Hierarchical Bayesian species distribution models with the **hSDM** R Package

July 1, 2014

Adansonia grandidieri Baill. next to Andavadoaka village (southwest Madagascar).

Ghislain Vieilledent[⋆,1] Cory Merow[2] Jérôme Guélat[3]
Andrew M. Latimer[4] Marc Kéry[3]
Alan E. Gelfand[5] Adam M. Wilson[6] Frédéric Mortier[1]
and John A. Silander Jr.[2]

[⋆] **Corresponding author:** \E-mail: ghislain.vieilledent@cirad.fr \Phone: +33.(0)4.67.59.37.51 \Fax: +33.(0)4.67.59.39.09
[1] **Cirad** – UPR BSEF, F–34398 Montpellier, France
[2] **University of Connecticut** – Department of Ecology and Evolutionary Biology, Storrs, CT 06269, USA
[3] **Swiss Ornithological Institute** – 6204 Sempach, Switzerland
[4] **University of California** – Department of Plant Sciences, Davis, CA 95616, USA
[5] **Duke University** – Department of Statistical Science, Durham, NC 27708, USA
[6] **Yale University** – Department of Ecology and Evolutionary Biology, New Haven, CT 06520, USA

Florebo quocumque ferar

"I will flower everywhere I am planted"

Abstract

Species distribution models (SDM) are useful tools to explain or predict species range from various environmental factors. SDM are thus widely used in conservation biology. Based on the observations of the species in the field (occurence or abundance data), SDM face two major problems which lead to bias in models' results: imperfect detection and spatial correlation of the observations.

At the present time, there is a lack of statistical tools to analyse large occurence or abundance data-sets (typically with tens of hundreds observation points) taking into account both imperfect detection and spatial correlation.

Here, we present the **hSDM** R package wich aims at providing user-friendly statistical functions to fill this gap. Functions were developped through a hierarchical Bayesian approach. They call a Metropolis-within-Gibbs algorithm coded in C to estimate model's parameters. Using compiled C code for the Gibbs sampler reduce drastically the computation time.

By making these new statistical tools available to the scientific community, we hope to democratize the use of more complex, but more realistic, statistical models for increasing knowledge in ecology and conserving biodiversity.

Keywords: R, C code, site-occupancy models, CAR process, spatial autocorrelation, biodiversity, SDM, niche modelling, detection probability, counts data, presence-absence, false absence, uncertainty, hierachical Bayesian models, Metropolis, MCMC, Gibbs sampler

CHAPTER 1

Introduction

1.1 Species distribution models

Biogeography is the study of the distribution of species over space and time and biogeographers try to understand the factors determining a species distribution (Smith, 1868; Wallace, 1876). A species distribution is often represented with a map (Wallace, 1876). This knowledge on the ecology of the species can be used for several applications such as conservation biology (Thuiller *et al.*, 2014).

Species distribution modelling (alternatively known as "environmental niche modelling", "ecological niche modelling", "predictive habitat distribution modelling", and "climate envelope modelling") refers to the process of using computer algorithms to predict the distribution of species in geographic space on the basis of a mathematical representation of their known distribution in environmental space (i.e. the realized ecological niche). The environment is in most cases represented by climate data (such as temperature, and precipitation), but other variables such as soil type and land cover can also be used. Species distribution models (SDM) allow estimating the probability of presence or abundance of a species on a large geographical range using a limited number of species observations (Elith & Leathwick, 2009; Guisan & Zimmermann, 2000). Species observations can be occurence data (presence-absence data or presence only data) or abundance data (also known as count data).

1.2 Imperfect detection and spatial correlation of the observations

When considering presence-absence or abundance data for species distribution modelling, strong assumptions are usually made (Araujo & Guisan, 2006; Guisan & Thuiller, 2005; Sinclair *et al.*, 2010). Among these assumptions, two can lead to biased estimates of species distribution. The first one deals with imperfect detection and the second one with spatial correlation of the observations.

Regarding imperfect detection, occurrence of a species is typically not observed perfectly. Species traits, survey-specific conditions and site-specific characteristics may influence species detection probability which is often < 1 (Chen *et al.*, 2013). Thus, observations might include false absences. For example, the habitat can be suitable and the species is present but individuals have not been seen during the census. Or the habitat can be suitable but the species has not dispersed yet to the site (typical example for plant species, see Latimer *et al.* (2006)) or was not present on the site at the moment of the observation (typical example for animal species such as birds, see Kéry *et al.* (2005)). Treating observed occurrence and species distributions as the true occurrence and distribution, failing to make amendments for imperfect detection, may lead to problems in species distribution studies, habitat models and biodiversity management (Kéry & Schmidt, 2008; Lahoz-Monfort *et al.*, 2014; Latimer *et al.*, 2006).

Regarding spatial correlation, most species present geographical patchiness (positive spatial autocorrelation). This pattern is often driven by multiple causes that may be associated to exogenous environmental factors such as climate or soil (which might be partly taken into account in species distribution models), but also to endogeneous biotic processes, called contagious processes, such as dispersal, migration, conspecific attraction or mortality which are rarely considered (Dormann *et al.*, 2007; Legendre, 1993; Lichstein *et al.*, 2002; Sokal & Oden, 1978). Due to the contagious biotic processes, the presence or abundance of a species at one site is influenced by the presence or abundance of the species at surrounding sites. A species might be present at a site where the environment is less suitable because of the presence of the species at neighbouring sites where the environment is higly suitable. Thus, ignoring spatial correlation may lead to biased conclusions about ecological relationships (Lichstein *et al.*, 2002) and even invert the slope of relationships from non-spatial analysis in some particular cases (Kühn *et al.*, 2006). In addition to its ecological significance, spatial autocorrelation is problematic for classical species distribution models which assume independently distributed errors (Dormann *et al.*, 2007; Legendre, 1993; Lichstein *et al.*, 2002).

1.3 Methods and software to account for imperfect detection and spatial correlation

New classes of models, called site-occupancy models (MacKenzie *et al.*, 2002) or zero inflated binomial (ZIB) models (Latimer *et al.*, 2006) for presence-absence data and N-mixture models (Royle, 2004) or zero inflated Poisson (ZIP) models for abundance data (Flores *et al.*, 2009), were developed to solve the problems created by imperfect detection. These models combine two processes, an ecological process which describes habitat suitability and an observation process which takes into account imperfect detection. Because they mix probability distributions to represent the suitability and observation processes, these models have also been called mixture models. Mixture models use information from repeated observations at several sites to estimate detectability. Detectability may vary with site characteristics (e.g., habitat variables) or survey characteristics (e.g., weather conditions), whereas suitability relates only to site characteristics.

One additional point regarding site-occupancy models is that they form a unifying framework for a very large array of capture-recapture models to estimate population size in animal ecology (Nichols, 1992): using parameter-expanded data augmentation (Royle *et al.*, 2007), most models for population size, survival, recruitment and similar demographic quantities (presented in detail in standard references such as Williams *et al.* (2002), Royle & Dorazio (2008) and Kéry & Schaub (2012)) can be cast into the framework of an occupancy model and this makes their fitting much easier.

Several studies have demonstrated the advantages of site-occupancy and N-mixture models over classical models which do not consider imperfect detection. These studies have focused on the distribution of various plant or animal species in marine and terrestrial ecosystems (see Chen *et al.* (2013); Latimer *et al.* (2006) for plants, Dorazio *et al.* (2006); Kéry *et al.* (2005); Rota *et al.* (2011); Royle (2004) for birds, Kéry *et al.* (2010) for insects, Bailey *et al.* (2004); Chelgren *et al.* (2011); MacKenzie *et al.* (2002) for amphibians, Monk (2014) for fishes, and Gray (2012); Poley *et al.* (2014) for mammals).

Several softwares can be used to fit site-occupancy and N-mixture models (Table 1.2). Some are based on the maximum likelihood approach (such as the widely used free Windows programs **MARK** and **PRESENCE** and the R package **unmarked**) while other are based on the hierarchical Bayesian approach (such as **WinBUGS** and **OpenBUGS** programs).

Softwares	Socc	Nmix	Sp	Approach	OS	Reference	URL
PRESENCE	1	1	0	ML	MS-W	MacKenzie (2006)	PRESENCE
MARK	1	1	0	ML	MS-W	White & Burnham (1999)	MARK
E-SURGE	1	0	0	ML	MS-W	Choquet *et al.* (2009)	E-SURGE
unmarked	1	1	0	ML	cross-platform	Fiske & Chandler (2011)	unmarked
stocc	1	0	1	Bayesian	cross-platform	Johnson *et al.* (2013)	stocc
JAGS	1	1	0	Bayesian	cross-platform		JAGS
Stan	1	1	0	Bayesian	cross-platform	Stan Development Team (2014)	Stan
WinBUGS	1	1	1	Bayesian	MS-W	Lunn *et al.* (2009)	WinBUGS
OpenBUGS	1	1	1	Bayesian	cross-platform	Lunn *et al.* (2009)	OpenBUGS
hSDM	1	1	1	Bayesian	cross-platform		hSDM

Table 1.2: **Softwares available for modeling species distribution including imperfect detection.**

A variety of methods have been developed to correct for the effects of spatial autocorrelation in species distribution models based on occurence or abundance data (Cressie & Cassie, 1993; Dormann *et al.*, 2007; Keitt *et al.*, 2002; Miller *et al.*, 2007). In their review article, Dormann *et al.* (2007) described six different statistical approaches to account for spatial autocorrelation: autocovariate regression; spatial eigenvector mapping; generalised least squares; autoregressive models and generalised estimating equations.

Several studies have demonstrated the advantages of these mehods focusing on a variety of plant or animal species (see Gelfand *et al.* (2005); Kühn *et al.* (2006); Latimer *et al.* (2006) for plants, Lichstein *et al.* (2002) for birds, and Johnson *et al.* (2013); Poley *et al.* (2014) for mammals).

Among the methods available to account for spatial autocorrelation, conditional autoregressive (CAR) models, which incorporate spatial autocorrelation through a neighbourhood structure, are commonly implemented in statistical softwares (Dormann *et al.*, 2007). The most commonly used softwares to implement CAR models are **OpenBUGS** and **WinBUGS** softwares (Lunn *et al.*, 2009) which have in-built functions (`car.normal` and `car.proper`) to describe the CAR process. CAR models can also be implemented in **BayesX** (Brezger *et al.*, 2005) and in the following R packages: **R-INLA** (Rue *et al.*, 2009), **CARBayes** (Lee, 2013), **stocc** (for binary data only), **spatcounts** (for count data only), **CARramps** (for Gaussian data only), and **spdep** (for Gaussian data only) (Table 1.4).

Softwares	Type of data	Approach	OS	Reference	URL
OpenBUGS	all	Bayesian	cross-platform	Lunn *et al.* (2009)	OpenBUGS
WinBUGS	all	Bayesian	MS-W	Lunn *et al.* (2009)	WinBUGS
BayesX	all	Bayesian	cross-platform	Brezger *et al.* (2005)	BayesX
R-INLA	all	Bayesian	cross-platform	Rue *et al.* (2009)	R-INLA
CARBayes	all	Bayesian	cross-platform	Lee (2013)	CARBayes
stocc	all	Bayesian	cross-platform	Johnson *et al.* (2013)	stocc
spatcounts	count	Bayesian	cross-platform		spatcounts
CARramps	Gaussian	Bayesian	cross-platform		CARramps
spdep	Gaussian	ML	cross-platform		spdep
hSDM	binomial and count	Bayesian	cross-platform		hSDM

Table 1.4: **Softwares available for modeling species distribution including spatial autocorrelation.**

1.4 Objectives of the hSDM R package

Among the available statistical programs, only **OpenBUGS** can be used on any operating system to fit both site-occupancy or N-mixture models including also a spatial autocorrelation process (Table 1.2 and Table 1.4). One problem is that **OpenBUGS**, for such models, cannot handle large data-sets (typically, data-sets with tens of thousands sites). Moreover, for smaller data-sets, models can be fitted but computation time can be long due to the fact that the **OpenBUGS** code is interpreted and not compiled. For this reason, we decided to develop the **hSDM** (for hierarchical Bayesian species distribution models) R package. The **stocc** R package (Johnson *et al.*, 2013; Poley *et al.*, 2014), which can handle binary data only, has been developed for the same reasons. The **hSDM** package allows the user to fit mixture models which take into account imperfect detection (site-occupancy, N-mixture, ZIB and ZIP models) and account for spatial autocorrelation. Spatial autocorrelation is represented through an intrinsic CAR process (Besag *et al.*, 1991). Functions in the **hSDM** R package use an adaptive Metropolis algorithm (Metropolis *et al.*, 1953; Robert & Casella, 2004) in a Gibbs sampler (Casella & George, 1992; Gelfand & Smith, 1990) to obtain the posterior distribution of model's parameters. The Gibbs sampler is written in C code and compiled to optimize computation efficiency. Thus, the **hSDM** package can be used for very large data-sets while reducing drastically the computation time.

In this vignette, we present examples to illustrate the use of the **hSDM** package in the R statistical environment (R Core Team, 2014). Examples use virtual or real data-sets. Results obtained with functions in the **hSDM** package are compared with the results obtained with other softwares and models.

