorm(0,0.001)                # Community mean of detection (logit)
tau.lpsi <- pow(sd.lpsi, -2)
sd.lpsi ~ dunif(0,5)                  # Species heterogeneity in logit(psi)
tau.lp <- pow(sd.lp, -2)
sd.lp ~ dunif(0,5)                    # Species heterogeneity in logit(p)

# Superpopulation process:this is the 'paramater expansion' part of PX-DA
for(k in 1:M){
  w[k] ~ dbern(omega)                 # Metacommunity membership indicator
}                                     # (or data augmentation variable)

# Ecological model for latent occurrence z (process model)
for(k in 1:M){
  mu.psi[k] <- w[k] * psi[k]          # species not part of community zeroed out for z
  logit(psi[k]) <- lpsi[k]
  for (i in 1:nsite) {
    z[i,k] ~ dbern(mu.psi[k])
  }
}

# Observation model for observed detection frequencies
for(k in 1:M){
  logit(p[k]) <- lp[k]
for (i in 1:nsite) {
   mu.p[i,k] <- z[i,k] * p[k]    # non-occurring species are zeroed out for p
   yaug[i,k] ~ dbin(mu.p[i,k], nrep[i])
   }
}

# Derived quantities
for(k in 1:M){
   Nocc.fs[k] <- sum(z[,k])           # Number of occupied sites among the 267
}
for (i in 1:nsite) {
   Nsite[i] <- sum(z[i,])             # Number of occurring species at each site
}
n0 <- sum(w[(nspec+1):(nspec+nz)])  # Number of unseen species in metacommunity
Ntotal <- sum(w[])                    # Total metacommunity size (= nspec + n0)
}
",fill = TRUE)
sink()

# Initial values
wst <- rep(1, nspec+nz)                          # Simply set everybody at 'occurring'
zst <- array(1, dim = c(nsite, nspec+nz))        # ditto for z
inits <- function() list(z = zst, w = wst, lpsi = rnorm(n = nspec+nz), lp =
rnorm(n = nspec+nz))
```

```
# Parameters monitored
params <- c("mu.lpsi", "sd.lpsi", "mu.lp", "sd.lp", "psi", "p", "Nsite",
"Ntotal", "omega", "n0")

# MCMC settings
ni <- 22000 ; nt <- 2 ; nb <- 2000 ; nc <- 3

# Call JAGS from R (ART 62 min), check convergence and summarize posteriors
out9 <- jags(win.data, inits, params, "model9.txt", n.chains = nc, n.thin = nt,
n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(2,2)) ; traceplot(out9, c('mu.lpsi', 'sd.lpsi', 'mu.lp', 'sd.lp'))
print(out9, dig = 3)
```

```
                 mean      sd     2.5%      50%    97.5% overlap0 f   Rhat n.eff
mu.lpsi        -2.511   0.343   -3.285   -2.480   -1.937    FALSE 1  1.004   808
sd.lpsi         2.596   0.263    2.153    2.573    3.178    FALSE 1  1.005   530
mu.lp           0.639   0.127    0.382    0.642    0.881    FALSE 1  1.000 14319
sd.lp           1.332   0.114    1.129    1.325    1.576    FALSE 1  1.001  2335
psi[1]          0.022   0.010    0.007    0.020    0.044    FALSE 1  1.000 28192
psi[2]          0.016   0.010    0.004    0.014    0.041    FALSE 1  1.001 15194
psi[3]          0.018   0.009    0.005    0.016    0.039    FALSE 1  1.000 30000
 [ ... output truncated ...]
Nsite[265]     17.654   2.222   14.000   18.000   22.000    FALSE 1  1.000 10161
Nsite[266]     49.997   1.333   48.000   50.000   53.000    FALSE 1  1.000  8094
Nsite[267]     52.619   1.209   51.000   52.000   55.000    FALSE 1  1.000  4639
Ntotal        164.818   9.102  152.000  163.000  187.000    FALSE 1  1.006   570
omega           0.558   0.042    0.484    0.556    0.647    FALSE 1  1.003   978
n0             19.818   9.102    7.000   18.000   42.000    FALSE 1  1.006   570
```

Now, estimates of local species richness or community size, `Nsite`, *do* include a contribution from those `n0` species that were never seen during the surveys in 2014. Consequently, we can now also estimate the total number of species in the larger area from which the 267 study quadrats were drawn as a random sample (this is quantity N or `Ntotal` in the model). We can look at the posterior distributions of the site-specific number of occurring species, and compare these estimates with the observed number of species (Figure 11.14). We see fairly different species-coverage rates in different quadrats (compare, for instance, quadrat 9 and 250, with a larger proportion of species estimated to have been missed in the latter). Quadrat 30 was not surveyed in 2014, and yet we can make a formal guess of the likely number of species occurring in it. In this trivial model, this may not be a very interesting estimate, but below we will see how we can refine such estimates for unsurveyed sites in a model with covariates on detection and occupancy.

```
# Plot posterior distribution of site-specific species richness (Nsite)
par(mfrow = c(3,3), mar = c(5,4,3,2))
for(i in 1:267){
  plot(table(out9$sims.list$Nsite[,i]), main = paste("Quadrat", i),
  xlab = "Local species richness", ylab = "", frame = F,
  xlim = c((min(C[i], out9$sims.list$Nsite[,i], na.rm = T)-2),
  max(out9$sims.list$Nsite[,i]) ))
  abline(v = C[i], col = "grey", lwd = 4)
  browser()
}
```

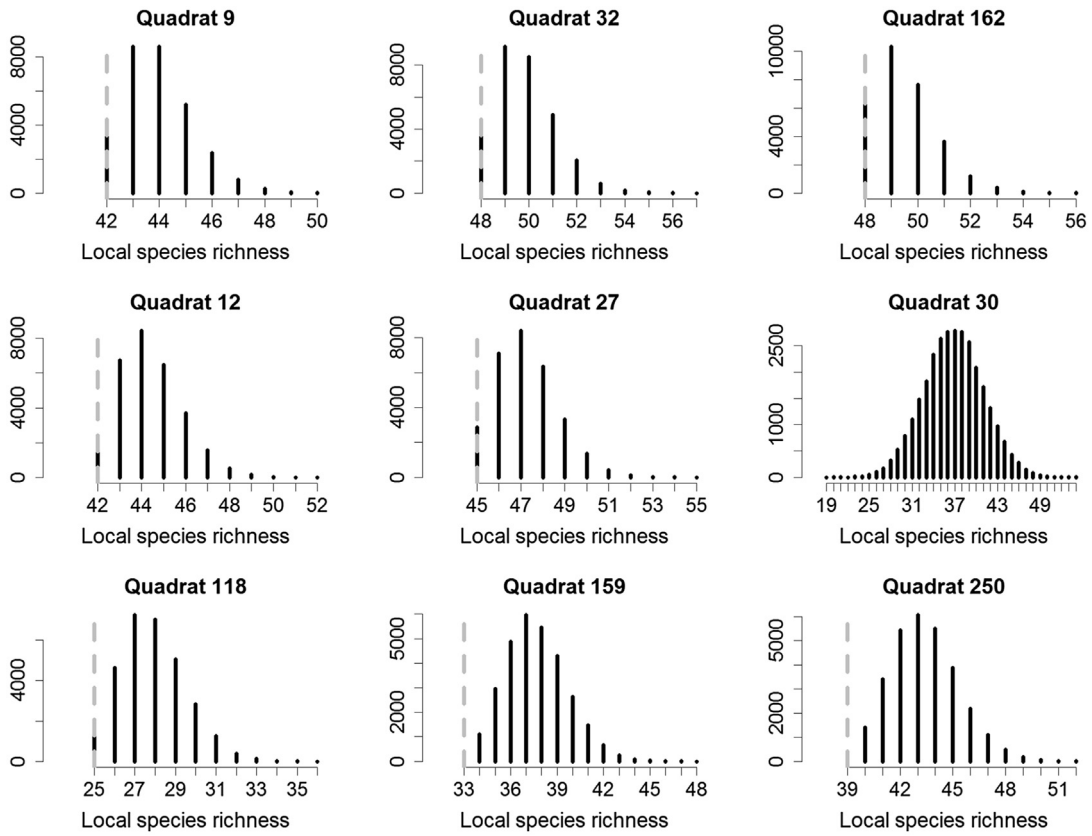

FIGURE 11.14

Posterior distributions of local species richness, or community size (`Nsite`), at a selection of nine 1-km^2 sample quadrats in the Swiss breeding bird survey MHB. Gray dashed lines show the observed number of species in that year. Site 30 was not surveyed in 2014.

```
# Plot it only for a selection of sites
par(mfrow = c(3,3), mar = c(5,4,3,2))
for(i in c(9, 32, 162, 12, 27, 30, 118, 159, 250)){
    plot(table(out9$sims.list$Nsite[,i]), main = paste("Quadrat", i),
    xlab = "Local species richness", ylab = "", frame = F,
    xlim = c((min(C[i], out9$sims.list$Nsite[,i], na.rm = T)-2),
    max(out9$sims.list$Nsite[,i]) ))
    abline(v = C[i], col = "grey", lwd = 4)
}
```

To finish up, we plot the posterior distribution of the metacommunity size (`Ntotal`)—i.e., the estimated number of species that occur in some wider area that is sampled by our 267 survey quadrats (Figure 11.15). This "regional pool" of species is a somewhat hypothetical value, and strictly speaking,

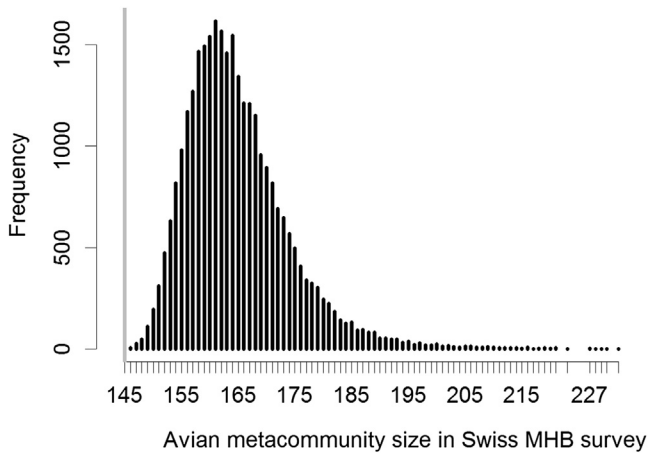

FIGURE 11.15

Posterior distribution of the metacommunity size in the Swiss MHB or, more precisely, the expected number of bird species occurring in some larger area that is sampled by the 267 quadrats in the hypothetical case that the number of sample quadrats goes to infinity. Gray line indicates the 145 species observed in the year 2014.

we don't know exactly to which area it refers. Given the nice spatial coverage of Switzerland by the MHB sample (see Figures 1.3 and 10.12), we can loosely think of `Ntotal` as an estimate of the total number of breeding bird species in Switzerland. Sometimes estimates of `Ntotal` agree quite well with the known number of species in some larger region such as Switzerland (e.g., in the analysis in Kéry and Royle, 2009), but there is no reason that they ought to do so always. Since our model is not spatially explicit, we do not know the effective sample area, and hence cannot say to which area our metapopulation size estimate refers.

```
# Plot posterior distribution of total species richness (Ntotal)
plot(table(out9$sims.list$Ntotal), main = "", ylab = "", xlab = "Avian
metacommunity size in Swiss MHB survey (267 1km2 quadrats)", frame = F, xlim =
c(144, 245))
abline(v = nspec, col = "grey", lwd = 4)
```

We estimate that there were 165 species (95% CRI 152−187), and therefore that 20 species (95% CRI 7−42) did either not occur in the 267 sample quadrats or they did occur in the surveyed quadrats, but they were missed by the observers surveying the quadrats. If we make the *ad hoc* interpretation that `Ntotal` is the number of Swiss breeding birds, then this is an underestimate. As of 2015, there were 215 known breeding bird species in Switzerland (179 regular, 20 irregular, 16 occasional; V. Keller, pers. comm.); hence, in any one year there might be 190−200 species breeding in Switzerland.

Our current model does not use all information about the occurrence of a species at a site where it was not detected. There are at least two further sources of such information: one comes from covariate relationships and the other from co-occurring species (Clark et al., 2014). (A third source of such information might be represented by "spatial relationships", i.e., if a species is known at a nearby site, it may also be more likely to occur at a given site, even if undetected; see Chapters 21−22 for possible ways in which such information could be introduced in the model.) Exploiting these relationships with

the habitat and the occurrence of other bird species would allow us to further improve our estimates of how likely it is for a given species to occur at a site where it was not detected. Therefore, we might also improve our community and metacommunity size estimates. In the next section, we will add covariate information into the model. Hierarchical modeling of species co-occurrence patterns as well as the modeling of spatial autocorrelation in HMs will be dealt with in volume 2.

11.7.2 DR COMMUNITY MODEL WITH COVARIATES

When we incorporate covariate information into our DR community model, we can study the aggregate response of all species in a community to such covariates by looking at the mean and variance hyperparameters governing the species-specific occupancy and detection parameters. In addition, we can also investigate species-specific relationships. This is a great advantage over simpler approaches that directly model species richness as a response to the environment. To model time-specific covariates into the detection part of the DR community model, our current response (the detection frequency $ysum_{ik}$) must be disaggregated. That is, we must go back to the original detection/nondetection data y_{ijk} and treat them as a Bernoulli random variable. We can write the most complex occupancy model in this chapter as:

1. Superpopulation process : $w_k \sim Bernoulli(\Omega)$
2. State process (occurrence) : $z_{ik}|w_k \sim Bernoulli(w_k\psi_k)$
3. Observation process (detection) : $y_{ijk}|z_{ik} \sim Bernoulli(z_{ik}p_{ijk})$
4. Models of species heterogeneity : $\text{logit}(\psi_{ik}) = lpsi_k + betalpsi_k * elevation_i + \ldots$

$$\text{logit}(p_{ijk}) = lp_k + betalp_k * date_{ij} + \ldots$$

with

$$lpsi_k \sim Normal\left(\mu_{lpsi}, \sigma^2_{lpsi}\right)$$

$$betalpsi_k \sim Normal\left(\mu_{betalpsi}, \sigma^2_{betalpsi}\right)$$

$$lp_k \sim Normal\left(\mu_{lp}, \sigma^2_{lp}\right)$$

$$betalp_k \sim Normal\left(\mu_{betalp}, \sigma^2_{betalp}\right)$$

So, occupancy and detection probability are regressed on a number of covariates, and the intercepts and slopes of both regressions are species-specific random effects, with a common prior distribution as well as mean and variance hyperparameters that are estimated. As in the previous section the index for species (k) runs from 1 to $M = n + nz$—i.e., up to the number of species in the full augmented data set. For covariates on occupancy, we will fit elevation (linear and squared) and forest cover; for detection covariates, we will fit survey date (linear and squared) and survey duration.

We will see that with covariates, metacommunity size will often be estimated at a (much) higher value than it is under the more simplistic previous DR model that assumed both the occurrence and detection probabilities of a species were constant over all sites and all surveys. This makes intuitive sense: according to the second law of capture-recapture, unmodeled heterogeneity in detection will bias low population size estimators, and here arguably, occupancy probability and therefore species richness. Our new analysis will account for much more such heterogeneity among species, sites, and surveys, and therefore should be better able to accommodate such heterogeneity and thereby avoid negative bias.

How should we choose the amount of DA—i.e., the number of "potential" species nz by which we augment our data set? There are basically two approaches. The first is the one that we chose for

model 9: we made `nz` so large that the posterior of `Ntotal` was not affected by our choice. Our use of DA induces a discrete uniform prior on `Ntotal`—i.e., *Ntotal ~ DU(0, M)*. Hence, if we want a vague prior for `Ntotal`, we choose `nz` by trial and error: we start by adding a couple of *nz* all-zero detection histories, fit the model, and inspect the posterior distribution of `Ntotal`. If its mass is piling up against the chosen value of $M = n + nz$, we repeat the analysis with a larger value of *nz*, and do this until the posterior distribution of `Ntotal` is no longer right-truncated. In model 9, the posterior of `Ntotal` was not truncated by our choice of $M = n + nz$ (Figure 11.15), so our prior for metacommunity size was vague in that model. A second approach (Dorazio et al., 2010, 2011) is to fix *M* at a biologically plausible value, thus effectively choosing an informative prior on `Ntotal` by fixing it at a known value of species richness in the wider studied area. Currently, there are about 215 known breeding bird species in Switzerland, and hence this approach would let us augment by $nz = 215 - 145 = 70$ all-zero species. We fitted the following model using both approaches. First, after some trial and error (and a l-o-o-o-n-g waiting time each), we found that $nz = 250$, and hence $M = 145 + 250 = 395$, represented a fairly vague prior for `Ntotal`. Second, we fit the model with $M = 145 + 70 = 215$ species. Below, we briefly compare the results from the two approaches, but first we show how to fit the model. (Also see Link, 2013, on use of scale priors in models with data augmentation, including DR models.)