CHAPTER 2

Occurence data

2.1 Binomial model

2.1.1 Mathematical formulation

Let's consider a random variable y_i representing the total number of presences of a species after several visits v_i at a particular site i. Random variable y_i can take values from 0 to v_i and can be assumed to follow a Binomial distribution having parameters v_i and θ_i (Eq. 2.1). Parameter θ_i can be interpreted as the probability of presence of the species at site i . Using a logit link function, θ_i can be expressed as a linear model combining explicative variables X_i and parameters β (Eq. 2.1).

$$
\begin{gathered}
y_i \sim \mathcal{B}inomial(v_i, \theta_i) \\
\text{logit}(\theta_i) = X_i \beta
\end{gathered}
\tag{2.1}
$$

Using this statistical model, we aim at representing a "suitability process". Given environmental variables X_i, how much is habitat at site i suitable for the species under consideration? Parameters β indicate how much each environmental variable contributes to the suitability process. Like every other function in the **hSDM** R package, function `hSDM.binomial()` estimates the parameters β of such a model in a Bayesian framework. Parameter inference is done using a Gibbs sampler including a Metropolis algorithm. The Gibbs sampler is coded in the C language to optimize computation efficiency.

2.1.2 Data generation

To explore the characteristics of the `hSDM.binomial()` function, we generate a virtual data-set on the basis of the Binomial model described above (Eq. 2.1). In the most general case, sites are visited once ($v_i = 1$). Thus, the random variable y_i follows a Bernoulli distribution of parameter θ_i and habitat characteristics X_i are fixed for site i. We generate a virtual data-set in this particular case. For data generation, we import virtual altitudinal data in R. Altitude is used as an explicative variable to determine habitat suitability, i.e. the probability of presence of a virtual species. Altitudinal data are loaded at the same time as the **hSDM** R package (data frame `altitude` in the working directory).

These data are transformed into a raster object using the function `rasterFromXYZ()` from the **raster** package. The raster has 2500 cells (50 columns and 50 rows) and the altitude ranges roughly between 100 and 600 m (Fig. 2.1). For linear models, explicative variables are usually centered and scaled to facilitate inference and interpretation of model parameters.

```
# Load altitudinal data and create raster
library(raster)
data(altitude,package="hSDM")
alt.orig <- rasterFromXYZ(altitude)
extent(alt.orig) <- c(0,50,0,50)
plot(alt.orig)
# Center and scale altitudinal data
alt <- scale(alt.orig,center=TRUE,scale=TRUE)
plot(alt)
```

A linear model including altitude (variable denoted A) is used to compute the probability of presence of the species (Eq. 2.2).

$$\begin{gathered} y_i \sim \mathcal{B}ernoulli(\theta_i) \\ \text{logit}(\theta_i) = \beta_0 + \beta_1 A_i \end{gathered} \tag{2.2}$$

We fix the parameters to $\beta_0 = -1$ and $\beta_1 = 1$. The species has a higher probability of presence at higher altitudes (Fig. 2.2).

```
# Load hSDM library
library(hSDM)
# Target parameters
beta.target <- matrix(c(-1,1),ncol=1)
# Matrix of covariates (including the intercept)
ncells <- ncell(alt)
X <- cbind(rep(1,ncells),values(alt))
# Probability of presence as a quadratic function of altitude
```

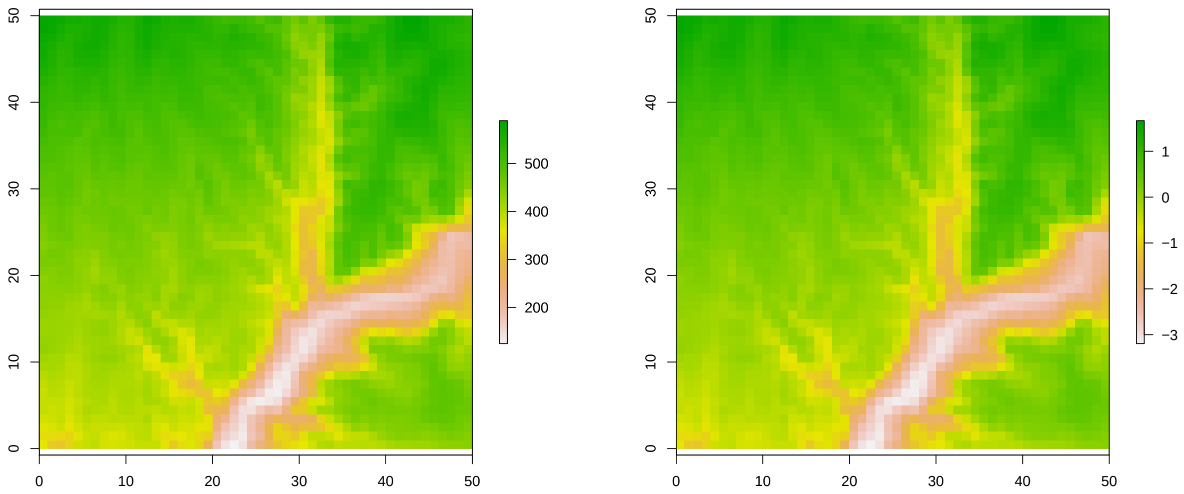


Figure 2.1: **Altitudinal data**. Original values (in m) on the left. Centered and scaled values on the right.

```
logit.theta <- X %*% beta.target
theta <- inv.logit(logit.theta)
# Coordinates of raster cells
coords <- coordinates(alt)
# Transform the probability of presence into a raster
theta <- rasterFromXYZ(cbind(coords,theta))
# Color palette for probability plots
colRP <- colorRampPalette(c("white","yellow","orange",
                            "red","brown","black"))
# Plot the probability of presence
brks <- seq(0,1,length.out=100)
arg <- list(at=seq(0,1,length.out=5), labels=c("0","0.25","0.5","0.75","1"))
nb <- length(brks)-1
plot(theta,main="Initial probabilities",col=colRP(nb),
     breaks=brks,axis.args=arg,zlim=c(0,1))
```

We can assume a number n of sites in the landscape where we have been able to observe or not the presence of the species. We can simulate the presence or absence of the species at these n sites given our model (Fig. 2.3).

```
# Number of observation sites
nsite <- 200
# Set seed for repeatability
seed <- 1234
```

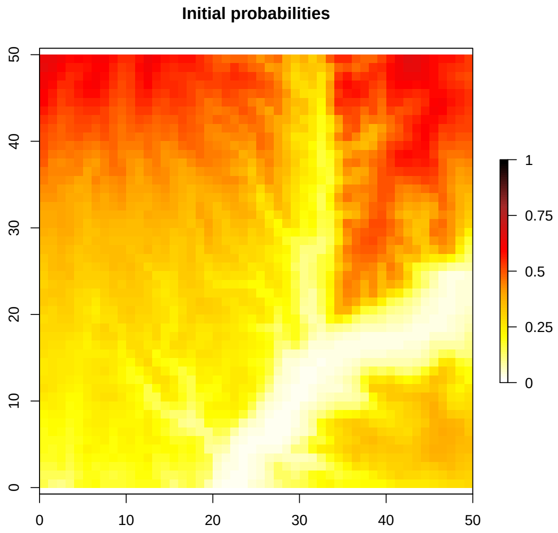


Figure 2.2: **Probability of presence**.

```
# Sample the observations in the landscape
set.seed(seed)
x.coord <- runif(nsite,0,50)
set.seed(2*seed)
y.coord <- runif(nsite,0,50)
library(sp)
sites.sp <- SpatialPoints(coords=cbind(x.coord,y.coord))
# Extract altitude data for sites
alt.sites <- extract(alt,sites.sp)
# Compute theta for these observations
X.sites <- cbind(rep(1,nsite),alt.sites)
logit.theta.site <- X.sites %*% beta.target
theta.site <- inv.logit(logit.theta.site)
# Simulate observations
visits <- rep(1,nsite) # One visit per site for the moment
set.seed(seed)
Y <- rbinom(nsite,visits,theta.site)
# Group explicative and response variables in a data-frame
data.obs.df <- data.frame(Y,visits,alt=X.sites[,2])
# Transform observations in a spatial object
data.obs <- SpatialPointsDataFrame(coords=coordinates(sites.sp),
                                   data=data.obs.df)
# Plot observations
plot(alt.orig)
points(data.obs[data.obs$Y==1,],pch=16)
```

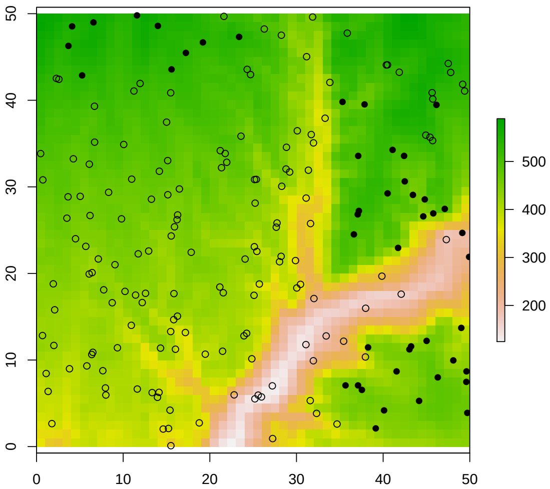


Figure 2.3: **Observation points**. Presences (full circles) and absences (empty circles) are localized on the altitude map (in m).

```
points(data.obs[data.obs$Y==0,],pch=1)
```

2.1.3 Parameter inference using the `hSDM.binomial()` function

The `hSDM.binomial()` function performs a Binomial logistic regression in a Bayesian framework. Before using this function we need to prepare a bit the data for predictions. We want to have predictions on the whole landscape, not only at observation points. To directly obtain these predictions, we can create a data frame including altitudinal data on the whole landscape. This data frame will be used for the `suitability.pred` argument. The data frame for predictions must include the same column names as those used in the formula for the `suitability` argument (i.e. “alt” our example).

```
data.pred <- data.frame(alt=values(alt))
```

We can now call the `hSDM.binomial()` function. Setting parameter `save.p` to 1, we can save in memory the MCMC values for predictions. These values can be used to compute several statistics for each predictions (mean, median, 95% quantiles). For example, mean and 95% quantiles are useful to estimate the uncertainty around the mean predictions.

```
mod.hSDM.binomial <- hSDM.binomial(presences=data.obs$Y,
                                   trials=data.obs$visits,
                                   suitability=~alt,
```

```r
                                   data=data.obs,
                                   suitability.pred=data.pred,
                                   burnin=1000, mcmc=1000, thin=1,
                                   beta.start=0,
                                   mubeta=0, Vbeta=1.0E6,
                                   seed=1234, verbose=1, save.p=1)
```

2.1.4 Analysis of the results

The `hSDM.binomial()` function returns an MCMC (Markov chain Monte Carlo) for each parameter of the model and also for the model deviance. To obtain parameter estimates, MCMC values can be summarized through a call to the `summary()` function from the **coda** package. We can check that the values of the target parameters, $\beta_0 = -1$ and $\beta_1 = 1$, are within the 95% confidence interval of the parameter estimates.

```r
summary(mod.hSDM.binomial$mcmc)

##
## Iterations = 1001:2000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                     Mean    SD Naive SE Time-series SE
## beta.(Intercept)  -1.413 0.226  0.00713         0.0223
## beta.alt           0.984 0.296  0.00936         0.0328
## Deviance         202.166 2.285  0.07225         0.1661
##
## 2. Quantiles for each variable:
##
##                     2.5%     25%     50%    75%   97.5%
## beta.(Intercept)  -1.843  -1.557  -1.413  -1.27  -0.958
## beta.alt           0.451   0.783   0.969   1.18   1.681
## Deviance         199.895 200.490 201.328 203.19 207.664
```

Parameters estimates can be compared to results obtained with the `glm()` function.

```r
#== glm results for comparison
mod.glm <- glm(cbind(Y,visits-Y)~alt,family="binomial",data=data.obs)
summary(mod.glm)
```

```
##
## Call:
## glm(formula = cbind(Y, visits - Y) ~ alt, family = "binomial",
##     data = data.obs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.129  -0.751  -0.604  -0.175   2.728
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.382      0.197   -7.03    2e-12 ***
## alt            0.952      0.276    3.44  0.00057 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 215.71  on 199  degrees of freedom
## Residual deviance: 199.79  on 198  degrees of freedom
## AIC: 203.8
##
## Number of Fisher Scoring iterations: 5
```

MCMC can also be graphically summarized with a call to the `plot.mcmc()` function, also in the **coda** package. MCMC are plotted with a trace of the sampled output and a density estimate for each variable in the chain (Fig. 2.4). This plot can be used to visually check that the chains have converged.

```
plot(mod.hSDM.binomial$mcmc)
```

The `hSDM.binomial()` function also returns two other objects. The first one, `theta.latent`, is the predictive posterior mean of the latent variable θ (the probability of presence) for each observation.

```
str(mod.hSDM.binomial$theta.latent)

##  num [1:200] 0.2191 0.0992 0.1038 0.1878 0.221 ...

summary(mod.hSDM.binomial$theta.latent)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0171  0.1540  0.2180  0.2300  0.2970  0.4970
```

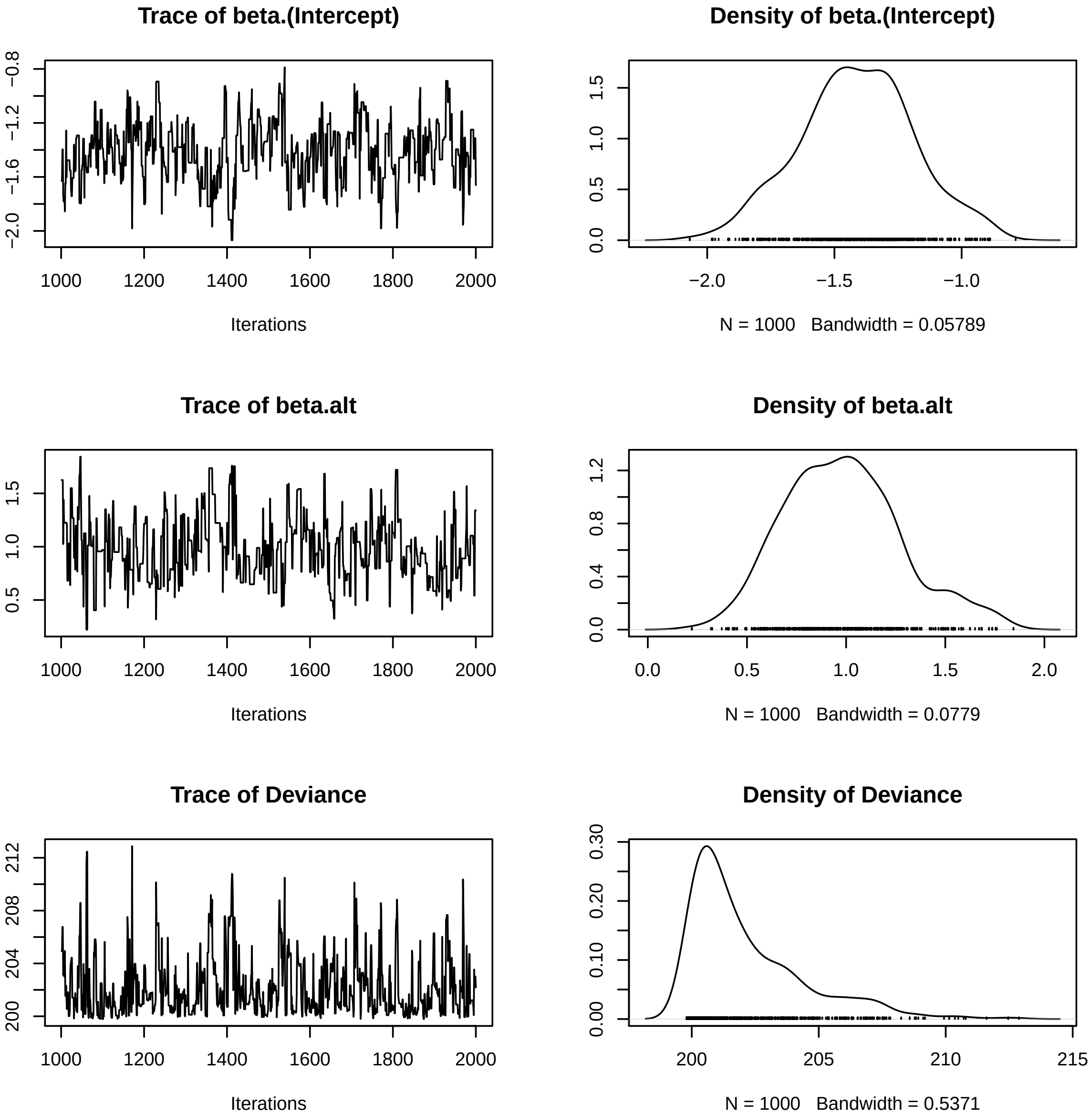


Figure 2.4: **Trace and density estimate for each variable of the MCMC.**

The second one, **theta.pred** is the set of sampled values from the predictive posterior (if parameter **save.p** is set to 1) or the predictive posterior mean (if **save.p** is set to 0) for each prediction. In our example, **save.p** is set to 1 and **theta.pred** is an **mcmc** object. Values in **theta.pred** can be used to plot the predicted probability of presence on the whole landscape and the uncertainty associated to predictions (Fig 2.5).

```r
# Create a raster for predictions
theta.pred.mean <- raster(theta)
# Create rasters for uncertainty
theta.pred.2.5 <- theta.pred.97.5 <- raster(theta)
# Attribute predicted values to raster cells
theta.pred.mean[] <- apply(mod.hSDM.binomial$theta.pred,2,mean)
theta.pred.2.5[] <- apply(mod.hSDM.binomial$theta.pred,2,quantile,0.025)
theta.pred.97.5[] <- apply(mod.hSDM.binomial$theta.pred,2,quantile,0.975)
# Plot the predicted probability of presence and uncertainty
plot(theta.pred.mean,main="Mean",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
plot(theta.pred.2.5,main="Quantile 2.5 %",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
plot(theta.pred.97.5,main="Quantile 97.5 %",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
```

In our example, we can compare the predictions to the initial probability of presence computed from our model to check that our predictions are correct (Fig. 2.6).

```r
# Comparing predictions to initial values
plot(theta[],theta.pred.mean[],cex.lab=1.4,xlim=c(0,1),ylim=c(0,1))
points(theta[],theta.pred.2.5[],cex.lab=1.4,col=grey(0.5))
points(theta[],theta.pred.97.5[],cex.lab=1.4,col=grey(0.5))
abline(a=0,b=1,col="red",lwd=2)
```

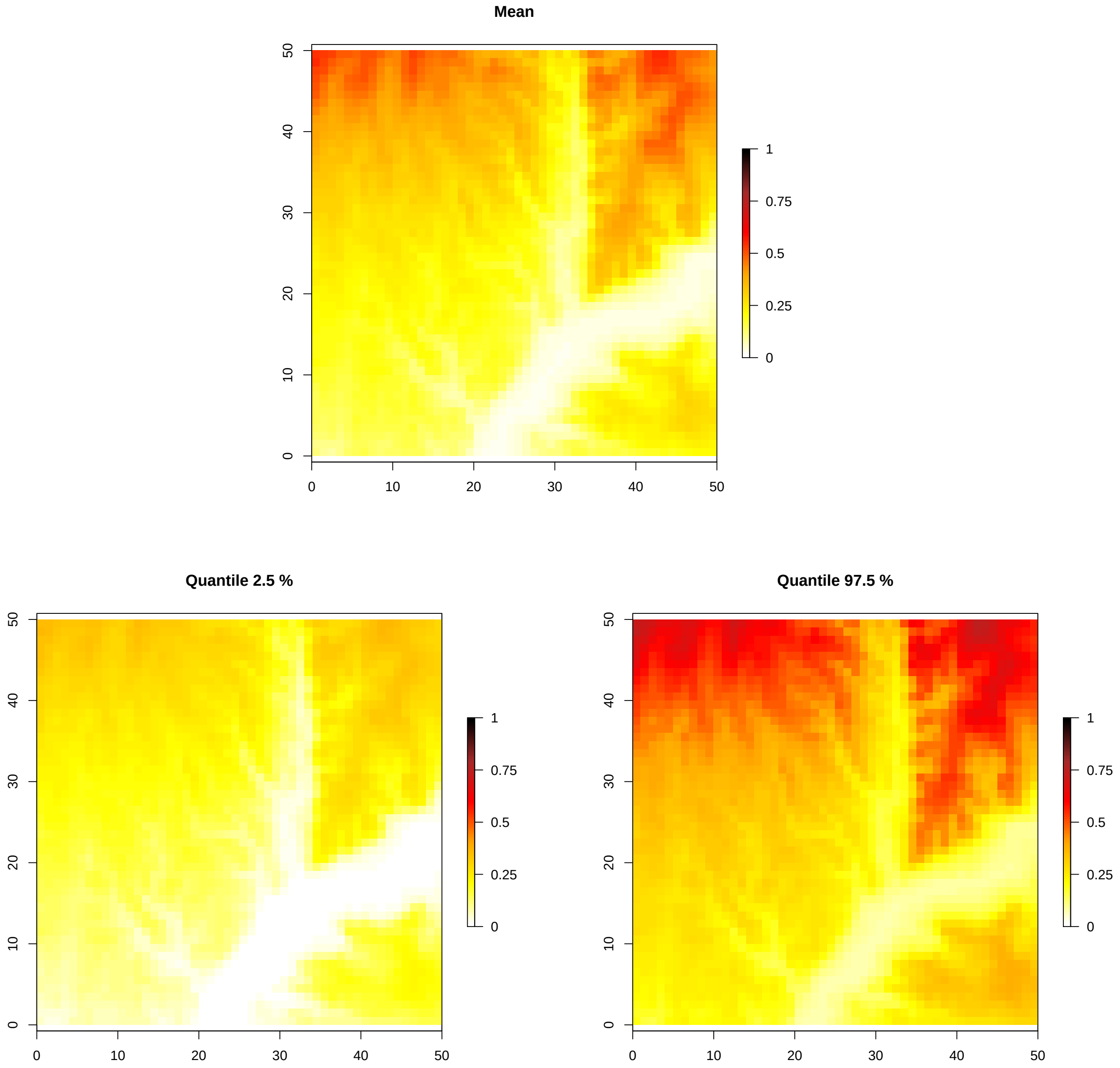


Figure 2.5: **Predicted probability of presence and uncertainty of predictions**. Mean probability of presence (top), predictions at 2.5% quantile (bottom left) and 97.5% quantile (bottom right) can be plotted from the `mcmc` object `plot.p.pred` returned by function `hSDM.binomial()`.

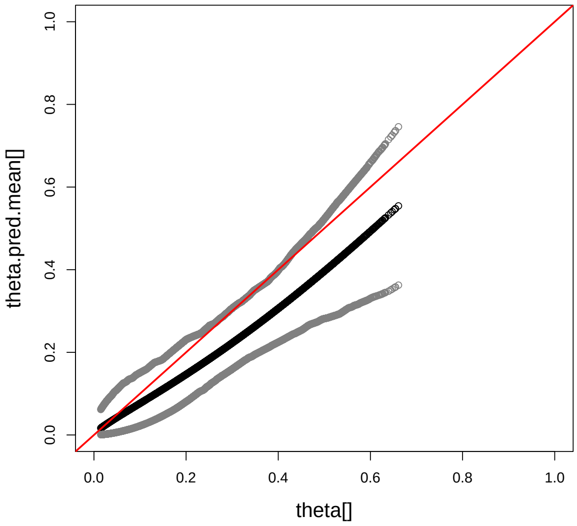


Figure 2.6: **Predicted vs. initial probabilities of presence**. Initial probabilities of presence are computed from the Binomial logistic regression model with target parameters.

2.2 Site-occupancy model

2.2.1 Mathematical formulation

Let's consider the random variable z_i describing habitat suitability at site i. The random variable z_i can take value 1 or 0 depending on the fact that the habitat is suitable ($z_i = 1$) or not ($z_i = 0$). Habitat at site i is described by environmental variables X_i. Random variable z_i can be assumed to follow a Bernoulli distribution of parameter θ_i (Eq. 2.3). In this case, θ_i is the probability that the habitat is suitable. Several visits at time t_1, t_2, etc., can occur at site i. Let's consider the random variable y_{it} representing the presence of the species at site i and time t. The species is observed at site i ($\sum_t y_{it} \geq 1$) only if the habitat is suitable ($z_i = 1$). The species is unobserved at site i ($\sum_t y_{it} = 0$) if the habitat is not suitable ($z_i = 0$), or if the habitat is suitable ($z_i = 1$) but the probability δ_{it} of detecting the species at site i and time t is inferior to 1. Thus, y_{it} is assumed to follow a Bernoulli distribution of parameter $z_i \delta_{it}$. Using a logit link function, δ_{it} can be expressed as a linear model combining explicative variables W_{it} and parameters γ (Eq. 2.3). Typically, explicative variables W_{it} are site characteristics (e.g., habitat variables) or survey characteristics (e.g., weather conditions). The function `hSDM.siteocc()` estimates the parameters β and γ of such a model.

$$\begin{gathered}
\textbf{Ecological process:}\\
z_i \sim \mathcal{B}ernoulli(\theta_i)\\
\text{logit}(\theta_i) = X_i\beta\\[1em]
\textbf{Observation process:}\\
y_{it} \sim \mathcal{B}ernoulli(z_i\delta_{it})\\
\text{logit}(\delta_{it}) = W_{it}\gamma
\end{gathered} \tag{2.3}$$

2.2.2 Data generation

To explore the characteristics of the `hSDM.siteocc()` function, we can generate a new virtual data-set on the basis of the site-occupancy model described above (Eq. 2.3). In the most general case, the observation protocol includes severals visits with varying survey conditions (e.g. weather conditions) to several sites with fixed sites characteristics (e.g. habitat variables). We will generate a virtual data-set following this protocole using the altitudinal data in the previous example for the Binomial model (Sec. 2.1).