```
# Augment data set: choose one of two different priors on Ntotal
nz <- 250                 # Use for vague prior on Ntotal: M = 395
nz <- 215 - nspec         # Use for informative prior on Ntotal: M = 215
yaug <- array(0, dim=c(nsite, nrep, nspec+nz)) # array with only zeroes
yaug[,,1:nspec] <- y      # copy into it the observed data

# Create same NA pattern in augmented species as in the observed species
missings <- is.na(yaug[,,1]) # e.g., third survey in high-elevation quads
for(k in (nspec+1):(nspec+nz)){
   yaug[,,k][missings] <- NA
}
```

In our experience, one of the main practical challenges in this kind of model (and also similar models where we deal with arrays rather than just vectors) is to come to grips with modeling the data in a multidimensional array (which you can imagine as some kind of an orderly "box"). Also, sometimes some lines of code must be moved around inside or outside of some of the two to three loops within which we define the elements of these arrays, to avoid defining a quantity repeatedly (this is the "multiple definition of…" trap in BUGS). This requires a *very* clear understanding of the "box" into which we place these quantities: you must know exactly which dimension stands for what index in an array. Note that we can loop over the dimensions of the array in the order we like (and some orders may result in better mixing—see tip 12 in Appendix 1 of Kéry and Schaub, 2012). However, each index of the array has a clearly defined meaning. In our case, one is for sites, another is for replicate surveys, and the last is for species. Which is which is defined by the way that we build this array in the first place.

For all but tiny communities, these are very parameter-rich models, and saving all species-specific parameters, including those for the *nz* potential species, will produce huge results files that will cost us a lot of memory. Moreover, the computation time to produce just the posterior summary when running BUGS from R will be huge. Therefore, to minimize both at the end of the BUGS program, we may save into new structures the parameter estimates for the observed species plus one potential species and save only these (though we do this only when we run the model with $nz = 250$).

We now package the data set and run the analysis.

```
# Bundle and summarize data
str(win.data <- list(y = yaug, nsite = dim(y)[1], nrep = dim(y)[2], nspec =
dim(y)[3], nz = nz, M = nspec + nz, ele = ele, forest = forest, DAT = DAT, DUR = DUR) )
List of 10
 $ y      : num [1:267, 1:3, 1:395] 0 0 0 0 0 0 0 0 0 0 ...
 $ nsite  : int 267
 $ nrep   : int 3
 $ nspec  : int 145
 $ nz     : num 250          # This is for nz = 250 and M = 395
 $ M      : num 395
 $ ele    : num [1:267] -1.1539 -1.1539 -0.2175 -0.3735 -0.0614 ...
 $ forest : num [1:267] -1.1471 -0.4967 -0.0992 -0.9303 0.0092 ...
 $ DAT    : num [1:267, 1:3] -1.415 -1.19 -1.235 -0.559 -1.64 ...
 $ DUR    : num [1:267, 1:3] -0.43511 -0.78764 -0.52324 1.23944 0.00556 ...

# Specify model in BUGS language
sink("model10.txt")
cat("
model {

# Priors
omega ~ dunif(0,1)
# Priors for species-specific effects in occupancy and detection
for(k in 1:M){
  lpsi[k] ~ dnorm(mu.lpsi, tau.lpsi)      # Hyperparams describe community
  betalpsi1[k] ~ dnorm(mu.betalpsi1, tau.betalpsi1)
  betalpsi2[k] ~ dnorm(mu.betalpsi2, tau.betalpsi2)
  betalpsi3[k] ~ dnorm(mu.betalpsi3, tau.betalpsi3)
  lp[k] ~ dnorm(mu.lp, tau.lp)
  betalp1[k] ~ dnorm(mu.betalp1, tau.betalp1)
  betalp2[k] ~ dnorm(mu.betalp2, tau.betalp2)
  betalp3[k] ~ dnorm(mu.betalp3, tau.betalp3)
}

# Hyperpriors
# For the model of occupancy
mu.lpsi ~ dnorm(0,0.01)
tau.lpsi <- pow(sd.lpsi, -2)
sd.lpsi ~ dunif(0,8)    # as always, bounds of uniform chosen by trial and error
mu.betalpsi1 ~ dnorm(0,0.1)
tau.betalpsi1 <- pow(sd.betalpsi1, -2)
sd.betalpsi1 ~ dunif(0, 4)
mu.betalpsi2 ~ dnorm(0,0.1)
tau.betalpsi2 <- pow(sd.betalpsi2, -2)
sd.betalpsi2 ~ dunif(0,2)
mu.betalpsi3 ~ dnorm(0,0.1)
tau.betalpsi3 <- pow(sd.betalpsi3, -2)
sd.betalpsi3 ~ dunif(0,2)
```

```
# For the model of detection
mu.lp ~ dnorm(0,0.1)
tau.lp <- pow(sd.lp, -2)
sd.lp ~ dunif(0, 2)
mu.betalp1 ~ dnorm(0,0.1)
tau.betalp1 <- pow(sd.betalp1, -2)
sd.betalp1 ~ dunif(0,1)
mu.betalp2 ~ dnorm(0,0.1)
tau.betalp2 <- pow(sd.betalp2, -2)
sd.betalp2 ~ dunif(0,1)
mu.betalp3 ~ dnorm(0,0.1)
tau.betalp3 <- pow(sd.betalp3, -2)
sd.betalp3 ~ dunif(0,1)

# Superpopulation process: Ntotal species sampled out of M available
for(k in 1:M){
   w[k] ~ dbern(omega)
}

# Ecological model for true occurrence (process model)
for(k in 1:M){
  for (i in 1:nsite) {
    logit(psi[i,k]) <- lpsi[k] + betalpsi1[k] * ele[i] + betalpsi2[k] * pow(ele[i],2) +
      betalpsi3[k] * forest[i]
    mu.psi[i,k] <- w[k] * psi[i,k]
    z[i,k] ~ dbern(mu.psi[i,k])
  }
}

# Observation model for replicated detection/nondetection observations
for(k in 1:M){
  for (i in 1:nsite){
    for(j in 1:nrep){
      logit(p[i,j,k]) <- lp[k] + betalp1[k] * DAT[i,j] + betalp2[k] * pow(DAT[i,j],2) +
        betalp3[k] * DUR[i,j]
      mu.p[i,j,k] <- z[i,k] * p[i,j,k]
      y[i,j,k] ~ dbern(mu.p[i,j,k])
    }
  }
}

# Derived quantities
#for(k in 1:M){
#    Nocc.fs[k] <- sum(z[,k])          # Number of occupied sites among the 267
#}
for (i in 1:nsite){
    Nsite[i] <- sum(z[i,])            # Number of occurring species at each site
}
n0 <- sum(w[(nspec+1):(nspec+nz)])    # Number of unseen species
Ntotal <- sum(w[])                    # Total metacommunity size
```

```
# Vectors to save (S for 'save'; discard posterior samples for
# all minus 1 of the potential species to save disk space)
# we do this for nz = 250 (i.e., M = 395)
lpsiS[1:(nspec+1)] <- lpsi[1:(nspec+1)]
betalpsi1S[1:(nspec+1)] <- betalpsi1[1:(nspec+1)]
betalpsi2S[1:(nspec+1)] <- betalpsi2[1:(nspec+1)]
betalpsi3S[1:(nspec+1)] <- betalpsi3[1:(nspec+1)]
lpS[1:(nspec+1)] <- lp[1:(nspec+1)]
betalp1S[1:(nspec+1)] <- betalp1[1:(nspec+1)]
betalp2S[1:(nspec+1)] <- betalp2[1:(nspec+1)]
betalp3S[1:(nspec+1)] <- betalp3[1:(nspec+1)]
}
",fill = TRUE)
sink()

# Initial values
wst <- rep(1, nspec+nz)                    # Simply set everybody at occurring
zst <- array(1, dim = c(nsite, nspec+nz)) # ditto
inits <- function() list(z = zst, w = wst, lpsi = rnorm(n = nspec+nz), betalpsi1 = rnorm(n =
nspec+nz), betalpsi2 = rnorm(n = nspec+nz), betalpsi3 = rnorm(n = nspec+nz), lp = rnorm(n =
nspec+nz), betalp1 = rnorm(n = nspec+nz), betalp2 = rnorm(n = nspec+nz), betalp3 = rnorm(n =
nspec+nz))
```

The number of estimated quantities ("parameters" or "estimands") in this analysis is truly enormous and may make your computer "explode," or you may have to wait for many days for models to be fit—literally! Hence, we may run the analyses in multiple steps, with a different set of estimands saved each time. We also choose different MCMC settings for each, and different modes of running JAGS. The first set is where we monitor convergence and save MCMC samples for the main structural parameters of the model, which are the hyperparameters describing the metacommunity. In addition, we want to look at the estimates of metacommunity and community size, or `Ntotal` and `Nsite`, respectively, yielding a total of $1 + 16 + 1 + 267 = 285$ "parameters." That may seem like a lot. But wait, you ain't seen nothing yet…

```
# Set 1
params1 <- c("omega", "mu.lpsi", "sd.lpsi", "mu.betalpsi1", "sd.betalpsi1",
"mu.betalpsi2", "sd.betalpsi2", "mu.betalpsi3", "sd.betalpsi3", "mu.lp", "sd.lp",
"mu.betalp1", "sd.betalp1", "mu.betalp2", "sd.betalp2", "mu.betalp3", "sd.betalp3",
"Ntotal", "Nsite")

# MCMC settings
ni <- 15000 ; nt <- 10 ; nb <- 5000 ; nc <- 3

# Run JAGS, check convergence and summarize posteriors
out101 <- jags(win.data, inits, params1, "model10.txt", n.chains = nc, n.thin =
nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
par(mfrow = c(2, 2))
traceplot(out101, c(c("omega", "mu.lpsi", "sd.lpsi", "mu.betalpsi1",
"sd.betalpsi1", "mu.betalpsi2", "sd.betalpsi2", "mu.betalpsi3", "sd.betalpsi3",
"mu.lp", "sd.lp", "mu.betalp1", "sd.betalp1", "mu.betalp2",
"sd.betalp2", "mu.betalp3", "sd.betalp3", "Ntotal")) )
```

In the second set, we want to get posterior samples from species-specific effects in occupancy and detection, data augmentation variable *w*, and the presence/absence matrix ***Z***. For this, we simply need the MCMC samples and are not interested in any summaries (which take *many* days to compute in R on a laptop); hence, we use a more basic way of running JAGS from R, with function `jags.basic` in the `jagsUI` package, that will only return the MCMC samples. For DA of up to $M = 215$ species, we will estimate $215 * 8 + 215 * 267 + 215 = 59{,}340$ "parameters." Now, *that* is a big model!

```
# Set 2
params2 <- c("mu.lpsi", "sd.lpsi", "mu.betalpsi1", "sd.betalpsi1", "mu.betalpsi2",
"sd.betalpsi2", "mu.betalpsi3", "sd.betalpsi3", "lpsi", "betalpsi1", "betalpsi2",
"betalpsi3", "lp", "betalp1", "betalp2", "betalp3", "z", "w")
ni <- 12000 ; nt <- 20 ; nb <- 2000 ; nc <- 3
out102 <- jags.basic(win.data, inits, params2, "model10.txt", n.chains = nc,
n.thin = nt, n.iter = ni, n.burnin = nb, parallel = TRUE)
library(coda)
all10 <- as.matrix(out102)          # Put output from 3 chains into a matrix
summary(out102)                     # May take a loooong time
gelman.diag(out102)                 # ditto
```

Augmenting a data set up to size M induces a discrete uniform prior on the range (0, M) for the metacommunity size *Ntotal*. We fitted this model with $Ntotal \sim DU(0, 215)$ and with $Ntotal \sim DU(0, 395)$. The former is a clearly informative prior, while the latter is only very weakly informative about metacommunity size. We needed much longer Markov chains to achieve convergence, and computation time was *very much* longer for the latter. Interestingly, however, in terms of inferences, the choice hardly mattered at all! The only exceptions were the estimates of metacommunity size `Ntotal`, and the mean and standard deviation hyperparameters for the species-specific intercepts of the occupancy model, which are μ_{lpsi} and σ^2_{lpsi}, respectively. With $M = 395$, we estimated a much greater metacommunity composed of species with lower mean occupancy probability, but more interspecific variability in the occupancy intercept. Estimates of all other hyperparameters were hardly changed, although posterior standard deviations in the occupancy model were just slightly larger for $M = 395$.

```
# Comparison of main hyperparameters when M = 215 and with M = 395
# (not all code to produce this output is shown)
print(cbind(out10.215$summary[1:17,c(1:3, 7)], out10.395$summary[1:17, c(1:3, 7)]), 2)
                mean     sd     2.5%    97.5%    mean      sd     2.5%    97.5%
mu.lpsi       -4.404  0.466   -5.309   -3.450   -6.32   1.197   -8.638   -4.135
sd.lpsi        4.785  0.357    4.126    5.520    5.66   0.634    4.451    6.960
mu.betalpsi1  -0.663  0.205   -1.065   -0.268   -0.65   0.198   -1.046   -0.270
sd.betalpsi1   2.443  0.178    2.115    2.809    2.41   0.171    2.091    2.760
mu.betalpsi2  -0.974  0.094   -1.159   -0.790   -1.00   0.098   -1.189   -0.812
sd.betalpsi2   0.804  0.085    0.649    0.979    0.80   0.082    0.653    0.976
mu.betalpsi3  -0.102  0.091   -0.279    0.075   -0.11   0.091   -0.286    0.066
sd.betalpsi3   0.997  0.078    0.854    1.155    0.99   0.074    0.853    1.141
mu.lp          0.729  0.137    0.451    0.991    0.71   0.140    0.428    0.981
sd.lp          1.417  0.127    1.192    1.696    1.45   0.129    1.220    1.727
mu.betalp1     0.068  0.052   -0.034    0.173    0.07   0.051   -0.026    0.170
sd.betalp1     0.461  0.048    0.374    0.563    0.46   0.048    0.377    0.564
```

```
mu.betalp2    -0.137  0.039   -0.216   -0.064   -0.14   0.039    -0.219    -0.064
sd.betalp2     0.333  0.039    0.260    0.414    0.33   0.040     0.261     0.416
mu.betalp3     0.210  0.031    0.148    0.272    0.21   0.031     0.149     0.272
sd.betalp3     0.227  0.033    0.167    0.296    0.23   0.033     0.168     0.294
Ntotal       205.763  8.238  184.975  215.000  273.50  42.635   200.000   361.000
```

In addition, community size estimates (`Nsite`) were essentially identical under both metacommunity size priors. Posterior means were on average greater by 0.16 species (corresponding to 0.4%) with $M = 395$ than with $M = 215$, and posterior standard deviations of `Nsite` were on average 1.2% greater. This suggests that the very rare species have only a negligible effect on the spatial patterns of local species richness. In summary then, in our example all of the important parameters were remarkably insensitive to the choice of prior on metacommunity size. Hence, we base our inference on the model with informative prior on the metacommunity size ($M = 215$), since this takes *much* less time to fit, plus we feel that it is motivated by existing knowledge of the community of birds in Switzerland.

```
out10 <- out101
```

Our model enables inferences at all three hierarchical levels—metacommunity, local community, and individual species—and we now give examples of each.

First the metacommunity, which is described by the following quantities in the model: `omega`, `mu.lpsi`, `sd.lpsi`, `mu.betalpsi1`, `sd.betalpsi1`, `mu.betalpsi2`, `sd.betalpsi2`, `mu.betalpsi3`, `sd.betalpsi3`, `mu.lp`, `sd.lp`, `sd.betalp1`, `mu.betalp2`, `sd.betalp2`, `mu.betalp3`, `sd.betalp3`, `n0`, `Ntotal`. There is perhaps not much point in estimating metacommunity size, since we used an informative prior on `Ntotal`. However, we can use the hyperparameters to describe the metacommunity—e.g., in terms of the species mean and among-species variability of occupancy and detection parameters, or the community average response to covariates. We can do this by simply sampling the respective normal distributions and then inverse-logit transforming the resulting values. Figure 11.16 suggests that there is plenty of among-species variability in both occupancy and detection probability, but that a minority of species are widespread, whereas many have an extremely limited

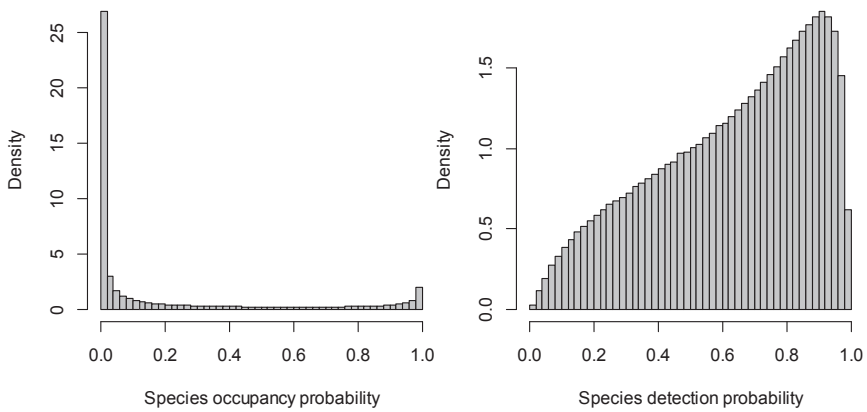

FIGURE 11.16

Community distribution of average occupancy and detection probability among Swiss breeding bird species based on estimates of the species intercept model parameters (`mu.lpsi`, `sd.lpsi`, `mu.lp`, `sd.lp`) for the spatiotemporal scale of 1 km^2 and the three-month-long breeding season in 2014.

distribution, and that a majority are relatively easy to detect. The average species occupancy probability for the spatiotemporal scale in the MHB survey was 0.19, and the average species detection probability at the mean survey conditions (date and duration) was 0.63.