We draw at random the number of visits at each site of the previous example (see Fig. 2.3 of Sec. 2.1).

```r
# Number of visits associated to each observation point
set.seed(seed)
visits <- rpois(nsite,lambda=3) # Mean number of visits ~3
# NB: Setting a too low mean number of visits per site (lambda < 3)
# leads to inaccurate parameter estimates
visits[visits==0] <- 1 # Number of visits must be > 0
# Vector of observation sites
sites <- vector()
for (i in 1:nsite) {
    sites <- c(sites,rep(i,visits[i]))
}
```

The survey conditions for each visit are determined by two explicative variables, w_1 and the altitude (variable denoted A). These two variables explain the observability of the species (Eq. 2.4).

$$\begin{gathered}
y_{it} \sim \mathcal{B}ernoulli(z_i\delta_{it})\\
\text{logit}(\delta_{it}) = \gamma_0 + \gamma_1 w_{1it} + \gamma_2 A_{it}
\end{gathered} \tag{2.4}$$

We fix the intercept and the effects of these two variables: $\gamma_0 = -1$, $\gamma_1 = 1$ and $\gamma_2 = -1$ for determining the detection probability. In our case, the detection probability decreases with altitude ($\gamma_2 < 0$).

```
# Explicative variables for observation process
nobs <- sum(visits)
set.seed(seed)
w1 <- rnorm(n=nobs,0,1)
W <- cbind(rep(1,nobs),w1,X.sites[sites,2])
# Target parameters for observation process
gamma.target <- matrix(c(-1,1,-1),ncol=1)
```

Using covariates and parameters for the two processes, we compute the probability that the habitat is suitable (θ_i) and the species detection probability (δ_i). We also draw the random variables z_i and y_i and construct the observation data-set.

```
# Ecological process (suitability)
logit.theta.site <- X.sites %*% beta.target
theta.site <- inv.logit(logit.theta.site)
set.seed(seed)
Z <- rbinom(nsite,1,theta.site)

# Observation process (detectability)
logit.delta.obs <- W %*% gamma.target
delta.obs <- inv.logit(logit.delta.obs)
set.seed(seed)
Y <- rbinom(nobs,1,delta.obs*Z[sites])

# Data-sets
data.obs <- data.frame(Y,w1,alt=X.sites[sites,2],site=sites)
data.suit <- data.frame(alt=X.sites[,2])
```

2.2.3 Parameter inference using the `hSDM.siteocc()` function

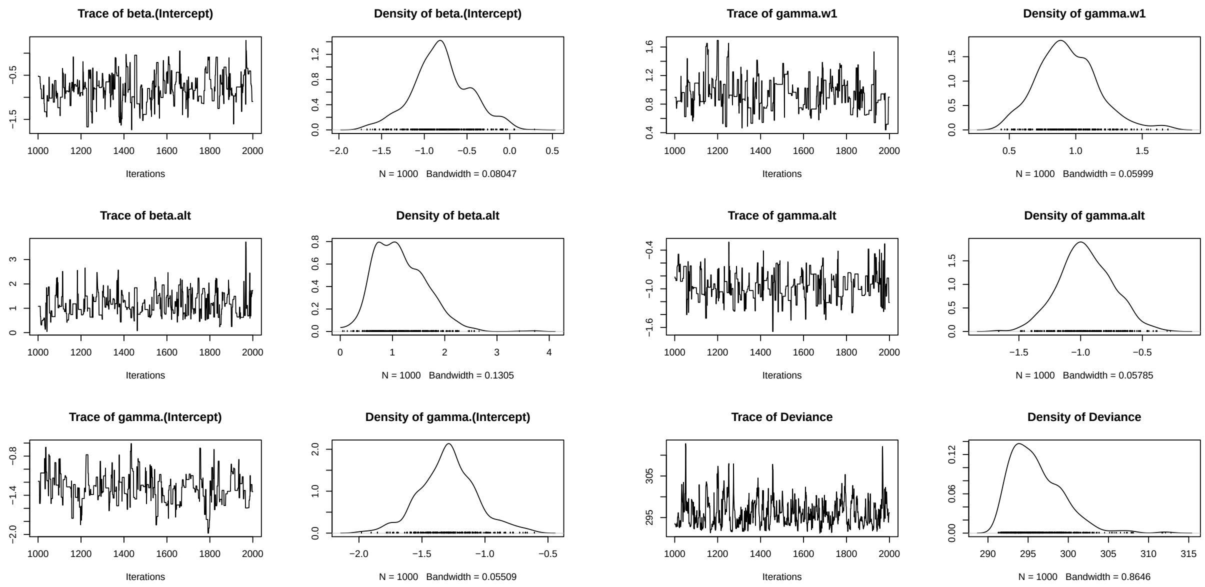
The `hSDM.siteocc()` function estimates the parameter of a site-occupancy model in a Bayesian framework.

```
mod.hSDM.siteocc <- hSDM.siteocc(# Observations
                                 presence=data.obs$Y,
                                 observability=~w1+alt,
                                 site=data.obs$site,
                                 data.observability=data.obs,
                                 # Habitat
                                 suitability=~alt,
                                 data.suitability=data.suit,
                                 # Predictions
                                 suitability.pred=data.pred,
                                 # Chains
                                 burnin=1000, mcmc=1000, thin=1,
                                 # Starting values
                                 beta.start=0,
                                 gamma.start=0,
                                 # Priors
                                 mubeta=0, Vbeta=1.0E6,
                                 mugamma=0, Vgamma=1.0E6,
                                 # Various
                                 seed=1234, verbose=1, save.p=1)
```

2.2.4 Analysis of the results

```
summary(mod.hSDM.siteocc$mcmc)

##
## Iterations = 1001:2000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean    SD Naive SE Time-series SE
## beta.(Intercept)   -0.801 0.336  0.01062         0.0316
## beta.alt            1.153 0.490  0.01550         0.0408
## gamma.(Intercept)  -1.292 0.226  0.00715         0.0245
## gamma.w1            0.938 0.227  0.00719         0.0210
## gamma.alt          -0.959 0.217  0.00687         0.0160
## Deviance          296.065 3.247  0.10269         0.2881
##
## 2. Quantiles for each variable:
##
##                       2.5%     25%     50%     75%   97.5%
## beta.(Intercept)   -1.477  -1.004  -0.811  -0.599  -0.089
## beta.alt            0.422   0.753   1.078   1.494   2.211
## gamma.(Intercept)  -1.753  -1.432  -1.288  -1.155  -0.778
## gamma.w1            0.531   0.775   0.925   1.077   1.462
## gamma.alt          -1.368  -1.105  -0.974  -0.810  -0.522
## Deviance          291.802 293.579 295.429 298.008 303.222
```



Trace of beta.(Intercept)

Density of beta.(Intercept)

Trace of gamma.w1

Density of gamma.w1

Trace of beta.alt

Density of beta.alt

Trace of gamma.alt

Density of gamma.alt

Trace of gamma.(Intercept)

Density of gamma.(Intercept)

Trace of Deviance

Density of Deviance

Figure 2.7: **Trace and density estimate for each variable of the MCMC.**

```
plot(mod.hSDM.siteocc$mcmc)
```

```
# Create a raster for predictions
theta.pred.mean <- raster(theta)
# Computing mean and quantiles for uncertainty
theta.pred.mean[] <- apply(mod.hSDM.siteocc$theta.pred,2,mean)
theta.pred.2.5 <- apply(mod.hSDM.siteocc$theta.pred,2,quantile,0.025)
theta.pred.97.5 <- apply(mod.hSDM.siteocc$theta.pred,2,quantile,0.975)
# Plot the predicted probability of presence
plot(theta.pred.mean,main="hSDM.siteocc",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
```

```
# Comparing predictions to initial values
plot(theta[],theta.pred.mean[],xlim=c(0,1),ylim=c(0,1),cex.lab=1.4)
points(theta[],theta.pred.2.5[],cex.lab=1.4,col=grey(0.5))
points(theta[],theta.pred.97.5[],cex.lab=1.4,col=grey(0.5))
abline(a=0,b=1,col="red",lwd=2)
```

Parameters estimates can be compared to results obtained with the `glm()` function assuming a perfect detection.

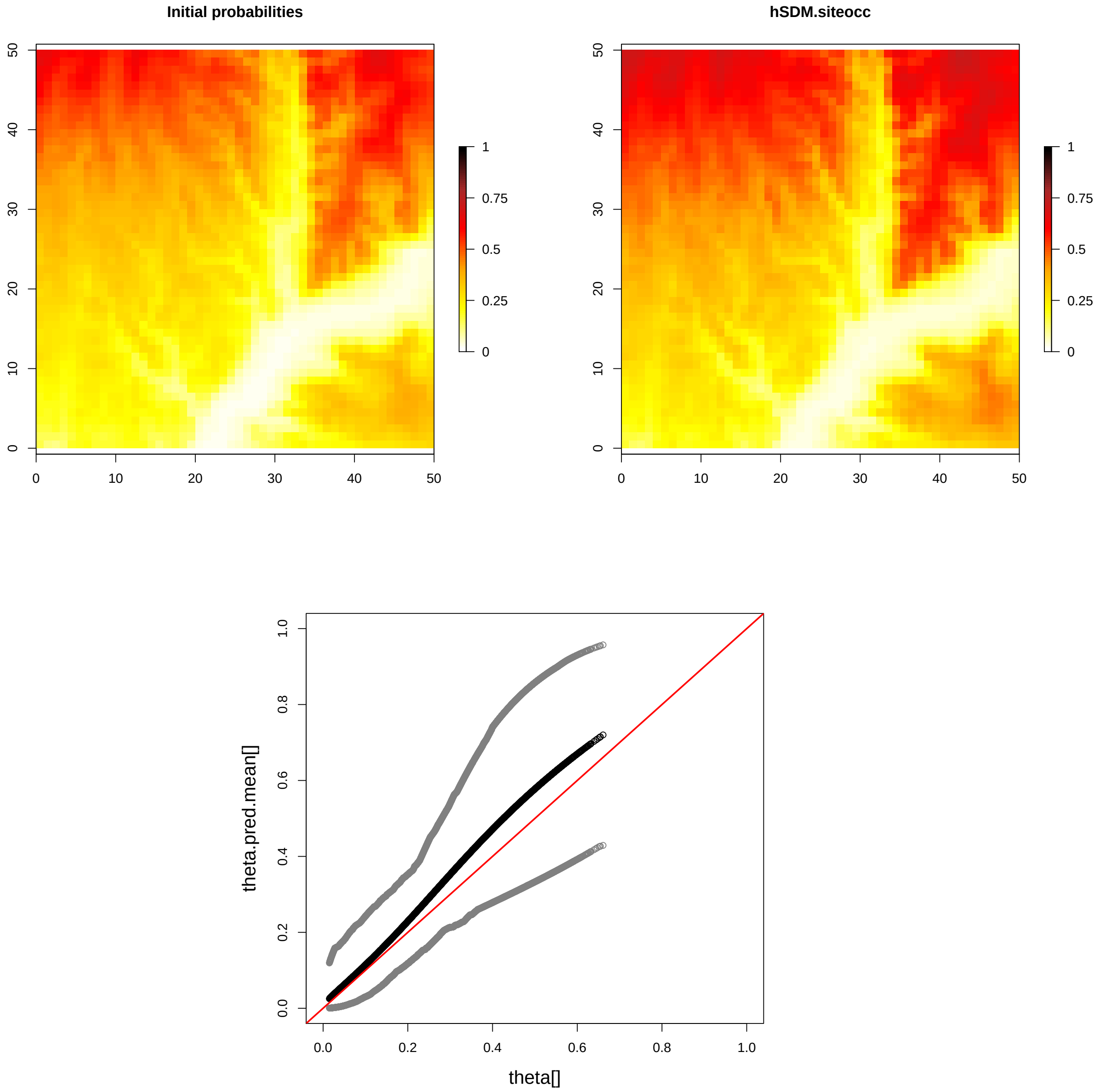


Figure 2.8: **Comparing predicted probability of presence with initial probabilities.**

```r
#== glm results for comparison
mod.glm <- glm(Y~alt,family="binomial",data=data.obs)
summary(mod.glm)

##
## Call:
## glm(formula = Y ~ alt, family = "binomial", data = data.obs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -0.528  -0.442  -0.427  -0.408   2.265
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -2.318      0.144  -16.06   <2e-16 ***
## alt           -0.133      0.129   -1.03      0.3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 362.10  on 594  degrees of freedom
## Residual deviance: 361.08  on 593  degrees of freedom
## AIC: 365.1
##
## Number of Fisher Scoring iterations: 5
```

```r
# Create a raster for predictions
theta.pred.glm <- raster(theta)
# Attribute predicted values to raster cells
theta.pred.glm[] <- predict.glm(mod.glm,newdata=data.pred,type="response")
# Plot the predicted probability of presence
plot(theta.pred.glm,main="GLM",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
```

```r
# Comparing predictions to initial values
plot(theta[],theta.pred.glm[],
     xlim=c(0,1),ylim=c(0,1),cex.lab=1.4)
points(theta[],theta.pred.mean[],col=grey(0.5))
abline(a=0,b=1,col="red",lwd=2)
```

On Figure 2.9, we can see that using a GLM in the case of imperfect detection can lead to very inaccurate parameter estimates and predictions for the probability of presence of the species. This is particularly true when detection probability is negatively correlated to presence probability (through an explicative variable such as the altitude in our example). This has been clearly demonstrated in an article by Lahoz-Monfort *et al.* (2014).

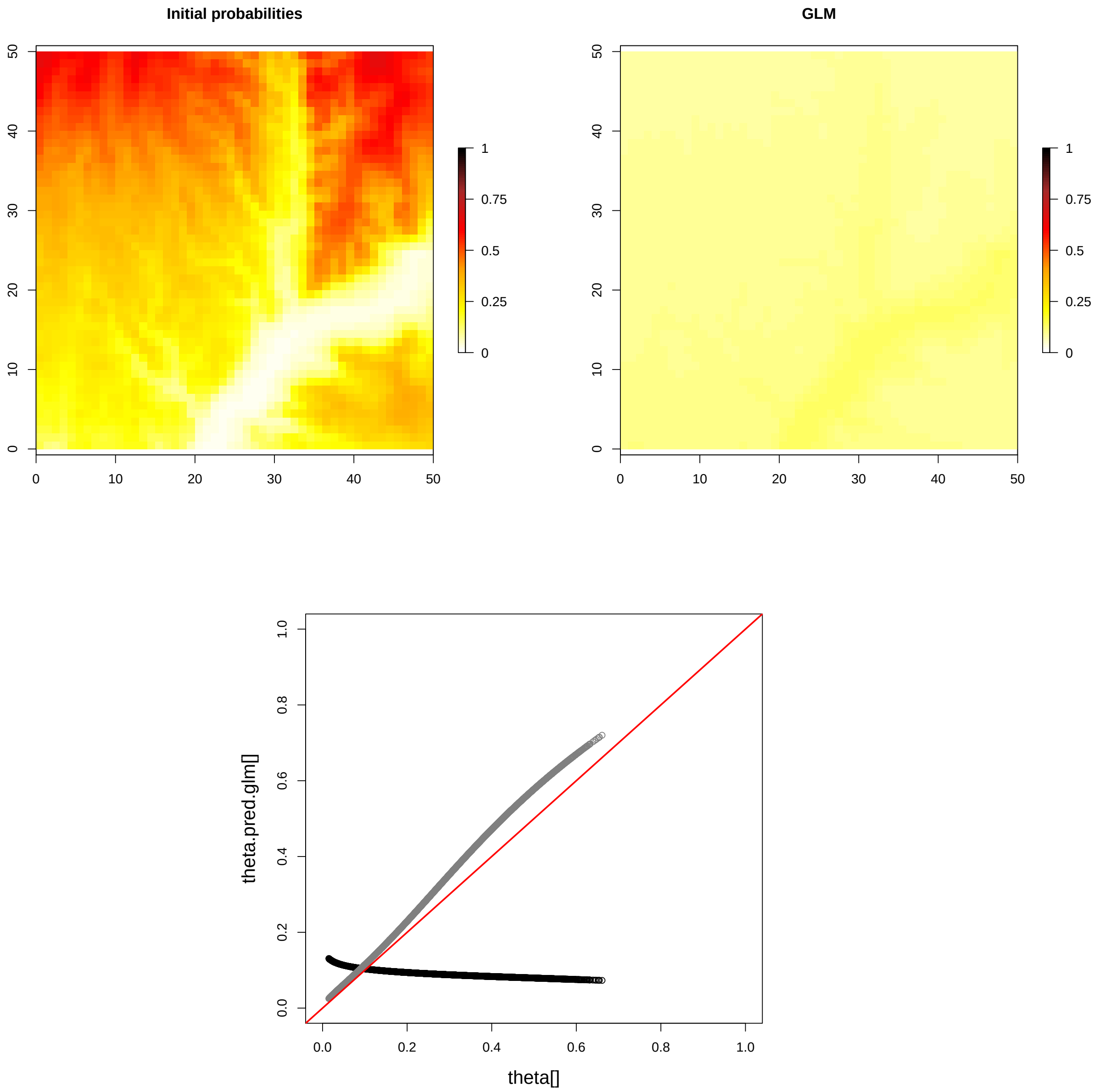


Figure 2.9: **Comparing predicted probability of presence using GLM with initial probabilities.** Grey dots figure the predictions with the `hSDM.siteocc()` function whereas black dots figure the prediction using the `glm()` function.

2.3 Binomial iCAR model

2.3.1 Mathematical formulation

2.3.2 Data generation with iCAR

```
# Rasters must be projected to correctly compute the neighborhood
crs(alt) <- '+proj=utm +zone=1'
# Neighborhood matrix
neighbors.mat <- adjacent(alt, cells=c(1:ncells), directions=8,
                          pairs=TRUE, sorted=TRUE)
# Number of neighbors by cell
n.neighbors <- as.data.frame(table(as.factor(neighbors.mat[,1])))[,2]
# Adjacent cells
adj <- neighbors.mat[,2]
# Generate symmetric adjacency matrix, A
A <- matrix(0,ncells,ncells)
index.start <- 1
for (i in 1:ncells) {
    index.end <- index.start+n.neighbors[i]-1
    A[i,adj[c(index.start:index.end)]] <- 1
    index.start <- index.end+1
}
```

```
# Function to draw in a multivariate normal
rmvn <- function(n, mu=0, V=matrix(1), seed=1234) {
    p <- length(mu)
    if (any(is.na(match(dim(V), p)))) {
        stop("Dimension problem!")
    }
    D <- chol(V)
    set.seed(seed)
    t(matrix(rnorm(n*p),ncol=p)%*%D+rep(mu,rep(n,p)))
}
```

```
# Generate spatial random effects
Vrho.target <- 5 # Variance of spatial random effects
d <- 1  # Spatial dependence parameter = 1 for intrinsic CAR
Q <- diag(n.neighbors)-d*A + diag(.0001,ncells) # Add small constant to
                                               # make Q non-singular
covrho <- Vrho.target*solve(Q) # Covariance of rhos
rho <- c(rmvn(1,mu=rep(0,ncells),V=covrho,seed=seed)) # Spatial Random Effects
rho <- rho-mean(rho) # Centering rhos on zero
```

```
rho.rast <- rasterFromXYZ(xyz=cbind(coords,rho))
# Probability of presence
theta.cells <- inv.logit(X %*% beta.target + rho)
theta <- rasterFromXYZ(cbind(coords,theta.cells))
```

```
# Ecological process (suitability)
cells <- extract(alt,sites.sp,cell=TRUE)[,1]
logit.theta.site <- X.sites %*% beta.target + rho[cells]
theta.site <- inv.logit(logit.theta.site)
set.seed(seed)
Y <- rbinom(nsite,visits,theta.site)
# Data-sets
data.suit <- data.frame(Y,visits,alt=X.sites[,2],cells)
data.pred <- data.frame(alt=values(alt),cell=c(1:ncells))
# Transform observations into a spatial object
data.suit <- SpatialPointsDataFrame(coords=coordinates(sites.sp),
                                    data=data.suit)
```

```
# Plot spatial random effects
plot(rho.rast,main="Spatial random effects")
# Plot initial probabilities and observations
plot(theta,main="Initial probabilities (iCAR model)",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
points(data.suit[data.suit$Y>0,],pch=16)
points(data.suit[data.suit$Y==0,],pch=1)
```

2.3.3 Parameter inference using the hSDM.binomial.iCAR() function

```
Start <- Sys.time() # Start the clock
mod.hSDM.binomial.iCAR <- hSDM.binomial.iCAR(presences=data.suit$Y,
                                             trials=data.suit$visits,
                                             suitability=~alt,
                                             spatial.entity=data.suit$cells,
                                             data=data.suit,
                                             n.neighbors=n.neighbors,
                                             neighbors=adj,
                                             suitability.pred=data.pred,
                                             spatial.entity.pred=data.pred$cell,
                                             burnin=5000, mcmc=5000, thin=5,
```

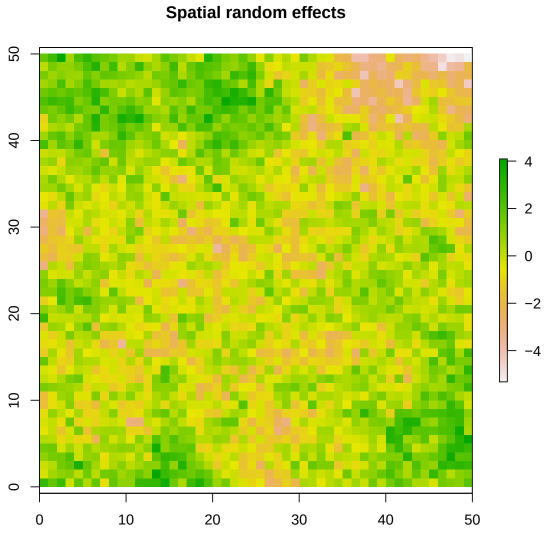


Figure 2.10: **Spatial random effects.**

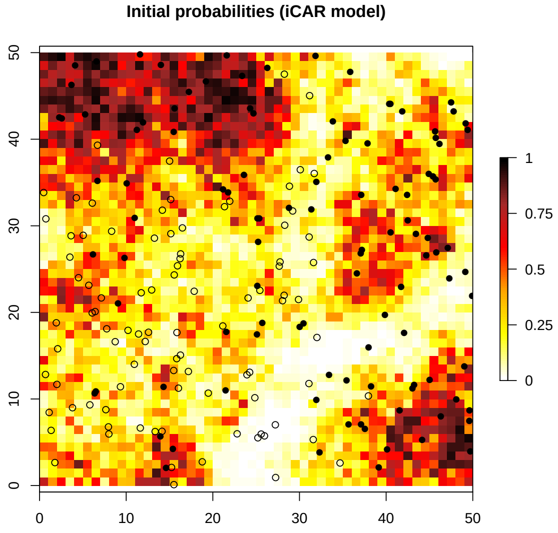


Figure 2.11: **Initial probability of presence and observations.** Presences (full circles) and absences (empty circles).

```
                                             beta.start=0,
                                             Vrho.start=1,
                                             priorVrho="1/Gamma",
                                             mubeta=0, Vbeta=1.0E6,
                                             shape=1, rate=1,
                                             Vrho.max=10,
                                             seed=1234, verbose=1,
                                             save.rho=1, save.p=0)
Time.hSDM <- difftime(Sys.time(),Start,units="sec") # Time difference
```

2.3.4 Analysis of the results with iCAR

```
summary(mod.hSDM.binomial.iCAR$mcmc)

##
## Iterations = 5001:9996
## Thinning interval = 5
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                     Mean     SD Naive SE Time-series SE
## beta.(Intercept)  -0.993  0.142  0.00449        0.00959
## beta.alt           0.750  0.194  0.00614        0.02230
## Vrho               2.800  0.980  0.03098        0.31537
## Deviance         327.835 15.114  0.47794        2.76813
##
## 2. Quantiles for each variable:
##
##                     2.5%     25%     50%     75%  97.5%
## beta.(Intercept)  -1.275  -1.095  -1.001  -0.891  -0.72
## beta.alt           0.413   0.609   0.736   0.877   1.16
## Vrho               1.361   2.031   2.647   3.420   5.04
## Deviance         297.871 317.162 327.325 338.720 356.81
```

```
# Predictions for spatial random effects
rho.pred <- apply(mod.hSDM.binomial.iCAR$rho.pred,2,mean)
rho.pred.rast <- rasterFromXYZ(cbind(coords,rho.pred))
plot(rho.pred.rast,main="Predictions rho")
# Predictions for probability of presence
```

```r
theta.pred <- mod.hSDM.binomial.iCAR$theta.pred
theta.pred.rast <- rasterFromXYZ(cbind(coords,theta.pred))
plot(theta.pred.rast,main="Predictions theta",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
# Predictions vs. initial spatial random effects
plot(rho[-cells],rho.pred[-cells],xlab="rho target",ylab="Predictions rho")
points(rho[cells],rho.pred[cells],col="blue",pch=16)
abline(a=0,b=1,col="red")
# Predictions vs. initial probabilities
plot(values(theta)[-cells],theta.pred[-cells],xlab="theta target",
     ylab="Predictions theta")
points(values(theta)[cells],theta.pred[cells],col="blue",pch=16)
abline(a=0,b=1,col="red")
```