```
par(mfrow = c(1,2))              # Fig. 11-16
psi.sample <- plogis(rnorm(10^6, mean = out10$mean$mu.lpsi, sd = out10$mean$sd.lpsi))
p.sample <- plogis(rnorm(10^6, mean = out10$mean$mu.lp, sd = out10$mean$sd.lp))
hist(psi.sample, freq = F, breaks = 50, col = "grey", xlab = "Species occupancy
probability", ylab = "Density", main = "")
hist(p.sample, freq = F, breaks = 50, col = "grey", xlab = "Species detection probability",
ylab = "Density", main = "")
summary(psi.sample) ; summary(p.sample)
```

We can also look at the interspecific variability in every parameter, which appears in our model as standard deviations of normal priors for species-specific effects. We simply plot histograms of the posteriors for each. Executing the next body of code, you will see that the difference among species in the Swiss breeding bird community varies a great deal among the different parameters of the occupancy and detection models (note different scales of abscissa).

```
par(mfrow = c(2,4))          # Among-species variability in parameters (not shown)
hist(out10$sims.list$sd.lpsi, breaks = 100, col = "grey", xlim = c(0,6), main =
"Occupancy: intercept")
abline(v = mean(out10$sims.list$sd.lpsi), col = "blue", lwd = 3)
hist(out10$sims.list$sd.betalpsi1, breaks = 100, col = "grey", xlim = c(0,3),
main = "Occupancy: linear effect of elevation")
abline(v = mean(out10$sims.list$sd.betalpsi1), col = "blue", lwd = 3)
hist(out10$sims.list$sd.betalpsi2, breaks = 100, col = "grey", xlim = c(0,3),
main = "Occupancy: quadratic effect of elevation")
abline(v = mean(out10$sims.list$sd.betalpsi2), col = "blue", lwd = 3)
hist(out10$sims.list$sd.betalpsi3, breaks = 100, col = "grey", xlim = c(0,3),
main = "Occupancy: linear effect of forest cover")
abline(v = mean(out10$sims.list$sd.betalpsi3), col = "blue", lwd = 3)
hist(out10$sims.list$sd.lp, breaks = 100, col = "grey", xlim = c(0,2),
main = "Detection: intercept")
abline(v = mean(out10$sims.list$sd.lp), col = "blue", lwd = 3)
hist(out10$sims.list$sd.betalp1, breaks = 100, col = "grey", xlim = c(0,1), main
= "Detection: linear effect of survey date")
abline(v = mean(out10$sims.list$sd.betalp1), col = "blue", lwd = 3)
hist(out10$sims.list$sd.betalp2, breaks = 100, col = "grey", xlim = c(0,1), main
= "Detection: quadratic linear effect of survey date")
abline(v = mean(out10$sims.list$sd.betalp2), col = "blue", lwd = 3)
hist(out10$sims.list$sd.betalp3, breaks = 100, col = "grey", xlim = c(0,1), main
= "Detection: linear effect of survey duration")
abline(v = mean(out10$sims.list$sd.betalp3), col = "blue", lwd = 3)
```

Next, we look at the community average response to modeled covariates, where we predict the mean community response of occupancy probability to elevation and forest cover, and of species detection probability to survey date and survey duration. As always, we create "original" covariates

that cover the entire range of a covariate over which we want to make predictions of a modeled quantity. Then we apply the identical scaling that we used for the actual covariates used in the analysis, and next we apply the linear model parameters estimated in the analysis to get the predictions (we do this here for each of the 3000 posterior samples). Finally, we plot posterior summaries of these predictions against the values of the "original" (i.e., unscaled) covariates. This time we put all four sets of predictions into a single array.

```
# Visualize covariate mean relationships for the average species
o.ele <- seq(200, 2500,,500)                # Get covariate values for prediction
o.for <- seq(0, 100,,500)
o.dat <- seq(15, 120,,500)
o.dur <- seq(100, 420,,500)
ele.pred <- (o.ele - mean.ele) / sd.ele
for.pred <- (o.for - mean.forest) / sd.forest
dat.pred <- (o.dat - mean.date) / sd.date
dur.pred <- (o.dur - mean.dur) / sd.dur

# Predict occupancy for elevation and forest and detection for date and duration
# Put all four predictions into a single
str( tmp <- out10$sims.list )               # grab MCMC samples
nsamp <- length(tmp[[1]])                   # number of mcmc samples
predC <- array(NA, dim = c(500, nsamp, 4))  # "C" for 'community mean'
for(i in 1:nsamp){
    predC[,i,1] <- plogis(tmp$mu.lpsi[i] + tmp$mu.betalpsi1[i] * ele.pred +
    tmp$mu.betalpsi2[i] * ele.pred^2 )
    predC[,i,2] <- plogis(tmp$mu.lpsi[i] + tmp$mu.betalpsi3[i] * for.pred)
    predC[,i,3] <- plogis(tmp$mu.lp[i] + tmp$mu.betalp1[i] * dat.pred +
    tmp$mu.betalp2[i] * dat.pred^2 )
    predC[,i,4] <- plogis(tmp$mu.lp[i] + tmp$mu.betalp3[i] * dur.pred)
}

# Get posterior means and 95% CRIs and plot (Fig. 11-17)
pmC <- apply(predC, c(1,3), mean)
criC <- apply(predC, c(1,3), function(x) quantile(x, prob = c(0.025, 0.975)))

par(mfrow = c(2, 2))
plot(o.ele, pmC[,1], col = "blue", lwd = 3, type = 'l', lty = 1, frame = F,
ylim = c(0, 0.05), xlab = "Elevation (m a.s.l)", ylab = "Community mean occupancy")
matlines(o.ele, t(criC[,,1]), col = "grey", lty = 1)
plot(o.for, pmC[,2], col = "blue", lwd = 3, type = 'l', lty = 1, frame = F,
ylim = c(0, 0.05), xlab = "Forest cover", ylab = "Community mean occupancy")
matlines(o.for, t(criC[,,2]), col = "grey", lty = 1)
plot(o.dat, pmC[,3], col = "blue", lwd = 3, type = 'l', lty = 1, frame = F, ylim =
c(0.2, 0.8), xlab = "Survey date", ylab = "Community mean detection")
matlines(o.dat, t(criC[,,3]), col = "grey", lty = 1)
plot(o.dur, pmC[,4], col = "blue", lwd = 3, type = 'l', lty = 1, frame = F, ylim =
c(0.2, 0.8), xlab = "Survey duration", ylab = "Community mean detection")
matlines(o.dur, t(criC[,,4]), col = "grey", lty = 1)
```

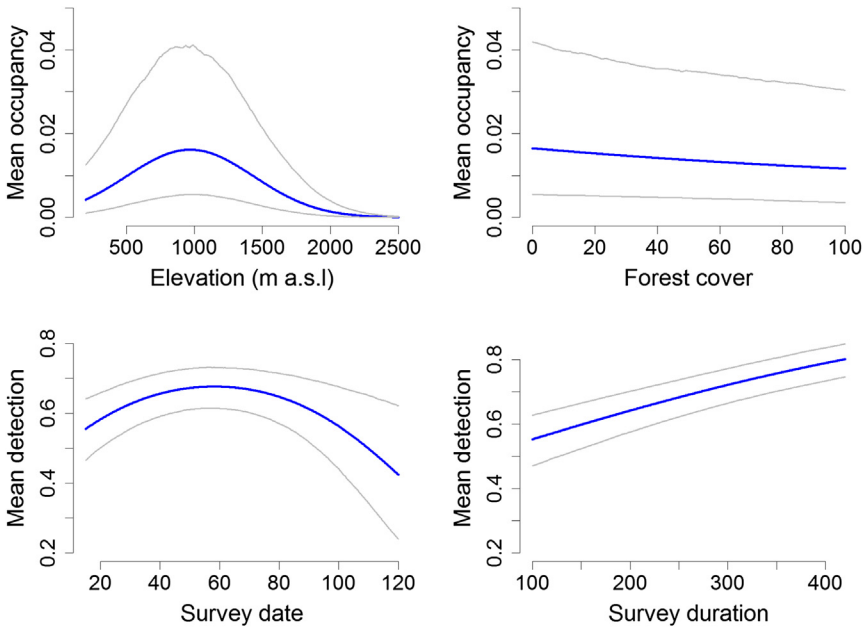

FIGURE 11.17

Community response at the 1-km^2 spatial scale of Swiss breeding bird occupancy probability to elevation and forest cover, and of species detection probability to survey date and survey duration.

The community average occupancy probability of Swiss breeding birds was greatest around 1000 m in elevation and essentially did not respond to forest cover (Figure 11.17, top). On average, Swiss breeding bird species were most detectable around day 60 (May 30), and increasing survey duration in 1-km^2 quadrats from 100 to 400 min pushed detection probability from 0.55 to 0.80 (Figure 11.17, bottom). You may wonder about the strange values of occupancy, whose community mean varies from just about nothing to slightly less than 1.5%. How can we ever observe more than just a handful of species with such low values? The reason we can do this is because we model species variability as random noise around a mean *on the logit scale*, and the community average of the logit occupancy probability (−4.351, corresponding to 0.012 on the probability scale) does not correspond to the average of the probability-scale occupancy intercept. We saw above that the expected value of the community distribution of occupancy (Figure 11.16, left) was about 19%.

Second, moving to inferences at the community (i.e., site-specific) level, we can again obtain the estimates of species richness at each site (`Nsite`), and then do so for the same sample of nine sites that we already plotted in Figure 11.14. The estimates under model 10 in Figure 11.18 are slightly larger, presumably because we accounted for more detection heterogeneity in this model than in model 9. Of course, you may plot estimates of `Nsite` against any site covariate in your model or even outside, to generate hypotheses about drivers of variation in community richness. When plotting the `Nsite` estimates against site elevation (Figure 11.19), we see that incorporating covariates increases `Nsite` estimates at lower elevations, but decreases them at higher elevations, compared with the

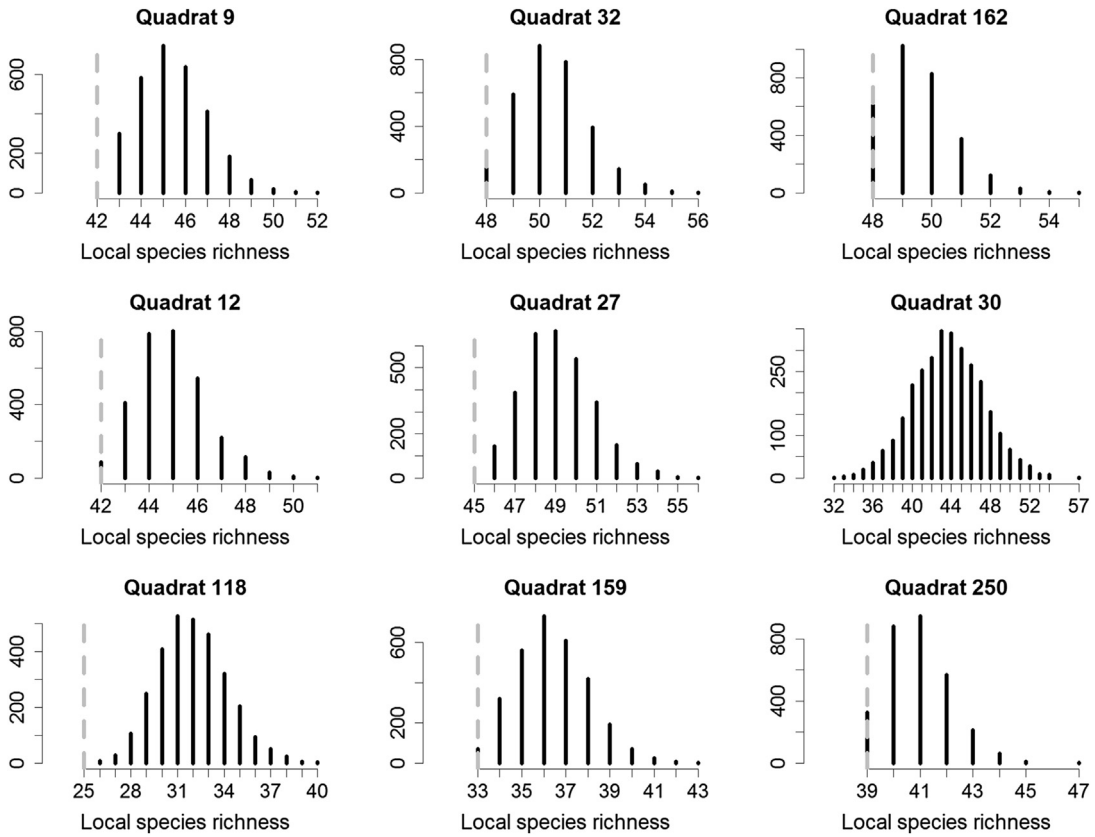

FIGURE 11.18

Sample of posterior distributions of local species richness (community size `Nsite` or alpha diversity) under model 10 at the same selection of nine 1-km^2 MHB sample quadrats as shown in Figure 11.14 (which are estimates of covariate-free model 9).

covariate-free estimates under model 9. This sensible pattern is due to elevation relationships for most species that are estimated in the model (we will see these species-specific estimates below).

```
# Plot posterior distribution of site-specific species richness (Nsite)
par(mfrow = c(3,3), mar = c(5,4,3,2))
for(i in 1:267){
   plot(table(out10$sims.list$Nsite[,i]), main = paste("Quadrat", i), xlab = "Local
   species richness", ylab = "", frame = F, xlim = c((min(C[i], out10$sims.list$Nsite[,i],
   na.rm = T)-2), max(out10$sims.list$Nsite[,i]) ))
   abline(v = C[i], col = "grey", lwd = 4)
   browser()
}
```

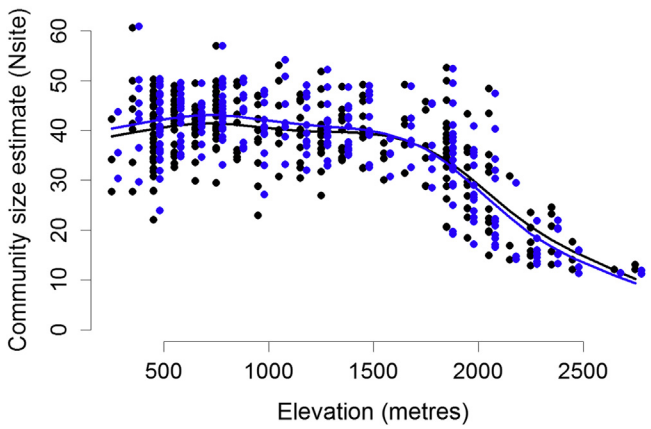

FIGURE 11.19

Comparison between DR models 9 (black, no covariates) and 10 (blue, with covariates) in terms of the relationship between community size (`Nsite`) and elevation. Note slight offset of blue in the x direction.