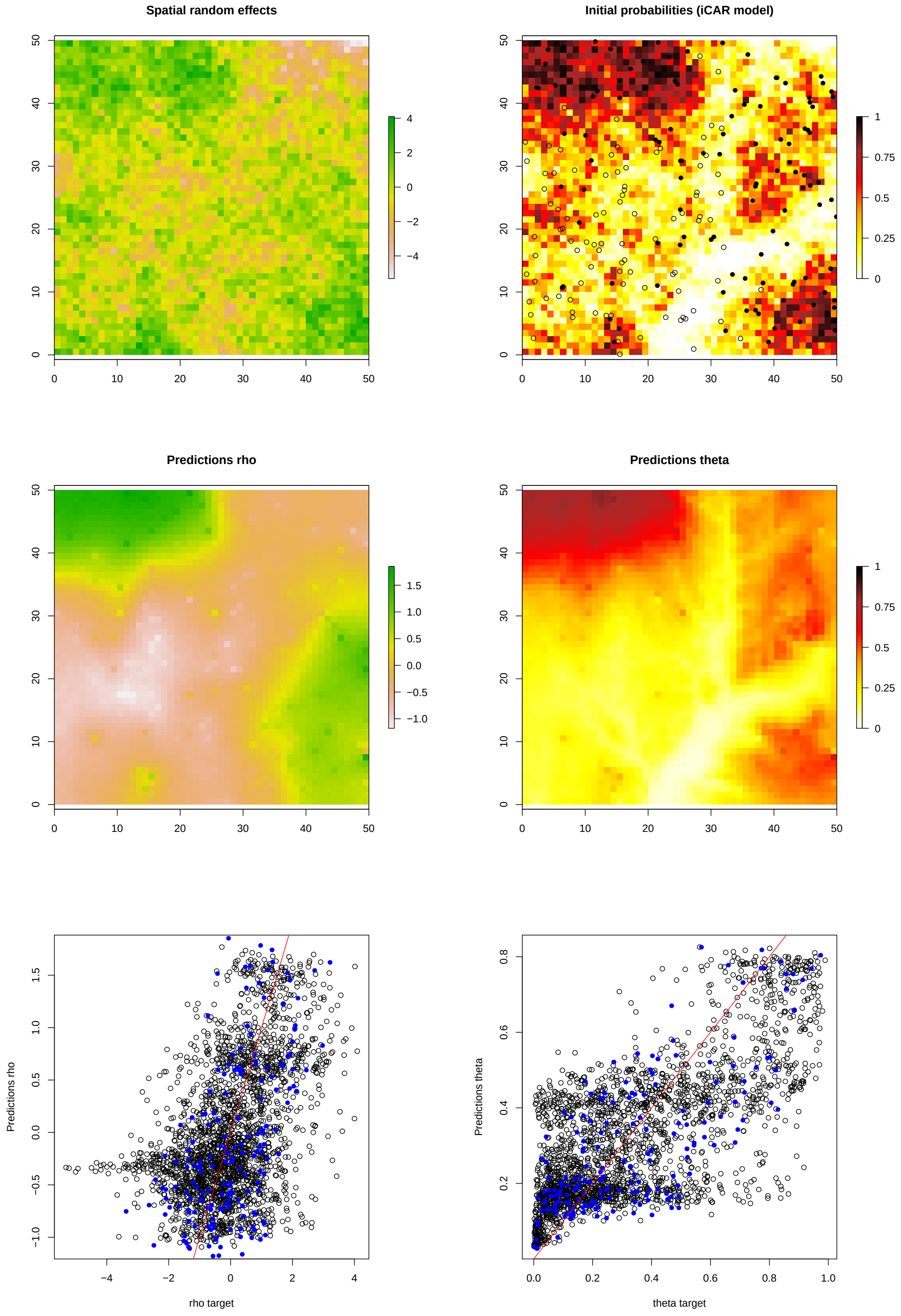


Figure 2.12: **Predictions vs. initial values**

2.3.5 Comparison with OpenBUGS results

```
# BUGS model
modelBUGS1.txt <-
"model {

# likelihood
for (n in 1:nobs) {
  y[n] ~ dbin(theta[n], visits[n])
  logit(theta[n]) <- Xbeta[n] + rho[IdCell[n]]
  Xbeta[n] <- beta[1] + beta[2]*x1[n]
}

# CAR prior distribution for spatial random effects:
rho[1:ncells] ~ car.normal(adj[], weights[], num[], tau)
for (k in 1:sumNumNeigh) {
  weights[k] <- 1 # set equal weights for all neighbors
}

# Other priors
for (i in 1:2) {
  beta[i] ~ dnorm(0,1.0E-6)
}
Vrho <- 1/tau
tau ~ dgamma(1,1)

}"

# Create model.txt file in the working directory
system(paste("echo \"",modelBUGS1.txt,"\" > modelBUGS1.txt",sep=""))

# Data for OpenBUGS
y <- data.suit$Y
visits <- data.suit$visits
IdCell <- data.suit$cells
x1 <- data.suit$alt
num <- n.neighbors
adj <- adj
nobs <- length(y)
ncells <- length(n.neighbors)
sumNumNeigh <- length(adj)
data <- list("y","visits","IdCell","x1","num",
             "adj","nobs","ncells","sumNumNeigh")

# Inits
```

Value	OpenBUGS	hSDM
β_0	-0.99	-0.99
β_{alt}	0.73	0.75
V_ρ	2.72	2.80
Deviance	328.62	327.84
Time (secs)	91	7

Table 2.1: **Comparison between hSDM and OpenBUGS outputs.**

```
inits <- list(list(beta=rep(0,2),rho=rep(0,ncells),tau=1))

# OpenBUGS call
library(R2OpenBUGS)
Start <- Sys.time() # Start the clock
Open <- bugs(data,inits,
             model.file="modelBUGS1.txt",
             parameters=c("beta","Vrho","rho"),
             n.chains=1,
             OpenBUGS.pgm="/usr/local/bin/OpenBUGS",
             n.iter=2000,
             n.burnin=1000,
             n.thin=5,
             DIC=TRUE,
             debug=FALSE,
             clearWD=FALSE)
Time.OpenBUGS <- difftime(Sys.time(),Start,units="sec") # Time difference

# Time difference
ratio.time <- as.numeric(Time.OpenBUGS)/as.numeric(Time.hSDM)
ratio.time # For this example, hSDM is X times faster

#== Outputs
Open$DIC
Open$pD
beta.pred.Open <- apply(Open$sims.list$beta,2,mean)
Vrho.pred.Open <- mean(Open$sims.list$Vrho)
deviance.Open <- mean(Open$sims.list$deviance)
rho.OpenBUGS <- apply(Open$sims.list$rho,2,mean)
plot(rho.pred,rho.OpenBUGS)
abline(a=0,b=1,col="red")
```

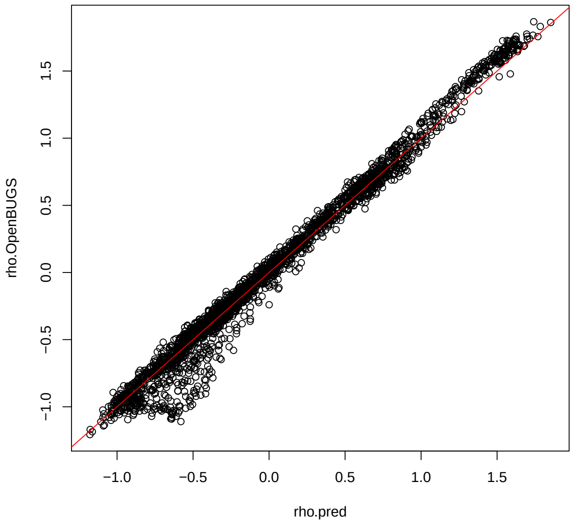


Figure 2.13: **Comparison between hSDM and OpenBUGS for spatial random effect estimates.**

2.3.6 Comparison with GLM results

```
#== glm results for comparison
mod.glm <- glm(cbind(Y,visits)~alt,family="binomial",data=data.suit)
summary(mod.glm)

##
## Call:
## glm(formula = cbind(Y, visits) ~ alt, family = "binomial", data = data.suit)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.371  -0.946  -0.602   0.454   1.811
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.2034     0.0885  -13.60  < 2e-16 ***
## alt           0.4416     0.1024    4.31  1.6e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 157.26  on 199  degrees of freedom
## Residual deviance: 135.34  on 198  degrees of freedom
## AIC: 355
##
## Number of Fisher Scoring iterations: 4
```

```
# Create a raster for predictions
theta.pred.glm <- raster(theta)
# Attribute predicted values to raster cells
theta.pred.glm[] <- predict.glm(mod.glm,newdata=data.pred,type="response")
# Plot the predicted probability of presence
plot(theta.pred.glm,main="GLM for iCAR",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
```

```
# Comparing predictions to initial values
plot(theta[],theta.pred.glm[],
     xlim=c(0,1),ylim=c(0,1),cex.lab=1.4)
abline(a=0,b=1,col="red",lwd=2)
```

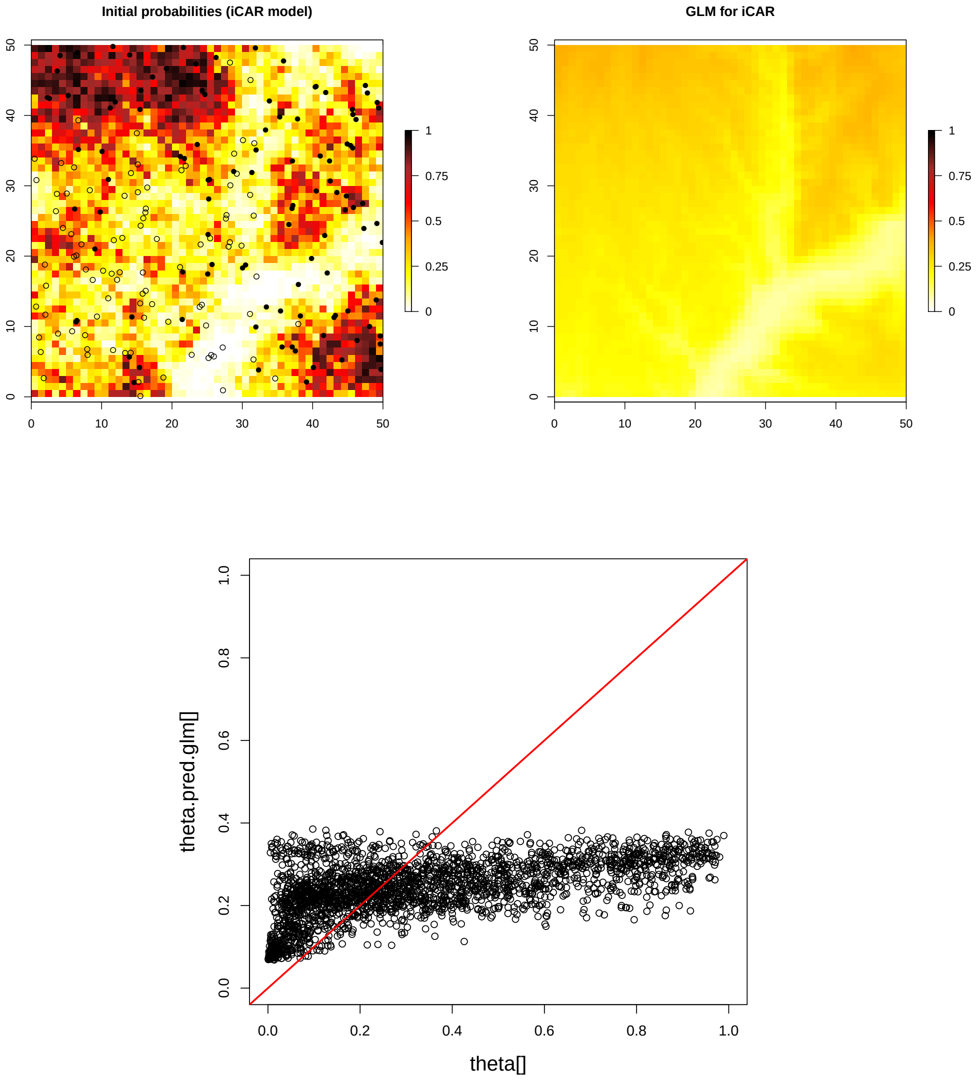


Figure 2.14: **Comparing predicted probability of presence using GLM with initial probabilities for a binomial model with iCAR process.**

2.4 Site-occupancy iCAR model

2.4.1 Mathematical formulation

2.4.2 Data generation

```
# Ecological process (suitability)
logit.theta.site <- X.sites %*% beta.target + rho[cells]
theta.site <- inv.logit(logit.theta.site)
set.seed(seed)
Z <- rbinom(nsite,1,theta.site)

# Observation process (detectability)
nobs <- sum(visits)
set.seed(seed)
Y <- rbinom(nobs,1,delta.obs*Z[sites])

# Data-sets
data.obs <- data.frame(Y,w1,alt=X.sites[sites,2],site=sites)
data.suit <- data.frame(alt=X.sites[,2],cell=cells)
```

2.4.3 Parameter inference using the hSDM.siteocc.iCAR() function

```
Start <- Sys.time() # Start the clock
mod.hSDM.siteocc.iCAR <- hSDM.siteocc.iCAR(# Observations
                    presence=data.obs$Y,
                    observability=~w1+alt,
                    site=data.obs$site, data.observability=data.obs,
                    # Habitat
                    suitability=~alt, data.suitability=data.suit,
                    # Spatial structure
                    spatial.entity=data.suit$cell,
                    n.neighbors=n.neighbors, neighbors=adj,
                    # Predictions
                    suitability.pred=data.pred,
                    spatial.entity.pred=data.pred$cell,
                    # Chains
                    burnin=5000, mcmc=5000, thin=5,
                    # Starting values
                    beta.start=0,
                    gamma.start=0,
                    Vrho.start=1,
                    # Priors
                    mubeta=0, Vbeta=1.0E6,
                    mugamma=0, Vgamma=1.0E6,
                    # priorVrho="1/Gamma",
                    # shape=1, rate=1,
                    priorVrho="Uniform",
                    Vrho.max=10,
                    # Various
                    seed=1234, verbose=1,
                    save.rho=1, save.p=0)
Time.hSDM <- difftime(Sys.time(),Start,units="sec") # Time difference
```

```
summary(mod.hSDM.siteocc.iCAR$mcmc)

##
## Iterations = 5001:9996
## Thinning interval = 5
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                        Mean     SD Naive SE Time-series SE
## beta.(Intercept)     -1.389  0.422  0.01336        0.03063
## beta.alt              1.079  0.479  0.01514        0.05328
## gamma.(Intercept)    -1.108  0.257  0.00812        0.01169
## gamma.w1              1.121  0.259  0.00818        0.01065
## gamma.alt            -0.561  0.224  0.00707        0.00881
## Vrho                  6.843  2.042  0.06457        0.57398
## Deviance            253.104 10.107  0.31960        1.01861
##
## 2. Quantiles for each variable:
##
##                        2.5%     25%     50%     75%   97.5%
## beta.(Intercept)     -2.214  -1.673  -1.388  -1.117  -0.564
## beta.alt              0.165   0.739   1.073   1.393   2.100
## gamma.(Intercept)    -1.621  -1.298  -1.102  -0.928  -0.642
## gamma.w1              0.638   0.949   1.123   1.278   1.641
## gamma.alt            -0.998  -0.713  -0.559  -0.418  -0.107
## Vrho                  3.262   5.164   7.017   8.674   9.820
## Deviance            233.396 246.376 252.940 260.129 274.643
```

```
# Predictions for spatial random effects
rho.pred <- apply(mod.hSDM.siteocc.iCAR$rho.pred,2,mean)
rho.pred.rast <- rasterFromXYZ(cbind(coords,rho.pred))
plot(rho.pred.rast,main="Predictions rho")
# Predictions for probability of presence
theta.pred <- mod.hSDM.siteocc.iCAR$theta.pred
theta.pred.rast <- rasterFromXYZ(cbind(coords,theta.pred))
plot(theta.pred.rast,main="Predictions theta",col=colRP(nb),breaks=brks,
axis.args=arg,zlim=c(0,1))
# Predictions vs. initial spatial random effects
plot(rho[-cells],rho.pred[-cells],xlab="rho target",ylab="Predictions rho")
points(rho[cells],rho.pred[cells],col="blue",pch=16)
abline(a=0,b=1,col="red")
# Predictions vs. initial probabilities
plot(values(theta)[-cells],theta.pred[-cells],xlab="theta target",
ylab="Predictions theta")
points(values(theta)[cells],theta.pred[cells],col="blue",pch=16)
abline(a=0,b=1,col="red")
```

2.4.4 Comparison with OpenBUGS results

```
# BUGS model
modelBUGS.txt <-
"model {

# Suitability process
for (i in 1:nsite) {
  z[i] ~ dbern(theta[i])
  logit(theta[i]) <- Xbeta[i] + rho[IdCellforSite[i]]
  Xbeta[i] <- beta[1] + beta[2]*alt.suit[i]
}

# Observability process
for (n in 1:nobs) {
  y[n] ~ dbern(delta.prim[n])
  delta.prim[n] <- delta[n]*z[IdSiteforObs[n]]
  logit(delta[n]) <- gamma[1] + gamma[2]*w1[n] + gamma[3]*alt.obs[n]
}

# CAR prior distribution for spatial random effects:
rho[1:ncells] ~ car.normal(adj[], weights[], num[], tau)
for (k in 1:sumNumNeigh) {
  weights[k] <- 1 # set equal weights for all neighbors
```

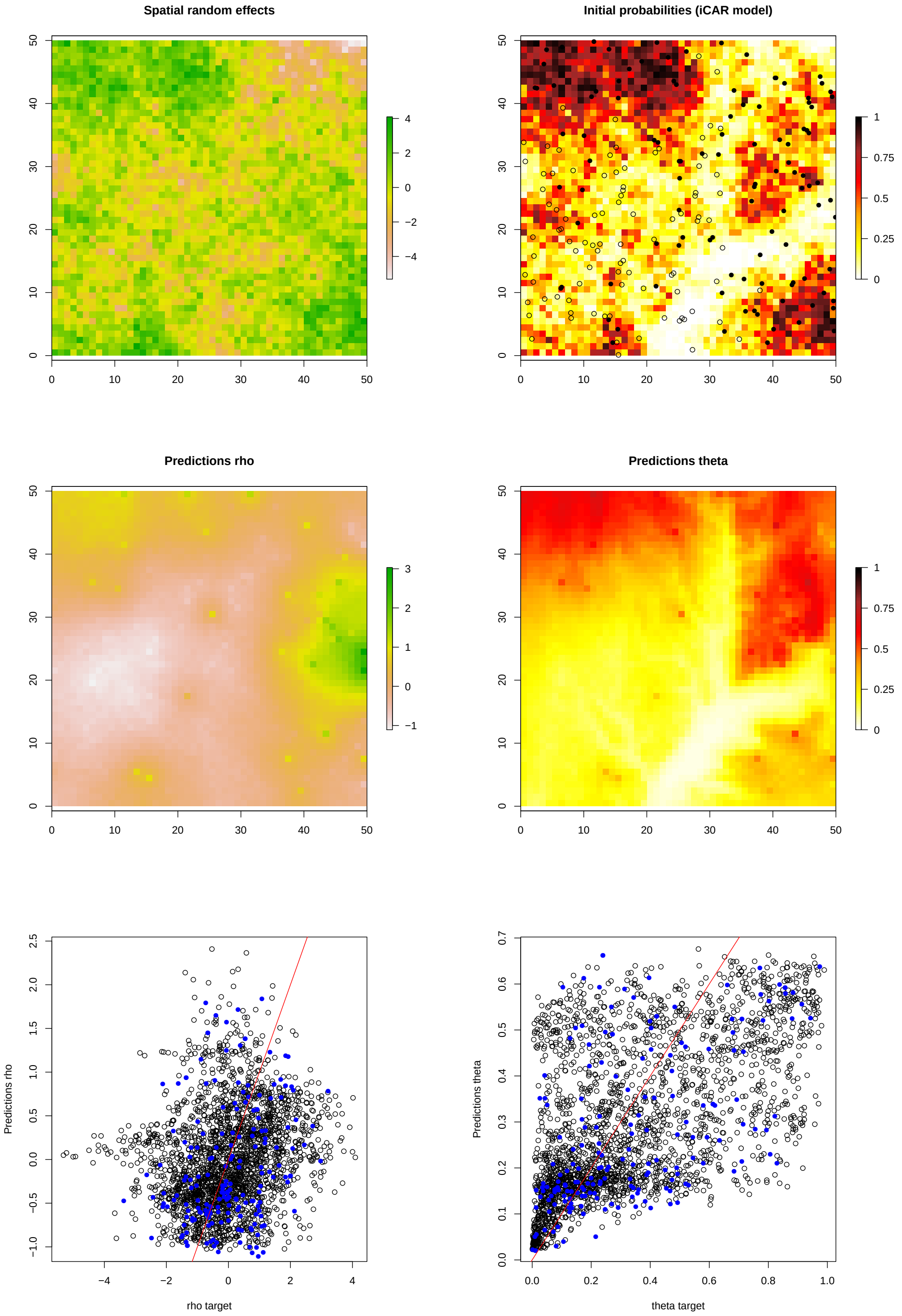


Figure 2.15: **Predictions vs. initial values**

```
}

# Other priors
for (i in 1:2) {
  beta[i] ~ dnorm(0,1.0E-6)
}
for (i in 1:3) {
  gamma[i] ~ dnorm(0,1.0E-6)
}
Vrho ~ dunif(0,10)
tau <- 1/Vrho

}"

# Create model.txt file in the working directory
system(paste("echo \"",modelBUGS.txt,"\" > modelBUGS.txt",sep=""))

# Data for OpenBUGS
y <- data.obs$Y
IdCellforSite <- data.suit$cell
IdSiteforObs <- data.obs$site
alt.suit <- data.suit$alt
w1 <- data.obs$w1
alt.obs <- data.obs$alt
num <- n.neighbors
adj <- adj
nobs <- length(y)
nsite <- length(IdCellforSite)
ncells <- length(n.neighbors)
sumNumNeigh <- length(adj)
data <- list("y","IdCellforSite","IdSiteforObs","alt.suit","w1","alt.obs","num",
"adj","nobs","nsite","ncells","sumNumNeigh")

# Inits
inits <- list(list(beta=rep(0,2),gamma=rep(0,3),rho=rep(0,ncells),Vrho=1))

# OpenBUGS call
library(R2OpenBUGS)
Start <- Sys.time() # Start the clock
Open <- bugs(data,inits,
             model.file="modelBUGS.txt",
             parameters=c("beta","gamma","Vrho","rho"),
             n.chains=1,
             OpenBUGS.pgm="/usr/local/bin/OpenBUGS",
             n.iter=2000,
```

Value	OpenBUGS	hSDM
β_0	-1.49	-1.39
β_{alt}	1.11	1.08
γ_0	-1.06	-1.11
γ_{w1}	1.13	1.12
γ_{alt}	-0.55	-0.56
V_ρ	6.38	6.84
Time (secs)	59	14

Table 2.2: **Comparison between hSDM and OpenBUGS outputs.**

```
             n.burnin=1000,
             n.thin=5,
             DIC=TRUE,
             debug=FALSE,
             clearWD=FALSE)
Time.OpenBUGS <- difftime(Sys.time(),Start,units="sec") # Time difference

# Time difference
ratio.time <- as.numeric(Time.OpenBUGS)/as.numeric(Time.hSDM)
ratio.time # For this example, hSDM is X times faster

#== Outputs
Open$DIC
Open$pD
beta.pred.Open <- apply(Open$sims.list$beta,2,mean)
gamma.pred.Open <- apply(Open$sims.list$gamma,2,mean)
Vrho.pred.Open <- mean(Open$sims.list$Vrho)
deviance.Open <- mean(Open$sims.list$deviance)
rho.OpenBUGS <- apply(Open$sims.list$rho,2,mean)
plot(rho.pred,rho.OpenBUGS)
abline(a=0,b=1,col="red")
```

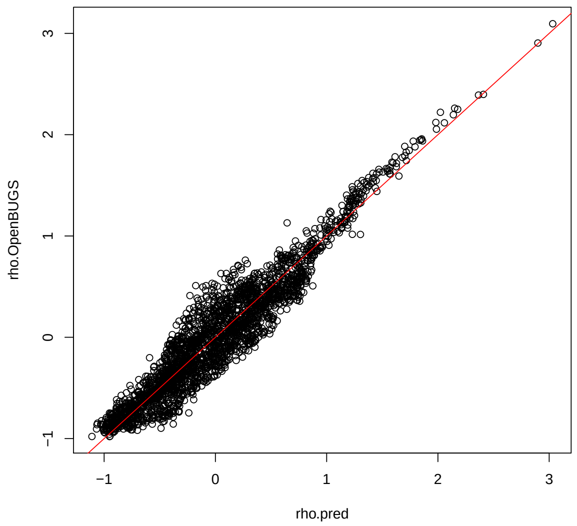


Figure 2.16: **Comparison between hSDM and OpenBUGS for spatial random effect estimates.**

2.4.5 Comparison with GLM results

```
#== glm results for comparison
mod.glm <- glm(y~alt,family="binomial",data=data.obs)
summary(mod.glm)

##
## Call:
## glm(formula = y ~ alt, family = "binomial", data = data.obs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -0.415  -0.415  -0.414  -0.414   2.235
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.41084    0.14924  -16.15   <2e-16 ***
## alt         -0.00103    0.14403   -0.01     0.99
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 338.53  on 594  degrees of freedom
## Residual deviance: 338.53  on 593  degrees of freedom
## AIC: 342.5
##
## Number of Fisher Scoring iterations: 5
```

```
# Create a raster for predictions
theta.pred.glm <- raster(theta)
# Attribute predicted values to raster cells
theta.pred.glm[] <- predict.glm(mod.glm,newdata=data.pred,type="response")
# Plot the predicted probability of presence
plot(theta.pred.glm,main="GLM for siteocc iCAR",col=colRP(nb),breaks=brks,
     axis.args=arg,zlim=c(0,1))
```

```
# Comparing predictions to initial values
plot(theta[],theta.pred.glm[],
     xlim=c(0,1),ylim=c(0,1),cex.lab=1.4)
abline(a=0,b=1,col="red",lwd=2)
```

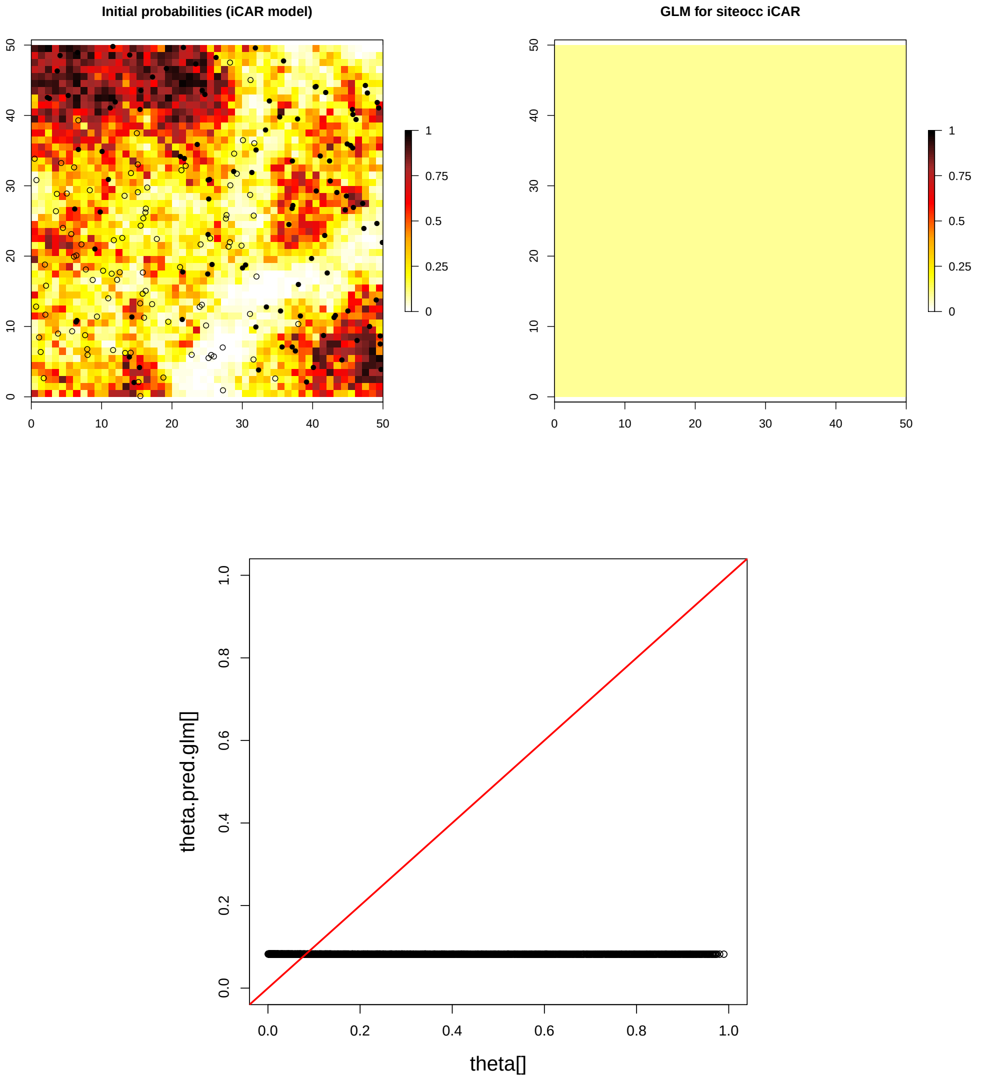


Figure 2.17: **Comparing predicted probability of presence using GLM with initial probabilities for a site-occupancy model with iCAR process.**