```
# Plot it only for a selection of sites (Fig. 11-18)
par(mfrow = c(3,3), mar = c(5,4,3,2))
for(i in c(9, 32, 162, 12, 27, 30, 118, 159, 250)){
   plot(table(out10$sims.list$Nsite[,i]), main = paste("Quadrat", i),
   xlab = "Local      species richness", ylab = "", frame = F,
   xlim = c((min(C[i], out10$sims.list$Nsite[,i], na.rm = T)-2),
   max(out10$sims.list$Nsite[,i]) ))
   abline(v = C[i], col = "grey", lwd = 4)
}

# Plot Nsite estimates under models 9 & 10 vs. elevation (Fig. 11-19)
offset <- 30      # Set off elevation for better visibility
plot(elev, out9$mean$Nsite, xlab = "Elevation (metres)", ylab = "Community size
estimate (Nsite)", frame = F, ylim = c(0,60), pch = 16) # black: model 9
lines(smooth.spline(out9$mean$Nsite ~ elev), lwd = 3)
points(elev+offset, out10$mean$Nsite, pch = 16, col = "blue") # blue: model 10
lines(smooth.spline(out10$mean$Nsite ~ elev), lwd = 3, col = "blue")
```

Lastly, we present inferences about individual species under our community occupancy model. To present the species-specific inferences and look at the parameter estimates of covariate effects, we use the second set of MCMC output and first compute posterior means and 95% CRIs of the species-specific parameters from the "naked" MCMC samples contained in `out102`.

```
str(all10)                    # Look at the MCMC output
pm <- apply(all10, 2, mean)   # Get posterior means and 95% CRIs
cri <- apply(all10, 2, function(x) quantile(x, prob = c(0.025, 0.975)))   # CRIs
```

Now we produce plots of the species-specific parameter estimates in the occupancy and detection models for all 145 observed species and compare them with the community average

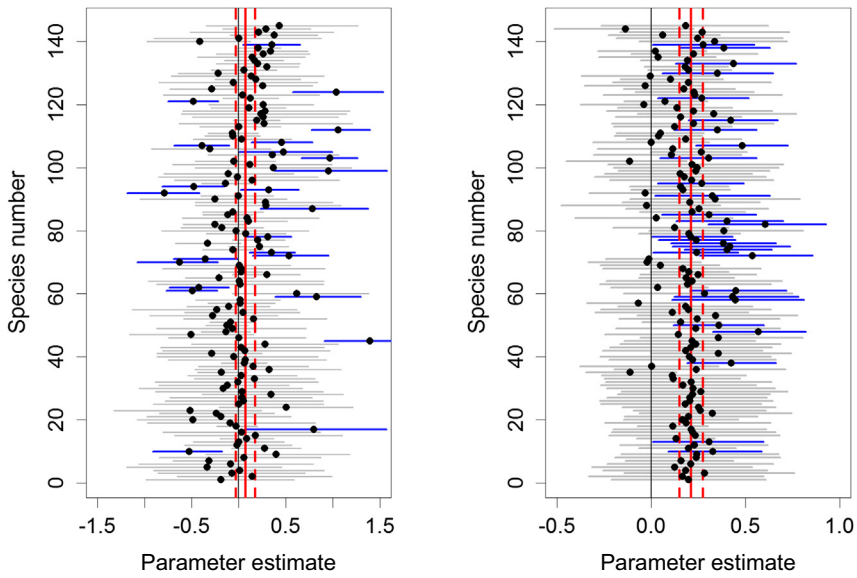

FIGURE 11.20

Comparison between community and individual species response of detection probability to survey date (linear, left) and survey duration (right). Red lines show posterior mean and 95% CRI of the community mean hyperparameter, and black dots and gray lines show the same thing for each individual species (for the 145 observed species), with species CRIs that do not overlap zero colored in blue.

(the hyperparameters). Beyond the interest of inspecting parameter estimates for each species, these plots should emphasize again that the community average may be quite different from that of an individual species. In case you know European birds and would like to inspect the parameter estimates for a particular species, remember that the 145 parameters are in the same order as in the object `ordered.spec.name.list`.

```
# Effects of date (linear and quadratic) and of duration on detection
#par(mfrow = c(1,3), cex.lab = 1.3, cex.axis = 1.3) # Can put all three in one
par(mfrow = c(1,2), cex.lab = 1.3, cex.axis = 1.3)
# Date linear (Fig. 11-20 left)
plot(pm[1:145], 1:145, xlim = c(-1.5, 1.5), xlab = "Parameter estimate", ylab
= "Species number", main = "Effect of date (linear) on detection", pch = 16)
abline(v = 0, lwd = 2, col = "black")
segments(cri[1, 1:145], 1:145, cri[2, 1:145], 1:145, col = "grey", lwd = 1)
sig1 <- (cri[1, 1:145] * cri[2, 1:145]) > 0
segments(cri[1, 1:145][sig1 == 1], (1:145)[sig1 == 1], cri[2, 1:145][sig1 == 1], (1:145)
[sig1 == 1], col = "blue", lwd = 2)
abline(v = out101$summary[11,1], lwd = 3, col = "red")
abline(v = out101$summary[11,c(3,7)], lwd = 2, col = "red", lty = 2)

# Date quadratic (not shown)
plot(pm[216:360], 1:145, xlim = c(-1.5, 1.5), xlab = "Parameter estimate", ylab = "Species
number", main = "Effect of date (quadratic) on detection", pch = 16)
```

```
abline(v = 0, lwd = 2, col = "black")
segments(cri[1, 216:360], 1:145, cri[2, 216:360], 1:145, col = "grey", lwd = 1)
sig2 <-  (cri[1, 216:360] * cri[2, 216:360]) > 0
segments(cri[1, 216:360][sig2 == 1], (1:145)[sig2 == 1], cri[2, 216:360][sig2 == 1],
(1:145)[sig2 == 1], col = "blue", lwd = 2)
abline(v = out101$summary[13,1], lwd = 3, col = "red")
abline(v = out101$summary[13, c(3,7)], lwd = 3, col = "red", lty = 2)
```

In the detection model, a total of 24 species among the 145 observed had a "significant" effect of date (linear, do `sum(sig1)`), in the sense that the 95% CRI of the parameter estimate did not cover zero (these are the blue lines in Figure 11.20). The community mean effect of date (linear), `mu.betalp1`, was estimated at 0.068 (95% CRI: −0.034, 0.173). Only 20 species had a "significant" effect of date (squared) on detection probability, although the community mean effect of date (squared), `mu.betalp2`, was "significantly" negative (posterior mean and 95% CRI: −0.137 (−0.216, −0.064)). Thirty-seven species had either a significant linear or a significant squared effect of date (or both) on detection probability (do `sum((sig1 + sig2) > 0)`). Finally, even though the community mean effect of survey duration on detection, `mu.betalp3`, was clearly positive (posterior mean and 95% CRI: 0.210 (0.148, 0.272), individually it was so for only 29 species (Figure 11.20, right).

```
# Survey duration (Fig. 11-20 right)
plot(pm[431:575], 1:145, xlim = c(-0.5, 1), xlab = "Parameter estimate", ylab =
"Species number", main = "Effect of survey duration on detection", pch = 16)
abline(v = 0, lwd = 2, col = "black")
segments(cri[1, 431:575], 1:145, cri[2, 431:575], 1:145, col = "grey", lwd = 1)
sig3 <-  (cri[1, 431:575] * cri[2, 431:575]) > 0
segments(cri[1, 431:575][sig3 == 1], (1:145)[sig3 == 1], cri[2, 431:575][sig3 == 1],
(1:145)[sig3 == 1], col = "blue", lwd = 2)
abline(v = out101$summary[15,1], lwd = 3, col = "red")
abline(v = out101$summary[15, c(3,7)], lwd = 3, col = "red", lty = 2)
```

In the model for occupancy probability, 104 species had a "significant" linear and 68 a "significant" quadratic effect of elevation (Figures 11.21 and 11.22). The community mean response was negative for both, although the "significant" individual species responses are both negative and positive for the former, but always negative for the latter. This shows that even though species have different elevation preferences, ultimately their occupancy probability declines toward zero with increasing elevation. As we have seen before, there was no effect of forest cover at the community mean level, but for 66 species individually, we did find such an effect (35 negative and 31 positive; Figure 11.23). Of course, this is not surprising at all, since everybody knows (well, at least most birdwatchers do) that some species like forest and others don't like it at all. But it illustrates nicely the limitations for community analyses that do not keep track of individual species responses (e.g., the models in Section 11.5).

```
# Effect of elevation (linear) on occupancy probability (Fig. 11-21)
plot(pm[646:790], 1:145, xlim = c(-8, 8), xlab = "Parameter estimate", ylab =
"Species number", main = "Effect of elevation (linear) on occupancy", pch = 16)
abline(v = 0, lwd = 2, col = "black")
segments(cri[1, 646:790], 1:145, cri[2, 646:790], 1:145, col = "grey", lwd = 1)
```

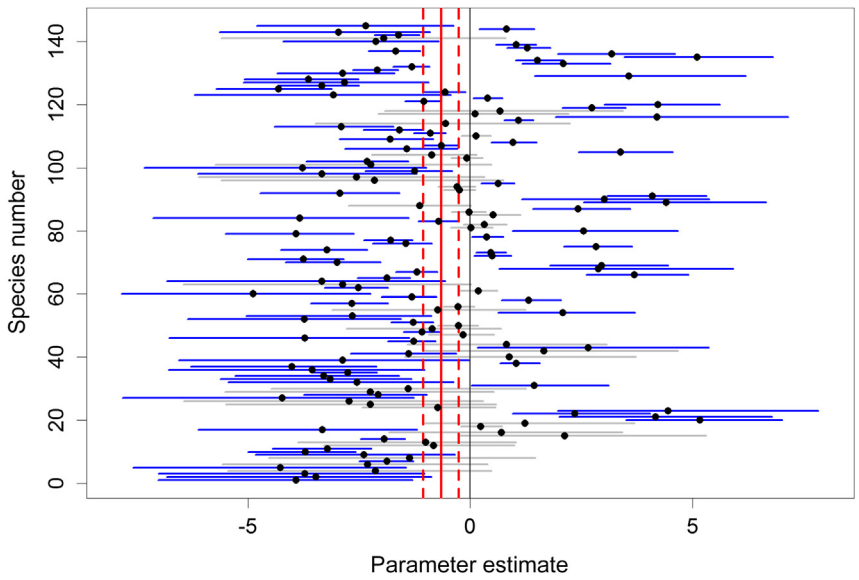

FIGURE 11.21

Comparison between community and individual species response of occupancy probability at the 1-km^2 scale to site elevation (linear) for 145 observed species. Color and symbols as in Figure 11.20.

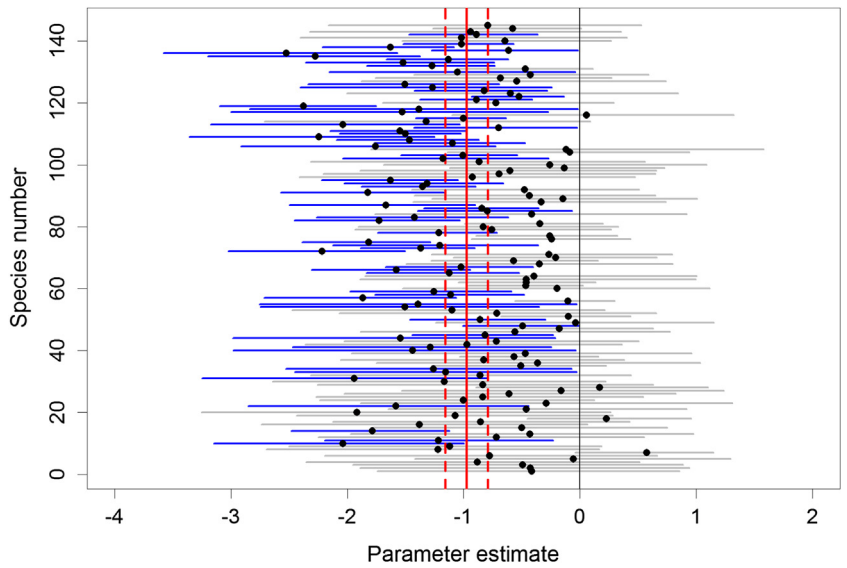

FIGURE 11.22

Comparison between community and individual species response of occupancy probability at the 1-km^2 scale to site elevation (squared) for 145 observed species. Color and symbols as in Figure 11.20.

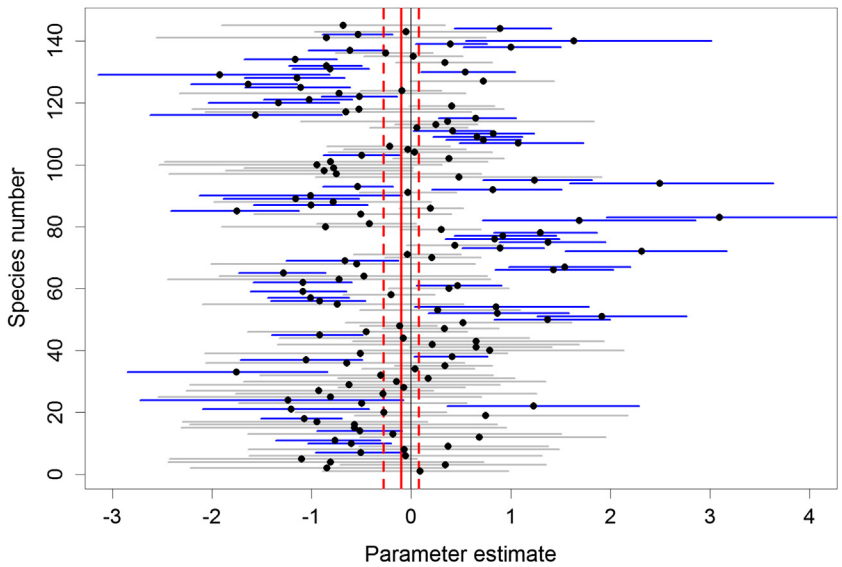

FIGURE 11.23

Comparison between community and individual species response of occupancy probability at the 1-km^2 scale to forest cover for 145 observed species. Color and symbols as in Figure 11.20.

```
sig4 <- (cri[1, 646:790] * cri[2, 646:790]) > 0
segments(cri[1, 646:790][sig4 == 1], (1:145)[sig4 == 1], cri[2, 646:790][sig4 == 1],
(1:145)[sig4 == 1], col = "blue", lwd = 2)
abline(v = out101$summary[3,1], lwd = 3, col = "red")
abline(v = out101$summary[3,c(3,7)], lwd = 3, col = "red", lty = 2)

# Effect of elevation (quadratic) on occupancy probability (Fig. 11-22)
plot(pm[861:1005], 1:145, xlim = c(-4, 2), xlab = "Parameter estimate", ylab =
"Species number", main = "Effect of elevation (quadratic) on occupancy", pch = 16)
abline(v = 0, lwd = 2, col = "black")
segments(cri[1, 861:1005], 1:145, cri[2, 861:1005], 1:145, col = "grey", lwd=1)
sig5 <- (cri[1, 861:1005] * cri[2, 861:1005]) > 0
segments(cri[1, 861:1005][sig5 == 1], (1:145)[sig5 == 1], cri[2, 861:1005][sig5 == 1],
(1:145)[sig5 == 1], col = "blue", lwd = 2)
abline(v = out101$summary[5,1], lwd = 3, col = "red")
abline(v = out101$summary[5,c(3,7)], lwd = 3, col = "red", lty = 2)

# Effect of forest (linear) on occupancy probability (Fig. 11-23)
plot(pm[1076:1220], 1:145, xlim = c(-3, 4), xlab = "Parameter estimate", ylab = "Species
number", main = "Effect of forest cover on occupancy", pch = 16)
abline(v = 0, lwd = 2, col = "black")
segments(cri[1, 1076:1220], 1:145, cri[2, 1076:1220],1:145, col = "grey", lwd=1)
sig6 <- (cri[1, 1076:1220] * cri[2, 1076:1220]) > 0
```

```
segments(cri[1, 1076:1220][sig6 == 1], (1:145)[sig6 == 1], cri[2, 1076:1220][sig6 == 1],
(1:145)[sig6 == 1], col = "blue", lwd = 2)
abline(v = out101$summary[7,1], lwd = 3, col = "red")
abline(v = out101$summary[7,c(3,7)], lwd = 3, col = "red", lty = 2)
negsig6 <- (cri[1, 1076:1220] < 0 & cri[2, 1076:1220] < 0) == 1 # sig negative
possig6 <- (cri[1, 1076:1220] > 0 & cri[2, 1076:1220] > 0) == 1 # sig positive
```

Finally, we plot species-specific predictions of detection and occupancy as a function of the covariates for the 145 observed species, using code for prediction covariates used for the community mean predictions above and doing the analogous thing as we did there. Figures 11.24 and 11.25 show

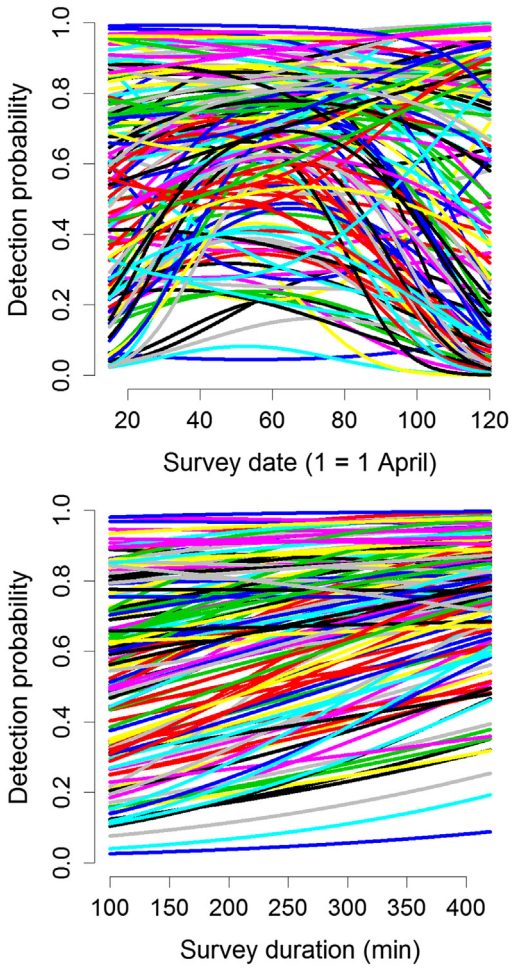

FIGURE 11.24

Species-specific predictions of detection probability as a function of survey date and survey duration in 1-km^2 quadrats in the Swiss breeding bird survey MHB in 2014 under community occupancy model number 10. Each line represents one of the 145 observed species. Note that one covariate is kept at the observed average for the computation of the prediction for the other covariate, and vice versa.

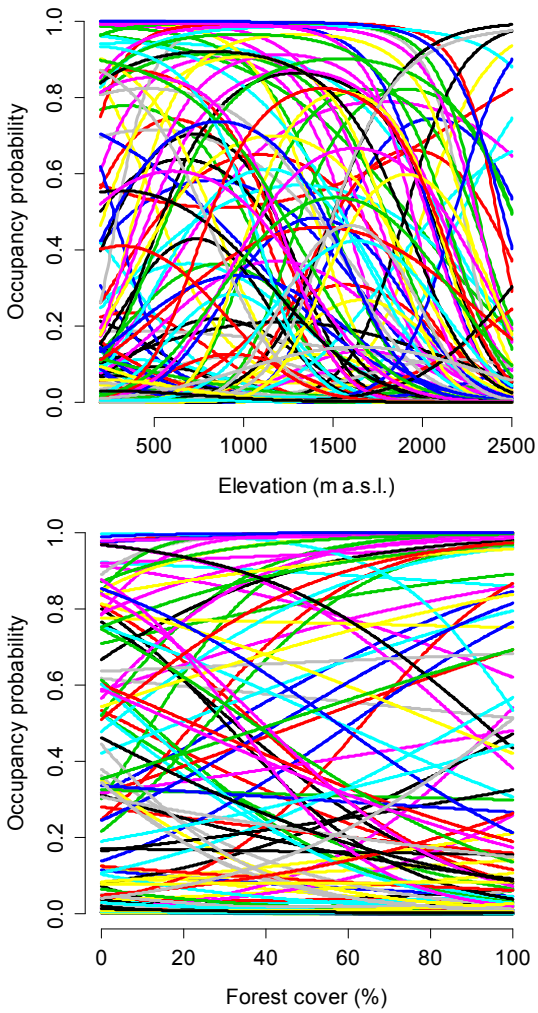

FIGURE 11.25

Species-specific predictions of occupancy probability as a function of elevation and forest cover in 1-km^2 quadrats in the Swiss breeding bird survey MHB in 2014 under community occupancy model number 10. Each line represents one of the 145 observed species. Note that one covariate is kept at the observed average for the computation of the prediction for the other covariate, and vice versa.

how the interspecies variability in the parameters that we have illustrated in the last few figures translates into variation among species in occupancy and detection probability.

```
# Predict detection for date and duration and occupancy for elevation and forest
# for each of the 145 observed species
predS <- array(NA, dim = c(500, nspec, 4))   # covariate value x species x
response, "S" for 'species'
```

```
p.coef <- cbind(lp=pm[1292:1436], betalp1 = pm[1:145], betalp2 = pm[216:360],
betalp3 = pm[431:575])
psi.coef <- cbind(lpsi=pm[1507:1651], betalpsi1 = pm[646:790], betalpsi2 = pm[861:1005],
betalpsi3 = pm[1076:1220])

for(i in 1:nspec){          # Loop over 145 observed species
    predS[,i,1] <- plogis(p.coef[i,1] + p.coef[i,2] * dat.pred +
      p.coef[i,3] * dat.pred^2 )          # p ~ date
    predS[,i,2] <- plogis(p.coef[i,1] + p.coef[i,4] * dur.pred) # p ~ duration
    predS[,i,3] <- plogis(psi.coef[i,1] + psi.coef[i,2] * ele.pred +
      psi.coef[i,3] * ele.pred^2 )        # psi ~ elevation
    predS[,i,4] <- plogis(psi.coef[i,1] + psi.coef[i,4] * for.pred) # psi ~ forest
}

# Plots for detection probability and survey date and duration (Fig. 11-24)
par(mfrow = c(1,2), cex.lab = 1.3, cex.axis = 1.3)
plot(o.dat, predS[,1,1], lwd = 3, type = 'l', lty = 1, frame = F,
   ylim = c(0, 1), xlab = "Survey date (1 = 1 April)",
   ylab = "Detection probability")
for(i in 2:145){
   lines(o.dat, predS[,i,1], col = i, lwd = 3)
}

plot(o.dur, predS[,1,2], lwd = 3, type = 'l', lty = 1, frame = F, ylim = c(0, 1), xlab = "Survey
duration (min)", ylab = "Detection probability")
for(i in 2:145){
   lines(o.dur, predS[,i,2], col = i, lwd = 3)
}
# Plots for occupancy probability and elevation and forest cover (Fig. 11-25)
par(mfrow = c(1,2), cex.lab = 1.3, cex.axis = 1.3)
plot(o.ele, predS[,1,3], lwd = 3, type = 'l', lty = 1, frame = F,ylim = c(0, 1),
   xlab = "Elevation (m a.s.l.)",ylab = "Occupancy probability")
for(i in 2:145){
   lines(o.ele, predS[,i,3], col = i, lwd = 3)
}

plot(o.for, predS[,1,4], lwd = 3, type = 'l', lty = 1, frame = F,ylim = c(0, 1), xlab = "Forest
   cover (%)", ylab = "Occupancy probability")
for(i in 2:145){
   lines(o.for, predS[,i,4], col = i, lwd = 3)
}
```

We see that the DR community occupancy models give us tremendous power to make inferences at all three levels involved in the sampling: metacommunity, community, and individual species. But there is more. One of the most exciting but perhaps underappreciated things about HMs such as this one is its estimate of the latent variable z, collected here in the presence/absence matrix $\boldsymbol{Z}$. We have used this matrix already to estimate alpha diversity (local species richness `Nsite`). However, we can use it for much more: to compare sites in terms of their occurring species (species turnover or beta diversity), and to compare species in terms of the sites at which they co-occur. We illustrate this next.

11.8 INFERENCES BASED ON THE ESTIMATED Z MATRIX: SIMILARITY AMONG SITES AND SPECIES

Our model provides us with a detection-error-corrected estimate of the true presence/absence matrix $\boldsymbol{Z}$ for every site and species, and with an estimate of the latent variable w for every species in the superpopulation of size M. Both enable computation of any presence/absence-based classical measure of alpha, beta, and gamma diversity. Summing up the values of $\boldsymbol{Z}$ over species for each site yields species richness at each site, which is the traditional measure of alpha diversity. Conversely, summing up vector w yields the total species richness in the area sampled by the study sites (metacommunity size); this is a measure of gamma diversity. For beta diversity, a very large number of measures have been proposed (Whittaker et al., 2001). One of them is the Jaccard index (J), which expresses the similarity of two sites in terms of their occurring species (Dorazio et al., 2011). For two sites r and s, J is the proportion of species shared among those species that occur either at r or s (note the momentary abuse of capital J in this section only):

$$J_{r,s} = \frac{\sum z_r z_s}{\sum z_r + \sum z_s - \sum z_r z_s},$$

where summations run over species. Values of this similarity index range from 0 to 1, corresponding respectively to the extremes of no shared species, and all species occurring at both sites. Instead of the similarity between two sites, we can choose to express the *dissimilarity* between two sites by taking $1 - J_{r,s}$. Similarly, we can express the similarity *between species* in terms of the sites where they co-occur by simply switching the dimensions in the above expressions. That is, we can compute the similarity between species r and species s in terms of the sites where they both occur, and where they occur alone, by applying this expression to the rows of the $\boldsymbol{Z}$ matrix. The summations in the expression are then over sites rather than species. For our last model, 10, we first have to format the MCMC samples for the $\boldsymbol{Z}$ matrix into a 3-dimensional array. After some inspection of the MCMC output, we find out that the 267×215 elements of the $\boldsymbol{Z}$ matrix are in rows 1937 through 59,341 of the vector `all10`, which we have computed already.

```
# Plug MCMC samples for full z matrix into 3D array
str(all10)
nsite <- 267
nspec <- 215
nsamp <- dim(all10)[1]      # 1200 MCMC samples
z <- array(NA, dim = c(nsite, nspec, nsamp))
Jacc <- array(NA, dim = c(nsite, nspec, nsamp))
for(j in 1:nsamp){       # Fill z matrix by column (default)
   cat(paste("\nMCMC sample", j, "\n"))
   z[,,j] <- all10[j, 1937:59341]
}

# Restrict computations to observed species
zobs <- z[,1:145,]      # Species 1 to 145

# Compute Jaccard index for sites and for species
Jsite <- array(NA, dim = c(nsite, nsamp))
Jspec <- array(NA, dim = c(145, nsamp))
```

We now compute the values of the two Jaccard indices for all pair-wise comparisons for any reference site or species that we like. For illustration, we here chose site 1 and species 13 (which is the sparrowhawk *Accipiter nisus*).

```
# Choose reference site and species for Jaccard indices
ref.site <- 1          # Just choose first site
ref.species <- 13      # European Sparrowhawk (check object 'obs.occ')

# Get posterior distributions for Jsite and Jspec (for references)
for(k in 1:nsamp){
  for(i in 1:nsite){ # Jaccard index for sites (in terms of shared species)
    Jsite[i,k] <- sum(zobs[ref.site,,k] * zobs[i,,k]) / (sum(zobs[ref.site,,k]) +
      sum(zobs[i,,k]) - sum(zobs[ref.site,,k] * zobs[i,,k]))
}
for(i in 1:(nspec-nz)){ # Jacc. index for species (in terms of shared sites)
  Jspec[i,k] <- sum(zobs[,ref.species,k] * zobs[,i,k]) / (sum(zobs[,ref.species,k]) +
    sum(zobs[,i,k]) - sum(zobs[,ref.species,k] * zobs[,i,k]))
}
}
# NA's arise when a site has no species or a species no sites

# Get posterior means, standard deviations and 95% CRI
# Jaccard index for sites, compared to reference site 1
pm <- apply(Jsite, 1, mean, na.rm = TRUE)      # Post. mean of Jsite wrt. site 1
psd <- apply(Jsite, 1, sd, na.rm = TRUE)       # Post. sd of Jsite wrt. site 1
cri <- apply(Jsite, 1, function(x) quantile(x, prob = c(0.025, 0.975), na.rm =
TRUE)) # CRI
cbind('post. mean' = pm, 'post. sd' = psd, '2.5%' = cri[1,], '97.5%' = cri[2,])
        post. mean     post. sd         2.5%        97.5%
 [1,]  1.00000000  0.00000000  1.00000000  1.00000000
 [2,]  0.56538646  0.03273553  0.50000000  0.63414634
 [3,]  0.34548127  0.02181290  0.30882353  0.39070786
 [4,]  0.51197748  0.03099446  0.45454545  0.57446809
 [5,]  0.42943330  0.02526238  0.38596491  0.48214286
 [6,]  0.60248009  0.03155804  0.54000000  0.66666667
 [7,]  0.54569316  0.03233974  0.48914689  0.61363636
 [8,]  0.36845865  0.02617650  0.32075472  0.41822727
 [9,]  0.52875245  0.03116781  0.47169811  0.59183673
[10,]  0.40707376  0.02977571  0.35294118  0.47058824
 [ output truncated ]
```

We can look at the table that expresses the similarity of the bird communities of all sites compared with site 1. As always with spatially referenced data, a map can be interesting, so we map the posterior means of these Jaccard indices (Figure 11.26). Looking at the Swiss geography in some other maps in this book (e.g., Figure 7.10), you will note that site 1 is most similar (in terms of the occurrence of bird species) to other lowland sites. And perhaps you will not be surprised, but still, it is nice to find a sensible result in this formal comparison of the bird communities among these sites.

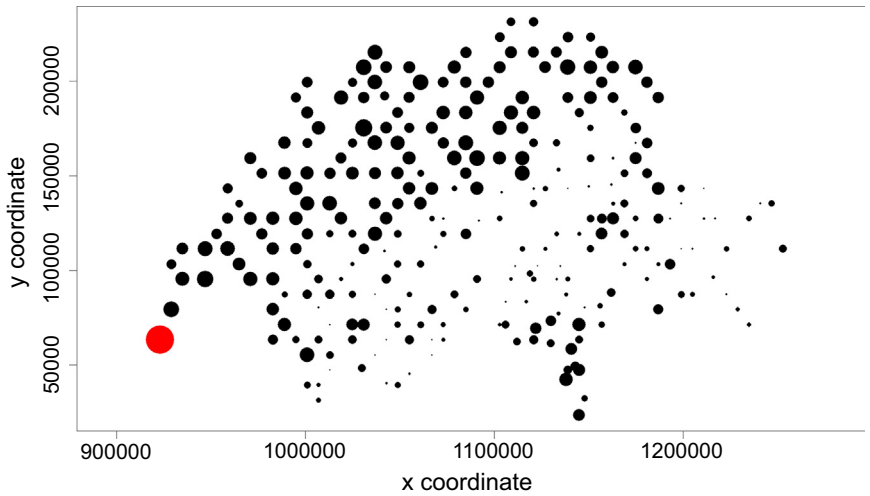

FIGURE 11.26

Map of the posterior means of the Jaccard site indices with respect to site 1 (shown in red; this is in the lowlands near Geneva). Circle size is proportional to the value of the Jaccard index, with red circle equal to 1.

```
# Make a map of Jaccard site indices (Fig. 11-26)
x <- 3          # size setting for plotting symbol
plot(data$coordx[1:267], data$coordy[1:267], xlab = "x coordinate", ylab = "y
coordinate", cex = x*pm, asp = 1, pch = 16)
points(data$coordx[which(pm == 1)], data$coordy[which(pm == 1)], cex = x*pm, col = "red",
pch = 16)
```

Turning to similarities among species in terms of presence/absence patterns, we compute posterior summaries for the 145 observed species and order them according to decreasing similarity in the next table.

```
# Jaccard index for species, compared with a reference species
# (species 13, European Sparrowhawk)
pm <- apply(Jspec, 1, mean, na.rm = TRUE)     # Post. mean of Jspec wrt. species 1
psd <- apply(Jspec, 1, sd, na.rm = TRUE)      # Post. sd of Jspec wrt. species 1
cri <- apply(Jspec, 1, function(x) quantile(x, prob = c(0.025, 0.975), na.rm =
TRUE)) # CRI
tmp <- cbind('post. mean' = pm, 'post. sd' = psd, '2.5%' = cri[1,], '97.5%' = cri[2,])
rownames(tmp) <- names(obs.occ)
print(tmp])           # print in systematic order
print(tmp[rev(order(tmp[,1])),]) # print in order of decreasing Jacc. values
```

```
                          post. mean      post. sd          2.5%       97.5%
Eurasian Sparrowhawk      1.000000000 0.000000000  1.000000000 1.00000000
Common Chaffinch          0.671351689 0.227396146  0.235251929 0.93308394
Common Blackbird          0.651996602 0.214434239  0.230418594 0.90376980
```

```
Eurasian Blackcap           0.650924688  0.208590706  0.240886364  0.90043290
Winter Wren                 0.646294658  0.217968460  0.232050947  0.89959878
European Robin              0.636539164  0.211018891  0.232380668  0.89270386
Black Redstart              0.631596397  0.210977120  0.227082359  0.91760300
Common Chiffchaff           0.627099631  0.199463573  0.237635548  0.86122449
Great Tit                   0.616446917  0.187948719  0.232227488  0.84719465
[ ... output truncated ... ]
Common Grasshopper Warbler  0.004857411  0.004743370  0.000000000  0.01695646
White Stork                 0.004850678  0.003766593  0.000000000  0.01408451
Common Pheasant             0.004783376  0.003576669  0.000000000  0.01398848
Savi's Warbler              0.004734415  0.003690696  0.000000000  0.01304634
Rook                        0.004598997  0.003595877  0.000000000  0.01351351
Northern Lapwing            0.004587117  0.003802500  0.000000000  0.01360777
Tawny Pipit                 0.004294793  0.004583141  0.000000000  0.01428571
Common Merganser            0.004178675  0.003455891  0.000000000  0.01265823
Golden Eagle                0.002372237  0.005180995  0.000000000  0.01105438

plot(1:145, tmp[rev(order(tmp[,1])),1]) # can also plot
```

There is a host of community analyses that are based on the presence/absence matrix and that you can now conduct for a corrected presence/absence matrix, thereby adjusting all your inferences for false-negative detection errors.

11.9 SPECIES RICHNESS MAPS AND SPECIES ACCUMULATION CURVES

We can further use the model output to make extrapolations in space. We show two such examples. First, we map the posterior predictive distribution of local species richness (always at the 1-km^2 scale) for all of Switzerland, and second, we compute and plot the species accumulation curve corrected for species presence/absence measurement error. This section is strongly based on code from Dorazio et al. (2006) and Dorazio et al. (2011). Remember our model for the occurrence of each species in superpopulation *M* (abbreviated slightly):

```
for(k in 1:M){              # Loop over 215 species
    w[k] ~ dbern(omega)     # Model for regional species pool
    for (i in 1:nsite) {    # Loop over sites: model for presence/absence Z
      logit(psi[i,k]) <- lpsi[k] + betalpsi1[k] * ele[i] + betalpsi2[k] * pow(ele[i],2) +
      betalpsi3[k] * forest[i]
      z[i,k] ~ dbern(w[k] * psi[i,k]) # Presence/absence Z
    }
}
```

We can use our estimates of the data augmentation variable (w) and of the community regression parameters (`lpsi[k]`, etc.) to make predictions in space, using the values of the covariates for a site. That is, for each species in the list of length *M*, we draw values of `w`, `lpsi`, `betalpsi1`, `betalpsi2`, and `betalpsi3` to obtain a sample from the posterior distribution of presence/absence z_{ik} of species k at every site i in a larger area, and then add up species and plot the result to produce a species richness map (as in Dorazio et al., 2011). We do this now for the Swiss landscape. To avoid a crash, we do this only for a sample of 50 posterior draws (of 1200).

```
# Get Swiss landscape data and standardise covariates as for model 10
library(unmarked)
data(Switzerland)
ch <- Switzerland
ELE <- (ch$elevation - mean.ele) / sd.ele
FOREST <- (ch$forest - mean.forest) / sd.forest

nsamp <- nrow(all10)                        # 1200 ..... far too many
nkm2 <- length(ch[[1]])                     # 42275, that's a LOT!
select.samp <- sort(sample(1:nsamp, 50))    # Chose random sample of 50
nsamp <- length(select.samp)                # new sample size 50

# Create posterior predictive distribution for Z for Swiss landscape
str( zCH <- array(NA, dim = c(nkm2, 215, nsamp)) )       # BIG array !
W <- all10[,1722:1936]                      # Grab MCMC samples from w
LPSI <- all10[,1507:1721]                   # Grab MCMC samples from logit(psi)
BETALPSI1 <- all10[,646:860]                # Grab MCMC samples from betalpsi1
BETALPSI2 <- all10[,861:1075]               # Grab MCMC samples from betalpsi2
BETALPSI3 <- all10[,1076:1290]              # Grab MCMC samples from betalpsi3
for(i in 1:nkm2){                           # takes about 5 mins !
   cat(paste("\nQuadrat", i, "\n"))
   for(u in 1:length(select.samp)){
      psi <- W[select.samp[u],] * plogis(LPSI[select.samp[u],] +
         BETALPSI1[select.samp[u],] * ELE[i] +
         BETALPSI2[select.samp[u],] * ELE[i]^2 +
         BETALPSI3[select.samp[u],] * FOREST[i] )
      zCH[i,,u] <- rbinom(215, 1, psi)
   }
}

# Compute posterior distribution of species richness by collapsing z array
SR <- apply(zCH, c(1,3), sum)     # posterior distribution
pmSR <- apply(SR, 1, mean)        # posterior mean
sdSR <- apply(SR, 1, sd)          # posterior standard deviation
```

We can now plot the posterior mean of species richness at a 1-km^2 scale in all of Switzerland based on the values of elevation and forest cover in the Swiss landscape, and on the parameters relating each individual species' occurrence with these covariates under our model 10. We see that the map is strongly influenced by the severe elevation gradient in Switzerland, and that the estimates are fairly precise (Figure 11.27). See Dorazio et al. (2010) for another such example of a species richness map that is corrected for measurement error.

```
library(raster)
par(mfrow = c(1,2), mar = c(2,2,3,5))
# Posterior mean map (code for more rudimentary plots only shown)
r1 <- rasterFromXYZ(data.frame(x = ch$x, y = ch$y, z = pmSR))
elev <- rasterFromXYZ(cbind(ch$x, ch$y,ch$elevation))
elev[elev > 2250] <- NA       # Mask areas > 2250 m a.s.l.
r1 <- mask(r1, elev)
```

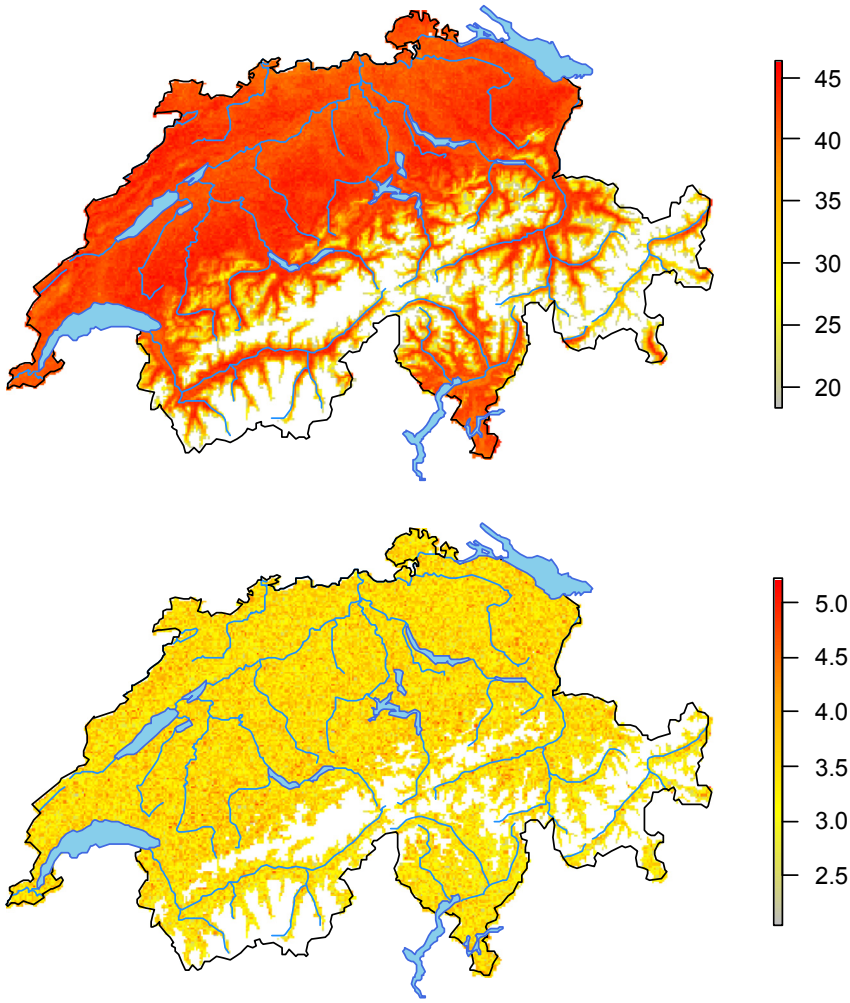

FIGURE 11.27

Breeding bird species richness at the 1-km^2 scale in Switzerland based on community occupancy model 10 for MHB survey data collected in 2014. Top: posterior mean; bottom: posterior standard deviation. Areas above 2250 m in elevation are masked (white).

```
mapPalette <- colorRampPalette(c("grey", "yellow", "orange", "red"))
plot(r1, col = mapPalette(100), axes = F, box = FALSE, main ="")

# Posterior standard deviation map (this code will only produce a more rudimentary map)
r2 <- rasterFromXYZ(data.frame(x = ch$x, y = ch$y, z = sdSR))
elev <- rasterFromXYZ(cbind(ch$x, ch$y,ch$elevation))
elev[elev > 2250] <- NA              # Mask areas > 2250 m a.s.l.
r2 <- mask(r2, elev)
plot(r2, col = mapPalette(100), axes = F, box = FALSE, main ="")
```

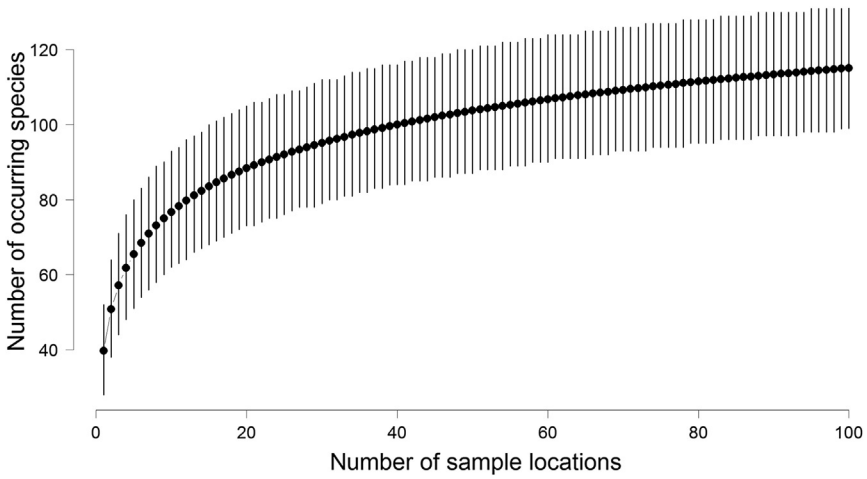

FIGURE 11.28

Species accumulation curve for Swiss breeding birds based on the analysis of the 2014 data harvested from the Swiss breeding bird survey MHB using a DR community occupancy model. These are predictions for 1-km^2 quadrats at the average values of the occupancy covariates elevation and forest cover. *(Code taken from Dorazio et al. (2006).)*

A final use of the parameter estimates from a DR community occupancy model that we illustrate is the computation of a detection-error-corrected species accumulation curve; that is, a curve showing the total number of occurring species in a sample of sites when the number of sites increases (Cam et al., 2002b,c). Such curves can be used, for instance, to standardize spatial sampling effort when comparing different sites. The following code is taken directly from the appendix in Dorazio et al. (2006). All computations are based on posterior samples of the data augmentation parameter `omega`, and on the mean and standard deviation of the prior distribution of the intercept of the occupancy model. We compute the theoretical relationship between the cumulative number of occurring species, and the number of sample sites for a random sample of sites with up to 100 sites. Since we do not account for covariate effects, our result applies at a covariate value of 0—i.e., for hypothetical 1-km^2 quadrats at 1189-m elevation and with 35% forest cover (Figure 11.28).

```
# Get 3,000 posterior samples of omega, and the mean and sd hyperparameters
omega <- out101$sims.list$omega
mu.lpsi <- out101$sims.list$mu.lpsi
str( sd.lpsi <- out101$sims.list$sd.lpsi )        # Confirms we have 3,000 draws

# compute posterior predictions of species occurrence probabilities
nsites <- 100
ndraws <- length(omega)
Nmax <- 215
psi <- matrix(NA, nrow=ndraws, ncol=Nmax)
for (i in 1:ndraws) {
   w <- rbinom(215, 1, omega[i])
   psi[i,] <- w * plogis(rnorm(Nmax, mean = mu.lpsi[i], sd=sd.lpsi[i]))
}
```

```
# compute posterior predictions of species presence at each site
z <- array(NA, dim=c(ndraws, Nmax, nsites))
for (i in 1:ndraws) {
  for (j in 1:Nmax) {
    z[i,j, ] <- rbinom(nsites, size=1, prob=psi[i,j])
  }
}

# compute posterior predictions of cumulative number of species present
Ntot <- matrix(NA, nrow=ndraws, ncol=nsites)
for (i in 1:ndraws) {
  for (j in 1:nsites) {
    zsum <- rep(NA, Nmax)
    if (j>1) {
      zsum <- apply(z[i, , 1:j], 1, sum)
    }
    else {
     zsum <- z[i, , 1]
    }
    Ntot[i,j] <- sum(zsum>0)
  }
}                              # takes about 4 min

# compute summary stats of species accumulation curve
nSpeciesPresent <- matrix(NA, nrow=3, ncol=nsites)
for (j in 1:nsites) {
  x <- Ntot[,j]
  nSpeciesPresent[1, j] <- mean(x)
  nSpeciesPresent[2:3, j] <- quantile(x, probs=c(0.05, 0.95))
}

# Plot species accumulation curve
ylim = c(min(nSpeciesPresent[2,]), max(nSpeciesPresent[3,]))
plot(1:nsites, nSpeciesPresent[1,], pch=16, ylim=ylim, type="b",
xlab="Number of sample locations", ylab="Number of occurring species",
las=1, cex.axis=1.2, cex.lab=1.5, cex=1.2, frame = F)
segments(1:nsites, nSpeciesPresent[2,], 1:nsites, nSpeciesPresent[3,])
```

11.10 COMMUNITY *N*-MIXTURE (OR DORAZIO/ROYLE/YAMAURA - DRY) MODELS

All community models in this chapter so far were "incidence-based," modeling the presence/absence of species or functions thereof, such as the number of observed species from detection/nondetection data. In this section, we extend the framework of the DR community occupancy model to become an "abundance-based" community *N*-mixture model. Interestingly, the history of this development has proceeded exactly along the same lines as that for single-species models, where the Royle-Nichols (RN) model in 2003 (Section 6.13) opened up the way from the simple Bernoulli-Bernoulli, or site-occupancy, mixture model toward more general HMs with abundance modeled as a latent state—i.e., the *N*-mixture models (Royle, 2004b). For community models, Yamaura et al. (2011) first extended the community occupancy models by adopting an RN formulation for the modeling of

species detection/nondetections. Then, Yamaura et al. (2012) modeled counts and developed a full community *N*-mixture community model, which we here call Dorazio/Royle/Yamaura (DRY) model. Such abundance-based community models are still in their infancy, and only very few papers have been published on them so far, including those by Chandler et al. (2013), Barnagaud et al. (2014), Beesley et al. (2014), Dorazio et al. (2015), Tobler et al. (2015), Sollmann et al. (in press), and Yamaura et al. (in press). However, we expect these to become more common soon.

The basic community *N*-mixture model with DA is really just a couple of trivial steps away from the community occupancy model (Section 11.7.2). It describes the relationship between latent abundance N_{ik} of species k at site i and the observed response y_{ijk}, which is the *count* of species k at site i during replicate j as follows.

1. Superpopulation process : $w_k \sim Bernoulli(\Omega)$
2. State process (abundance) : $N_{ik}|w_k \sim Poisson(w_k \lambda_k)$
3. Observation process (detection) : $y_{ijk}|N_{ik} \sim Binomial(N_{ik}, p_{ijk})$
4. Models of species heterogeneity : $\log(\lambda_{ik}) = beta0_k + beta1_k * elevation_i + \ldots$
   $\text{logit}(p_{ijk}) = alpha0_k + alpha1_k * date_{ij} + \ldots$

with

$$beta0_k \sim Normal(\mu_{beta0}, \sigma^2_{beta0})$$
$$beta1_k \sim Normal(\mu_{beta1}, \sigma^2_{beta1})$$
$$alpha0_k \sim Normal\left(\mu_{alpha0}, \sigma^2_{alpha0}\right)$$
$$alpha1_k \sim Normal\left(\mu_{alpha1}, \sigma^2_{alpha1}\right)$$

You can see that this is simply an *N*-mixture model for *M* species, where *M* may be either the total number of observed species or (as in the equations above) the number of species in the augmented data set, exactly analogous to what we have seen in Sections 11.6 and 11.7. Hence, in principle you can apply, inside a community model, everything you learned about *N*-mixture models in Chapters 6–9. For instance, Chandler et al. (2013) modeled removal counts, and Sollmann et al. (in press) modeled distance sampling counts, for a whole community of species. Below, we will be extending the model to contain a zero-inflation component. We start by organizing the 2014 MHB counts into a 3-D array that we call `yc`.

```
# Organize counts in 3D array: site x rep x species
COUNTS <- cbind(data$count141, data$count142, data$count143)     # Counts 2014
nsite <- 267                    # number of sites in Swiss MHB
nrep <- 3                       # number of replicate surveys per season
nspec <- length(species.list)  # 158 species occur in the 2014 data
yc <- array(NA, dim = c(nsite, nrep, nspec))
for(i in 1:nspec){
   yc[,,i] <- COUNTS[((i-1)*nsite+1):(i*nsite),]   # 'c' for counts
}
dimnames(yc) <- list(NULL, NULL, names(ordered.spec.name.list))
```

We very briefly look into the raw data to get an idea of the variability of these counts, among sites, among replicates, among species. You have to execute this and see yourself.

```
# Observed maximum and mean maximum count per species
tmp <- apply(yc, c(1,3), max, na.rm = TRUE)
```

```
tmp[tmp == -Inf] <- NA          # 1 quadrat with NA data in 2014
sort(round(meanmax <- apply(tmp, 2, mean, na.rm = TRUE), 3))    # mean of max
sort(obs.max.C <- apply(tmp, 2, max, na.rm = TRUE))             # max

# Plot observed species abundance distribution
plot(sort(meanmax), xlab = "Species number", ylab = "Mean maximum count")

# Spatio-temporal patterns in counts (mean over sites)
tmp <- apply(yc, c(2,3), mean, na.rm = TRUE)
matplot(log10(tmp+0.1), type = "l", lty = 1, lwd = 3, xlab = "MHB survey 1 - 3",
ylab = "log10 of mean count over sites", frame = F, cex.lab = 1.3, cex.axis = 1.3)

# Drop data from 13 species not observed in 2014
toss.out <- which(obs.max.C == 0)    # list of species not seen
yc <- yc[,,-toss.out]                # toss them out
obs.max.C <- obs.max.C[-toss.out]
nspec <- dim(yc)[3]                  # Redefine nspec as 145

# So here are our data
str(yc)
int [1:267, 1:3, 1:145] 0 0 0 0 0 0 0 0 0 0 ...
 - attr(*, "dimnames")=List of 3
  ..$ : NULL
  ..$ : NULL
  ..$ : chr [1:145] "Little Grebe" "Great Crested Grebe" "Grey Heron" ...
plot(table(yc)) # Extremely skewed distribution of observed counts
```

Hence, we model the observed 145 species only and do not data-augment. When developing the material here, we started with a model entirely analogous to the simplest community occupancy model in Section 11.6.1. This model did not pass a commonsense goodness-of-fit (CSGoF) test, since it yielded a flat rather than declining relationship between estimated community size (`Nsite`) and elevation (unlike in Figure 11.19). Adding in our standard set of covariates led to the desired decline of `Nsite` at higher elevations, but the overall level was about 14 species higher than were estimates from the comparable community occupancy model (model 10). Especially at high elevations, this does not seem plausible. So, we finally added zero-inflation for every species separately, and this model then did pass our CSGoF test. Compared with the algebra above, the ZIP-DRY community model has an additional hierarchical level that you can think of as the usual "suitability indicator," but it lacks the superpopulation process (the data augmentation variable w). We chose not to data augment to make the model simpler and quicker to converge. 'Quick' is relative, because it takes much longer to get this multispecies N-mixture model to converge than it does for the analogous occupancy model. However, the combination of DA and zero inflation would not pose any problems in principle.

2a. State process 1 (suitability, zero-inflation) : $a_{ik} \sim Bernoulli(\phi_k)$

2b. State process 2 (abundance) : $N_{ik}|a_{ik} \sim Poisson(a_{ik}\lambda_k)$

We made the model fully hierarchical by adopting community priors for all species-specific parameters except for the zero-inflation parameter ϕ_k.

```
# Bundle and summarize data set
str(win.data <- list(yc = yc, nsite = dim(yc)[1], nrep = dim(yc)[2],
nspec = dim(yc)[3], ele = ele, forest = forest, DAT = DAT, DUR = DUR))
List of 9
 $ yc : int [1:267, 1:3, 1:145] 0 0 0 0 0 0 0 0 0 0 ...
  ..- attr(*, "dimnames")=List of 3
  .. ..$ : NULL
  .. ..$ : NULL
  .. ..$ : chr [1:145] "Little Grebe" "Great Crested Grebe" "Grey Heron" ...
 $ nsite : int 267
 $ nrep  : int 3
 $ nspec : int 145
 $ ele   : num [1:267] -1.1539 -1.1539 -0.2175 -0.3735 -0.0614 ...
 $ forest: num [1:267] -1.1471 -0.4967 -0.0992 -0.9303 0.0092 ...
 $ DAT   : num [1:267, 1:3] -1.415 -1.19 -1.235 -0.559 -1.64 ...
 $ DUR   : num [1:267, 1:3] -0.43511 -0.78764 -0.52324 1.23944 0.00556 ...
```

In the model, we adopted suitably wide normal priors for all mean parameters, and uniform priors for all variance parameters. The presence/absence matrix $\boldsymbol{Z}$ is computed as a derived quantity from the posterior samples of the latent abundance variable, and then species richness can be tallied up in exactly the same way as before (and we could also compute similarity etc).

```
# Specify model in BUGS language
sink("model11.txt")
cat("
model{

# Community priors (with hyperparameters) for species-specific parameters
for(k in 1:nspec){
  phi[k] ~ dunif(0,1)                                  # Zero-inflation
  alpha0[k] ~ dnorm(mu.alpha0, tau.alpha0)             # Detection intercepts
  beta0[k] ~ dnorm(mu.beta0, tau.beta0)                # Abundance intercepts
  for(v in 1:3){
    alpha[k, v] ~ dnorm(mu.alpha[v], tau.alpha[v])  # Slopes detection
    beta[k, v] ~ dnorm(mu.beta[v], tau.beta[v])     # Slopes abundance
  }
}

# Hyperpriors for community hyperparameters
# abundance model
mu.beta0 ~ dunif(-1, 2)
tau.beta0 <- pow(sd.beta0, -2)
sd.beta0 ~ dunif(0, 3)
for(v in 1:3){
  mu.beta[v] ~ dunif(-1.5, 1)
  tau.beta[v] <- pow(sd.beta[v], -2)
}
sd.beta[1] ~ dunif(0, 3)
sd.beta[2] ~ dunif(0, 1.5)
sd.beta[3] ~ dunif(0, 1)
```

```
# detection model
mu.alpha0 ~ dunif(-2, 0)
tau.alpha0 <- pow(sd.alpha0, -2)
sd.alpha0 ~ dunif(0, 2)
for(v in 1:3){
  mu.alpha[v] ~ dunif(-0.5, 0.5)
  tau.alpha[v] <- pow(sd.alpha[v], -2)
}
sd.alpha[1] ~ dunif(0, 0.8)
sd.alpha[2] ~ dunif(0, 0.5)
sd.alpha[3] ~ dunif(0, 0.3)

# Ecological model for true abundance (process model)
for(k in 1:nspec){
  for (i in 1:nsite){
    a[i,k] ~ dbern(phi[k])              # zero-inflation
    N[i,k] ~ dpois(a[i,k] * lambda[i,k])
    log(lambda[i,k]) <- beta0[k] + beta[k,1] * ele[i] + beta[k,2] * pow(ele[i],2) +
      beta[k,3] * forest[i]
    # Compute presence/absence matrix z (for N > 0) from latent abundance
    z[i,k] <- step(N[i,k]-1)            # returns TRUE if N >= 1
  }
}

# Observation model for replicated counts
for(k in 1:nspec){
  for (i in 1:nsite){
    for (j in 1:nrep){
      yc[i,j,k] ~ dbin(p[i,j,k], N[i,k])
      logit(p[i,j,k]) <- alpha0[k] + alpha[k,1] * DAT[i,j] +
        alpha[k,2] * pow(DAT[i,j],2) + alpha[k,3] * DUR[i,j]
    }
  }
}

# Other derived quantities
for(k in 1:nspec){
  mlambda[k] <- phi[k] * exp(beta0[k])  # Expected abundance on natural scale
  logit(mp[k]) <- alpha0[k]             # Mean detection on natural scale
  Nocc.fs[k] <- sum(z[,k])              # Number of occupied sites among the 267
}
for (i in 1:nsite) {
  Nsite[i] <- sum(z[i,])                # Number of occurring species at each site
}
}
",fill = TRUE)
sink()

# Initial values
ast <- matrix(rep(1, nspec*nsite), nrow = nsite)
some.more <- 5           # May have to play with this until JAGS is happy
Nst <- apply(yc, c(1,3), max, na.rm = T) + some.more
```

```
Nst[Nst == '-Inf'] <- 20            # May have to play with this, too
Nst <- Nst
inits <- function()list(a = ast, N = Nst)

# OR: use inits at earlier solutions (greatly speeds up convergence)
pm <- out11$mean      # Pull out posterior means from earlier run
inits <- function() list(a = ast, N = Nst, alpha0 = rnorm(nspec), beta0 =
rnorm(nspec), alpha = matrix(rnorm(n = nspec*3), ncol = 3), beta =
matrix(rnorm(n = nspec*3), ncol = 3), mu.beta0 = pm$mu.beta0, sd.beta0 = pm$sd.beta0,
mu.beta = pm$mu.beta, sd.beta = pm$sd.beta, mu.alpha0 = pm$mu.alpha0, sd.alpha0 =
pm$sd.alpha0, mu.alpha = pm$mu.alpha, sd.alpha = pm$sd.alpha )

# Parameters monitored
params <- c("phi", "mp", "mlambda", "alpha0", "beta0", "alpha", "beta", "mu.beta0",
"sd.beta0", "mu.beta", "sd.beta", "mu.alpha0", "sd.alpha0", "mu.alpha", "sd.alpha",
"Nsite")

# MCMC settings
ni <- 180000 ; nt <- 90 ; nb <- 90000 ; nc <- 3

# Call JAGS from R, check convergence and summarize posteriors
out11 <- jags(win.data, inits, params, "model11.txt", n.chains = nc,
n.thin = nt, n.iter = ni, n.burnin = nb, parallel = FALSE)
par(mfrow = c(3,3)) ;traceplot(out11, c("mu.beta0", "sd.beta0", "mu.beta", "sd.beta",
"mu.alpha0", "sd.alpha0", "mu.alpha", "sd.alpha") )
print(out11, 2)
```

	mean	sd	2.5%	50%	97.5%	overlap0	f	Rhat	n.eff
mu.beta0	0.48	0.21	0.07	0.48	0.89	FALSE	0.99	1.00	598
sd.beta0	2.14	0.18	1.80	2.14	2.50	FALSE	1.00	1.00	3000
mu.beta[1]	-0.47	0.21	-0.91	-0.46	-0.08	FALSE	0.99	1.00	2173
mu.beta[2]	-0.60	0.11	-0.83	-0.60	-0.38	FALSE	1.00	1.00	3000
mu.beta[3]	-0.18	0.06	-0.30	-0.18	-0.07	FALSE	1.00	1.00	526
sd.beta[1]	2.27	0.17	1.98	2.26	2.66	FALSE	1.00	1.01	181
sd.beta[2]	1.13	0.10	0.94	1.12	1.34	FALSE	1.00	1.02	92
sd.beta[3]	0.60	0.05	0.51	0.60	0.71	FALSE	1.00	1.00	1473
mu.alpha0	-1.17	0.11	-1.39	-1.16	-0.96	FALSE	1.00	1.02	136
sd.alpha0	0.98	0.10	0.80	0.97	1.19	FALSE	1.00	1.02	120
mu.alpha[1]	0.04	0.04	-0.04	0.04	0.13	TRUE	0.84	1.00	1885
mu.alpha[2]	-0.11	0.04	-0.18	-0.11	-0.04	FALSE	1.00	1.00	1051
mu.alpha[3]	0.26	0.02	0.22	0.26	0.29	FALSE	1.00	1.04	53
sd.alpha[1]	0.44	0.04	0.37	0.44	0.52	FALSE	1.00	1.00	1186
sd.alpha[2]	0.34	0.03	0.28	0.34	0.41	FALSE	1.00	1.00	1441
sd.alpha[3]	0.14	0.02	0.10	0.14	0.18	FALSE	1.00	1.35	10

Convergence has not formally been achieved for one hyperparameter, but we have been running the model for a really long time and its traceplots do not look so bad, so we're still going to summarize the results from this run for now. Basically, we are able to do everything we did when summarizing the analysis of model 10. For instance, we can find that the average per-individual detection probability is only 0.27.

```
summary(p.sample <- plogis(rnorm(10^6, mean = -1.170, sd = 0.980)) )
hist(p.sample, breaks = 50, col = "grey", xlab = "Per-individual detection probability",
freq = FALSE)
```

For illustration, we plot the species-specific covariate relationships in the detection (Figure 11.29) and abundance parts of the model (Figure 11.30). Note that for predicting expected abundance, we have to multiply with the zero-inflation parameter. We will make use of the prediction covariates generated in the last section.

```
# Predict detection for date and duration and expected abundance for elevation and forest
# for each of the 145 observed species
predI <- array(NA, dim = c(500, nspec, 4))      # covariate value x species x
response, "I" for 'individual' (as opposed to 'species' in model 10)
pm <- out11$mean       # Grab posterior means from model 11
```

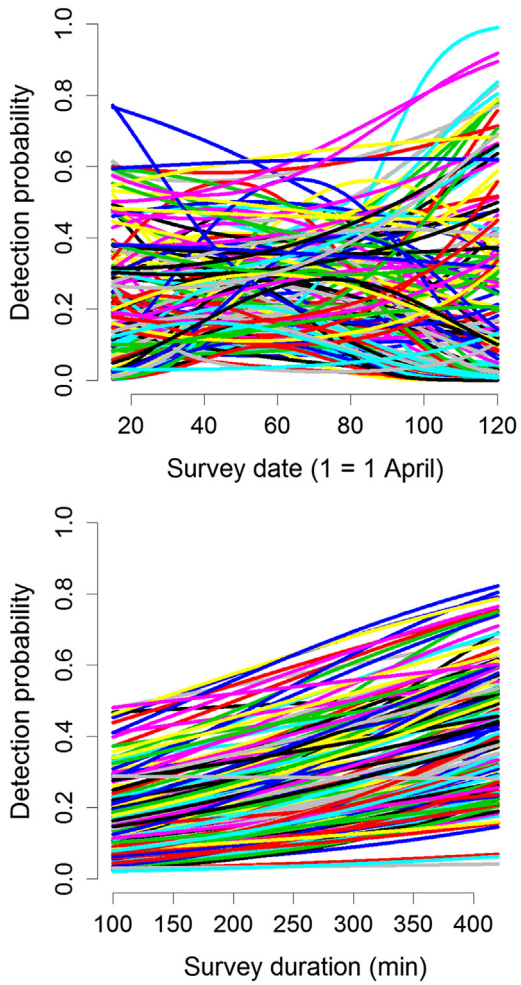

FIGURE 11.29

Species-specific predictions of detection probability (per individual) as a function of survey date and survey duration in 1-km^2 quadrats in the Swiss breeding bird survey MHB under community *N*-mixture model number 11. Each line represents one of the 145 species observed in 2014. Note that one covariate is kept at the observed average for the computation of the prediction for the other covariate, and vice versa.

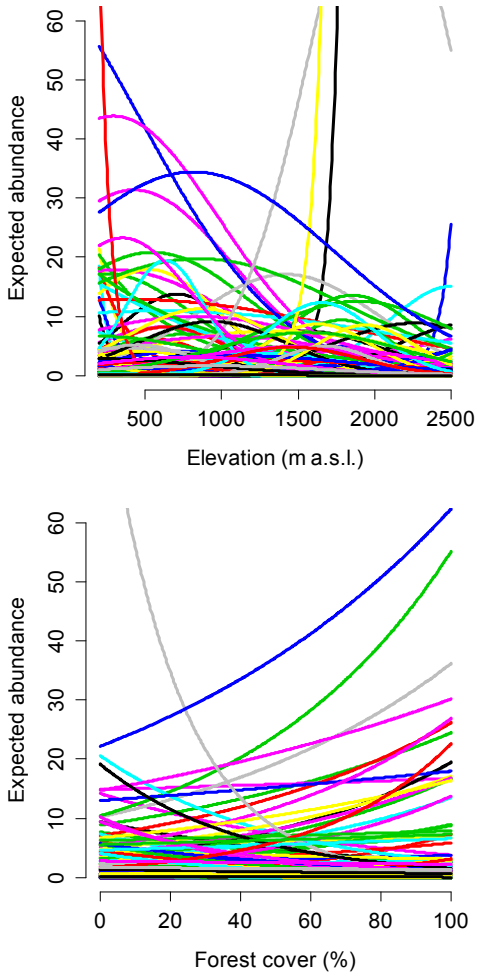

FIGURE 11.30

Species-specific predictions of expected abundance probability as a function of elevation and forest cover in 1-km^2 quadrats in the Swiss breeding bird survey MHB under community *N*-mixture model number 11. Each line represents one of the 145 species observed in 2014. Note that one covariate is kept at the observed average for the computation of the prediction for the other covariate, and vice versa.

```
for(i in 1:nspec){     # Loop over 145 observed species
   predI[,i,1] <- plogis(pm$alpha0[i] + pm$alpha[i,1] * dat.pred +
   pm$alpha[i,2] * dat.pred^2 )        # p ~ date
   predI[,i,2] <- plogis(pm$alpha0[i] + pm$alpha[i,3] * dur.pred)    # p ~ duration
   predI[,i,3] <- pm$phi[i] * exp(pm$beta0[i] + pm$beta[i,1] * ele.pred +
   pm$beta[i,2] * ele.pred^2 )         # E(N) ~ elevation
   predI[,i,4] <- pm$phi[i] * exp(pm$beta0[i] + pm$beta[i,3] * for.pred)  # E(N) ~ forest
}

# Plots for detection probability and survey date and duration (Fig. 11-29)
par(mfrow = c(1,2), cex.lab = 1.3, cex.axis = 1.3)
```

```
plot(o.dat, predI[,1,1], lwd = 3, type = 'l', lty = 1, frame = F, ylim = c(0, 1),
    xlab = "Survey date (1 = 1 April)", ylab = "Per-individual detection probability")
for(i in 2:145){
   lines(o.dat, predI[,i,1], col = i, lwd = 3)
}

plot(o.dur, predI[,1,2], lwd = 3, type = 'l', lty = 1, frame = F, ylim = c(0, 1),
    xlab = "Survey duration (min)", ylab = "Per-individual detection probability")
for(i in 2:145){
   lines(o.dur, predI[,i,2], col = i, lwd = 3)
}
# Plots for expected abundance and elevation and forest cover (Fig. 11-30)
par(mfrow = c(1,2), cex.lab = 1.3, cex.axis = 1.3)
plot(o.ele, predI[,1,3], lwd = 3, type = 'l', lty = 1, frame = F,
ylim = c(0, 60), xlab = "Elevation (m a.s.l.)",ylab = "Expected abundance")
for(i in 2:145){
   lines(o.ele, predI[,i,3], col = i, lwd = 3)
}

plot(o.for, predI[,1,4], lwd = 3, type = 'l', lty = 1, frame = F,
ylim = c(0, 60),     xlab = "Forest cover (%)", ylab = "Expected abundance")
for(i in 2:145){
   lines(o.for, predI[,i,4], col = i, lwd = 3)
}
```

Comparing these figures with those from the community occupancy model (Figures 11.24 and 11.25) we note broadly similar patterns. Obviously, the per-individual detection probability under the *N*-mixture model is lower than the per-species detection probability under the occupancy model (because P* = 1 − (1 − p)^N, where P* is the per-species detection probability, p the per-individual detection probability and N is local abundance). Therefore, the increase of p with increasing survey duration is much more striking under the *N*-mixture model.

11.11 SUMMARY AND OUTLOOK

We have given an overview of the community versions of site-occupancy models and *N*-mixture models: that is, the Dorazio/Royle (DR) community occupancy model (Dorazio and Royle, 2005; Dorazio et al., 2006) and the Dorazio/Royle/Yamaura (DRY) community abundance model (Yamaura et al., 2012; Chandler et al., 2013). In both models, spatial patterns in the presence/absence or the abundance of a community of species are described by a collection of elemental models for each individual species. Thus, these models are straightforward extensions to a community of species of any of the single-species models for occurrence and abundance that we presented in Chapters 6—10. Both the DR and the DRY model meet our main requirements for a flexible and robust community modeling framework: they allow us to make inferences at all levels of a metacommunity (species, community, metacommunity), and they explicitly deal with measurement error so that inferences can be made beyond just the sample of observed species, including estimation of species richness at multiple scales. Naturally, this object of inference includes the number of observed species at a point in time as well as the number of unobserved species. Only the DR/DRY class of models allows for this generality and breadth of inference.

These powerful community models can describe patterns, such as covariate relationships, at the level of the individual species, the local community (all species occurring at one site), and the whole

metacommunity—i.e., for the regional pool of species in the wider area of study. We have contrasted the DR/DRY community models with a much more simplistic approach that directly models emerging properties of a community, such as species richness, but that lacks the ability to make inferences about individual species, and we have seen that the DR community model was considerably more powerful. For instance, at the community level there was no measurable effect of forest cover on species richness, but at the level of individual species, almost half of the observed species were found to have a strong response to forest (31 species positive and 35 negative). In the Swiss breeding bird survey MHB, the average species in the metacommunity sampled had a detection probability of 0.63 at the level of a presence/absence measurement (i.e., in the occupancy model) and of 0.27 at the level of the abundance measurement for an individual (i.e., in the model for abundance). Therefore, by correcting for the resulting measurement error, arguably the DR and DRY community models had much reduced bias compared to other community models that do not contain a measurement error submodel.

We have also shown some further analyses that are possible after fitting a DR/DRY community model. Examples include the inspection of estimates for individual species, and extrapolation of species richness to a region such as the whole of Switzerland to produce a detection-corrected species richness map along with its associated uncertainty map (Dorazio et al., 2010). We also showed the computation of a species accumulation curve that corrects for imperfect and heterogeneous detection probability and that is insensitive to the ordering of sites (Dorazio et al., 2006). One estimand of particular interest is the presence/absence matrix $\boldsymbol{Z}$, which has been called the fundamental unit of analysis in biogeography and community ecology (McCoy and Heck, 1987). Most ecologists mistake the *observed* presence/absence patterns for the *true* presence/absence matrix, and therefore will obtain biased inferences due to presence/absence measurement error. In contrast, in the DR/DRY community models, we obtain an *estimate* of $\boldsymbol{Z}$ that is *corrected* for detection error. We can now do any calculation we would like on this estimate of $\boldsymbol{Z}$ to describe the size and composition of the metacommunity. As an illustration, we have computed Jaccard indices to compare sites in terms of occurring species, and to compare species in terms of the sites where they occur.

The community models in this chapter are joint species distribution models (JSDMs) in the sense that they simultaneously describe, in a single model, the occurrence or abundance of all species in a community. This simultaneous modeling has great advantages; for instance, we can formally compare species or even fit models to explain differences among species; see Section 11.6.3. In addition, estimates for information-poor rare species benefit from the richer information that comes from more common species, resulting in improved estimates for such species (Kéry and Royle, 2008; Zipkin et al., 2009). Also, we can easily compute derived quantities such as the number of occurring species with full propagation of the associated uncertainty. However, these models are based on the assumption that all species occur and are detected independently from one another given the modeled covariates. Clearly, this assumption will not always hold, and several recent lines of research have developed community models that do not make the independence assumption (Ovaskainen et al., 2010; Ovaskainen and Soininen, 2011; Clark et al., 2014; Dorazio and Connor, 2014; Pollock et al., 2014). This is exciting work. However, whether the more complex dependence models should be favored over simpler independence models will depend on many things, such as the spatial and temporal scale of a study, the taxonomic group, or even the specific set of species and the richness or sparsity of the data. For instance, it seems likely that the larger the grain (or spatial scale) of study, the more independent will the species occurrence or abundance appear to be. Similar reasoning also seems possible for the temporal scale: when analyzing data over larger temporal "observation windows," it would appear that observed dependencies in occurrence among species would be dampened. Finally, we would surmise that dependence is stronger for plants (e.g., trees, Clark et al., 2014) than for animals, because plants

compete very heavily along few environmental gradients (light, nutrients), while the ability to move would seem to greatly reduce competition among animal species by increasing the number of niches available at any given place.

In our analyses of breeding bird species data collected in 1-km^2 quadrats for a temporal window of two to three months, we think it unlikely that substantial dependencies would occur beyond what can be modeled by covariates. However, in principle, extending the DR/DRY models to contain species dependency appears simple: all we have to do is to add site-specific random effects to the linear predictor of occupancy or expected abundance for every species, and then specify a multivariate normal distribution as a prior for these "site residuals." The off-diagonals in the variance-covariance matrix then describe the positive or negative residual association for all pairs of observed species, where "residual" means "after all effects explained by the covariates in the model have been accounted for." We will cover multispecies models with (statistical) species interactions in Chapter 20 of volume 2.

We have presented two variants of community models, with and without data augmentation (DA). DA lets us make inferences about the whole metacommunity, including not only species that were detected but also species that were never detected. This allows one to estimate the full size of the metacommunity or the regional species pool. Also, it will reduce bias in inferences about the community that occur whenever a species trait is associated with detection probability. For instance, we found that there was a positive correlation between occupancy and detection probability in Swiss birds, and hence it is likely that the 145 observed species represent a biased sample from the entire Swiss avian metacommunity (because we are more likely to ever observe those with higher-than-average detection probability). Doing DA and modeling the entire metacommunity will ideally eliminate this bias. On the other hand, DR/DRY community models are naturally very parameter-rich and therefore costly in terms of computation time; DA usually *greatly* increases that cost. Moreover, we very rarely know what the metacommunity actually refers to in terms of an effective sample area, because none of these models is spatially explicit; see the discussion in Section 6.10. Hence, the estimated metacommunity size is often a largely hypothetical parameter and may therefore be of little practical interest. As a consequence, it may often be fine to apply a DR/DRY community model to the observed portion of a metacommunity only, as we have done for the DRY community *N*-mixture model.

Where will the development of DR/DRY and related community models go from here? First of all, we would like to think that the number of applications of these models will greatly increase. These models are just tremendously sensible mechanistically, and they offer powerful and multifaceted insights into communities at multiple scales. In addition, the required data are quite common, especially detection/nondetection data for the DR community model.

We have seen how easily we can move from a model for presence/absence data to a model for abundance, either for counts (Yamaura et al., 2012) or for detection/nondetection data combined with an RN model (Yamaura et al., 2011). Something in between might be a model with multiple states of occurrence—i.e., the community analog of a multistate occupancy model (Royle and Link, 2005; Nichols et al., 2007; MacKenzie et al., 2009)—see Fukaya and Royle (2013) for an example of such a model. In addition, in Chapters 6−10 we have presented a multitude of observation protocols for measuring presence/absence or abundance. All of these could be applied in a community context. For instance, it would seem to be easy to develop a community occupancy model for a removal or a time-to-detection protocol. Chandler et al. (2013) and Sollmann et al. (in press) have developed community *N*-mixture models for removal and distance sampling protocols, respectively, and the same could be done for any other multinomial mixture model discussed in Chapter 7.

One way of looking at a DR/DRY community model is that it simply represents the most sensible way of integrating the information about all species in a community or a subset thereof (Ovaskainen

and Soininen, 2011). Thus, whenever your aim is to characterize a community by some average over the species that occur in it, clustering the species together in a single analysis and specifying community hyperparameters to describe the mean and the variability among species seems a very sensible thing to do. This leads exactly to a DR/DRY type of community model, and not surprisingly, this has also been discovered independently (Gelfand et al., 2005) and rediscovered multiple times since 2005 (e.g., Ovaskainen et al., 2010; Ovaskainen and Soininen, 2011).

One area that is almost always underinvestigated is study design, and currently only Sanderlin et al. (2014) and Yamaura et al. (in press) have looked at DR and DRY community model design (also see McNew and Handel, 2015). Presumably, much may be deduced about community model study design from studies of single-species occupancy and *N*-mixture model study design (e.g., MacKenzie and Royle, 2005; Bailey et al., 2007; Guillera-Arroita et al., 2010, 2014b; McIntyre et al., 2012; Yamaura, 2013; Ellis et al., 2015), but there is certainly room for other and more focused studies. Perhaps the data simulation function and the analysis code given in this chapter may serve as a starting point.

One important difference between the DR/DRY community models and other otherwise similar community models recently developed (e.g., Gelfand et al., 2005; Ovaskainen et al., 2010; Clark et al., 2014; Pollock et al., 2014) is that the DR/DRY models contain an explicit description of the measurement error for occurrence or abundance. Unless this measurement error is small, all such inferences from community models will be biased to an unknown degree. However, DR/DRY community models only account for one of the two possible errors, false-negatives. False-negative errors are likely to be much more common than false-positive errors, especially in well-designed surveys like the Swiss MHB and for taxa such as birds, which are relatively species poor, comparatively easy to identify, and for which there are armies of well-trained volunteer observers in countries such as Switzerland. However, clearly there must be many situations in which species misidentifications can be a serious problem, especially in burgeoning Internet-based citizen-science programs that are now mushrooming all over the world, and where observer quality is extremely variable, generally unknown, and certainly often dubious. The consequences of false-positives on estimators of community models have not been studied. Some may be deduced from studies for single-species occupancy models (e.g., Miller et al., 2015), but the topic would warrant a thorough simulation study. Even more important would be the development of community models that can quantify and thereby correct for false-positives. However, this would appear to be a very difficult problem, because the basic DR/DRY model is already very parameter-rich and may be difficult to fit in practice. Adding complexity for false-positives would greatly increase the number of parameters that need to be estimated and exacerbate these problems.

It is intriguing to hypothesize about the interaction between false-positive errors in community studies and the perceived dependence in the *observed* occurrence of species (e.g., Clark et al., 2014; Pollock et al., 2014). We think it is possible that apparent dependency in the observed occurrence among species could be partly the result of misidentification. Falsely detecting a species A, which was in reality species B, is also the commission of a false-negative error for species B and, at the same time, should lead to understatement of occurrence probability for species B and vice versa for species A. This looks like a good research question: To investigate the mechanism underlying associations in the *observed* occurrence of species in a community; are they true (biological) or are they spurious, due to measurement error only, or perhaps both?

Spatial models for occurrence or abundance accommodate spatial autocorrelation and are increasingly used (e.g., Wintle and Bardos, 2006; Bled et al., 2011a,b; Yackulic et al., 2012; Broms et al., 2014; see also Chapters 21 and 22 in volume 2). Mattsson et al. (2013) developed a community model with autologistic formulation of spatial autocorrelation on occupancy, and presumably a conditional autoregressive (CAR) or related formulation would work as well (Johnson et al., 2013), or else

a spline model (Collier et al., 2012, see also Section 10.14). For single-species models, accommodating spatial autocorrelation by use of splines or CAR random effects can produce *much* better species distribution maps than do models that ignore spatial autocorrelation (Guélat and Kéry, in review). Hence, presumably maps of species richness would benefit similarly. But while in principle it would be straightforward to add autocorrelation terms for each species to the model, in practice, as for the modeling of false-positives, such models may just become too data hungry to be fittable in practice. However, this is clearly something that ought to be investigated.

Finally, an obvious extension is to accommodate time. All models in this chapter assumed closure. Dorazio et al. (2010) have extended the community model to include colonization/extinction dynamics, and so have Fukaya and Royle (2013), Henden et al. (2013), and Tobler et al. (2015). This is a straightforward extension conceptually, starting with the basic dynamic occupancy model for a single species (MacKenzie et al., 2003; Royle and Kéry, 2007) and using the same ideas of treating species-level parameters as fixed or random effects and of accounting or not for unseen species via DA, as we saw in Sections 11.6 and 11.7. We cover dynamic community models in Chapter 17 in volume 2.

EXERCISES

1. "Play community" by running the data simulation function `simComm` with changed function arguments, and observe how the latent state of presence/absence or abundance, and the observed data, are affected by your choice. Ideally, run a little simulation study to improve your intuition about metacommunity studies. Perhaps you can come up with some idea to even morph such play into a project for a paper.
2. In Sections 11.4 and 11.5.3 we mentioned that a multinomial mixture model would be intermediate between an *N*-mixture and a community site-occupancy model in the sense of keeping track of species identity over replicate surveys (unlike the former), though not across sites (unlike the latter). Use `unmarked` to fit a multinomial mixture model to the Swiss MHB data that we used in this chapter. You will not be able to incorporate any covariates into p, but do add the same covariates into "abundance" (which is the model part for local species richness in this application of the model) that we do for species occupancy in later sections of this chapter. Also fit the same multinomial mixture model with BUGS; there, you will be able to add any covariates into the model for p that you like.
3. In Section 11.6.1, turn model 5 (which has a binomial response) into a model with Bernoulli data distribution by fitting the Bernoulli variant of the same model to the disaggregated 3-D data array. Compare the timings between this and the binomial model 5.
4. In model 10, add into the detection model the effects of elevation and its interaction with date. The reason for this is that a survey date of, say, April 15, does not mean the same at an elevation of 250 and one of 2500 m. In the lowlands, this is the middle of spring, while at higher elevations, this is late winter.
5. At the start of Section 11.7.2, we mentioned that unmodeled detection heterogeneity among sites would likely bias low species richness estimators. Conduct a simulation study to investigate whether any bias results from site heterogeneity in occupancy probability. (Hint: you can use function `simComm`.)