CHAPTER 3

Abundance data

CHAPTER 4

Additional examples with real data

4.1 Binomial iCAR model with tens of thousands spatial cells

This exemple illustrates the use of the `hSDM.binomial.iCAR()` function on a large region (tens of thousands grid cells). The data-set includes presence-absence observations for *Protea punctata* Meisn. (Fig. 4.1) in the Cap Floristic Region. The data-set also includes environmental variables for 36909 one minute by one minute grid cells on the whole South Africa's Cap Floristic Region (Fig. 4.2).

```r
# Libraries
require(sp)
require(raster)
library(hSDM)

# Load data
data(cfr.env, package="hSDM")
dim(cfr.env) # 36909 cells
data(punc10, package="hSDM")
dim(punc10) # 2934 observations

# Standardize predictors
for (i in 3:8) {
    m <- cfr.env[,i]-mean(cfr.env[,i], na.rm=T)
    cfr.env[,i] <- m/sd(cfr.env[,i], na.rm=T)
}
```

Figure 4.1: Photography of *Protea punctata* Meisn.

```
# Make both data sets spatial objects
cfr.env <- SpatialPixelsDataFrame(points=cfr.env[c("lon","lat")],
                                  tol=0.175039702866343,
                                  data=cfr.env[,-c(1,2)])
fullgrid(cfr.env) <- TRUE
coordinates(punc10) <- c(2,3)

# Plot the whole data set
spplot(cfr.env,sp.layout=list('sp.points',
                    punc10[punc10$Occurrence==1,],
                    col='black',pch='o'),
       col.regions=rainbow(100,start=0.67,end=0))
```

```
# Get the indices of cells where presences and absences have been observed.
cfr.env.rast <- stack(cfr.env)
pres <- extract(cfr.env.rast, SpatialPoints(punc10[punc10$Occurrence==1,]),
                cellnumbers=TRUE)[,1]
abs <- extract(cfr.env.rast, SpatialPoints(punc10[punc10$Occurrence==0,]),
               cellnumbers=TRUE)[,1]

# Make the data frame used in regressions
ncelltot <- length(cfr.env) # Including NULL cells
d <- data.frame(lon=coordinates(cfr.env)[,1],lat=coordinates(cfr.env)[,2],
                Y=rep(0,ncelltot),
                trials=rep(0,ncelltot),
                cell.orig=1:ncelltot,
                cfr.env@data)
```

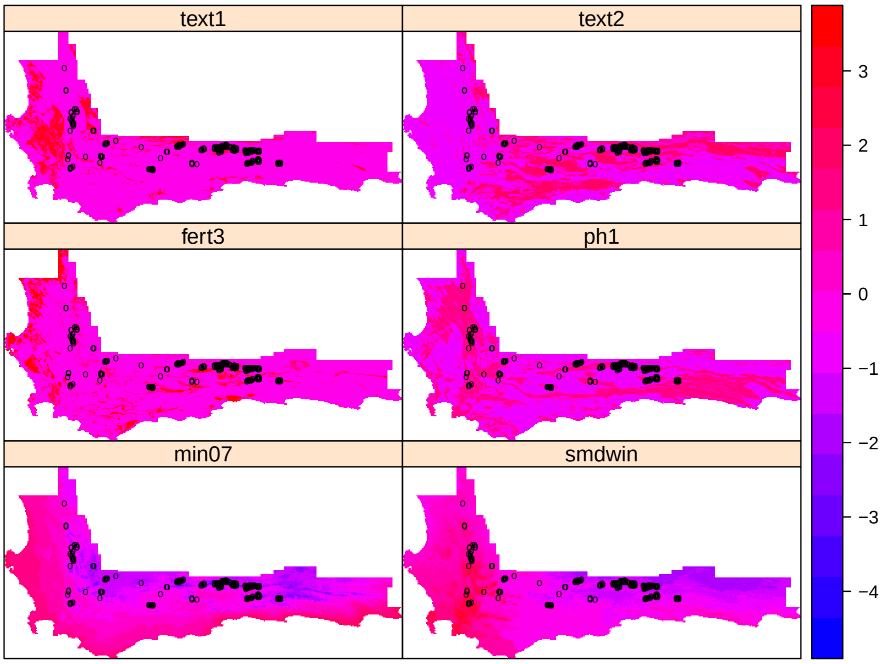


Figure 4.2: **Values of environmental variables in the Cap Floristic Region.** Points of presence of *Protea punctata* are represented by a circle.

```
d$Y[pres] <- 1
d$trials[c(pres,abs)] <- 1

# Remove NAs
to.remove <- which(!complete.cases(d))
d <- d[-to.remove,]
summary(d)

# Make d a spatial object for later use
coordinates(d) <- c(1,2)

# Find cells' neighborhood with function 'adjacent' from the 'raster' package
plot(cfr.env.rast)
sel.cell <- d$cell.orig
neighbors.mat <- adjacent(cfr.env.rast, cells=sel.cell, directions=8,
                          pairs=TRUE, target=sel.cell, sorted=TRUE)
n.neighbors <- as.data.frame(table(as.factor(neighbors.mat[,1])))[,2]
neighbors.orig <- neighbors.mat[,2]

# Sorting cells from 1 to dim(d)[1] (dim(d)[1]=36907)
s.cell <- sort(unique(d$cell.orig))
d$cell <- match(d$cell.orig,s.cell)
s.neighbors <- sort(unique(neighbors.orig))
neighbors <- match(neighbors.orig,s.neighbors)
```

```
# glm, just to compare
mod.glm <- glm(cbind(Y,trials-Y)~min07+smdwin,data=d, family="binomial")
summary(mod.glm)

# hSDM
mod.hSDM.binomial.iCAR <- hSDM.binomial.iCAR(presences=d$Y[d$trials>0],
                                             trials=d$trials[d$trials>0],
                                             suitability=~min07+smdwin,
                                             spatial.entity=d$cell[d$trials>0],
                                             data=d[d$trials>0,],
                                             n.neighbors=n.neighbors,
                                             neighbors=neighbors,
                                             suitability.pred=d,
                                             spatial.entity.pred=d$cell,
                                             burnin=1000,
                                             mcmc=1000, thin=1,
                                             beta.start=c(0,0,0),
                                             Vrho.start=10,
                                             priorVrho="1/Gamma",
                                             mubeta=0, Vbeta=1.0E6,
                                             shape=2, rate=1,
                                             Vrho.max=10,
                                             seed=1234, verbose=1, save.rho=0)
```

```
# Outputs
summary(mod.hSDM.binomial.iCAR$mcmc)

##
## Iterations = 1001:2000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                      Mean     SD Naive SE Time-series SE
## beta.(Intercept)  -6.792  0.272  0.00860         0.0573
## beta.min07        -2.734  0.163  0.00515         0.0347
## beta.smdwin        0.758  0.140  0.00444         0.0267
## Vrho               6.778  1.061  0.03355         0.6747
## Deviance         489.104 20.475  0.64748         4.0209
##
## 2. Quantiles for each variable:
##
##                     2.5%     25%     50%     75%  97.5%
## beta.(Intercept)  -7.438  -6.969  -6.752  -6.606  -6.36
## beta.min07        -3.093  -2.840  -2.718  -2.634  -2.42
## beta.smdwin        0.466   0.674   0.758   0.844   1.01
## Vrho               5.365   5.934   6.548   7.640   8.84
## Deviance         451.825 474.131 488.086 502.830 533.92

# Put output together
out <- data.frame(d,pred=mod.hSDM.binomial.iCAR$theta.pred,
                  sp.ef=mod.hSDM.binomial.iCAR$rho.pred)

# Plot results
coordinates(out) <- coordinates(d)
out <- SpatialPixelsDataFrame(out,tol=0.175039702866343,data=data.frame(out))
fullgrid(out) <- TRUE
p1 <- spplot(out['pred'],col.regions=rainbow(100,start=0.67,end=0),
             sp.layout=list('sp.points',punc10[punc10$Occurrence==1,],
                 col='black',pch='o'))
```

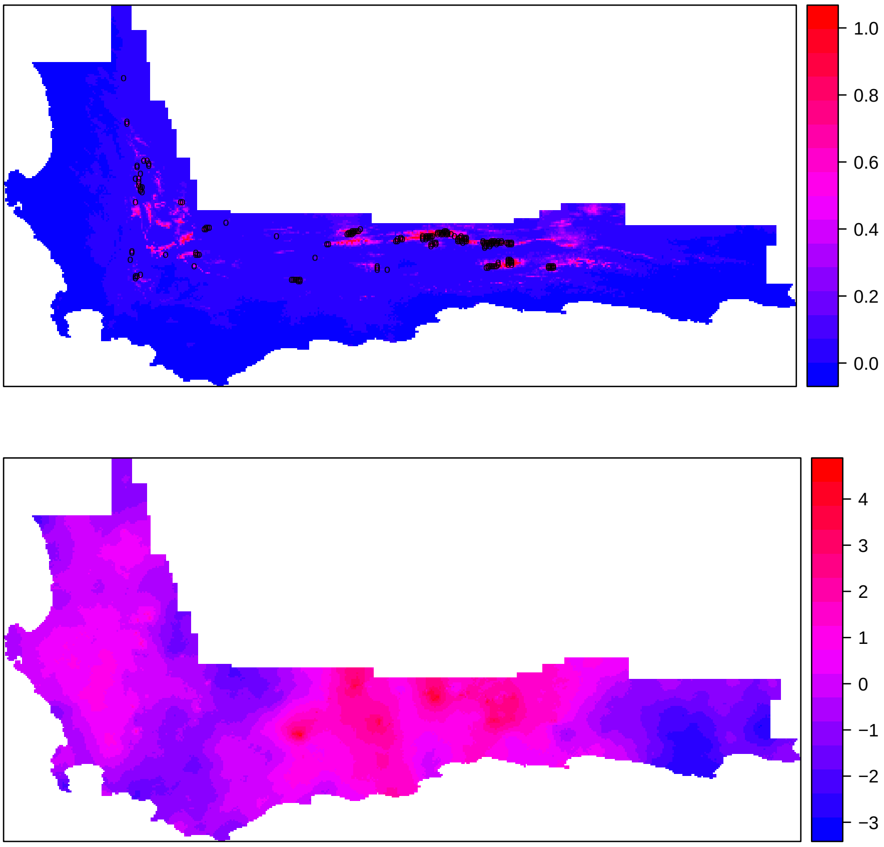


Figure 4.3: **Predicted probability of presence (top) and estimated spatial random effects (bottom).** Points of presence of *Protea punctata* are represented by a circle.

```r
p2 <- spplot(out['sp.ef'],col.regions=rainbow(100,start=0.67,end=0))
print(p1,position=c(0,0.5,1,1),more=T)
print(p2,position=c(0,0,1,0.5))
```

Using function `hSDM.binomial.iCAR()`, we were able to estimate the spatial random effect of 36907 cells (Fig. 4.3) and we demonstrated that the use of this function is not limited (through memory problem or a much too long computation time) by the number of spatial grid cells. Nevertheless, in this particular example, it is very difficult to reach convergence for the variance of the spatial random effects (see MCMC outputs above). This is likely due to the low information content of binary maps and the relatively low number of observations (2934). As previously underlined by Dormann *et al.* (2007), we argue that binomial intrinsic CAR models require further study and caution in their use. The **hSDM** R package offers tools to help ecologist explore the behavior and performance of such models.

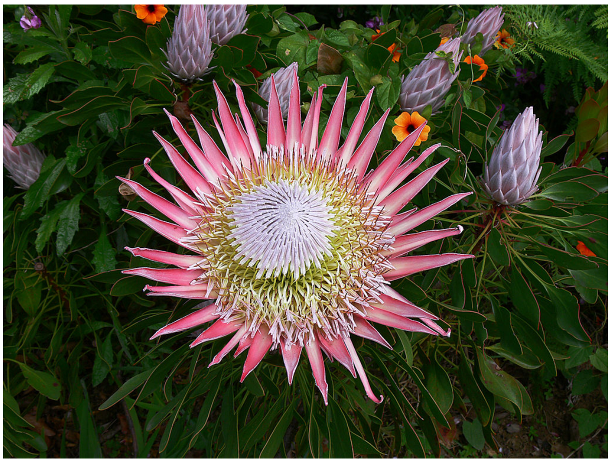


Figure 4.4: Photography of *Protea cynaroides* (L.) L..

4.2 Binomial iCAR model with data from Latimer *et al.* (2006)

In the Appendix B of their scientific article, Latimer *et al.* (2006) provide some code to fit what they called “Model 2”, a Binomial iCAR model using presence/absence data for species *Protea cynaroides* (L.) L., a common *Protea* and the national flower of South Africa (Fig. 4.4).

For the purpose of their example, Latimer *et al.* (2006) provide data for a small region including 476 one minute by one minute grid cells. This region is is a small corner of South Africa’s Cape Floristic Region, and includes very high plant species diversity and a World Biosphere Reserve. Contrary to the previous example, the data-set includes several visits at the same site.

```
# Library
library(hSDM)

# Load data
data(datacells.Latimer2006,package="hSDM")
datacells.Latimer2006$cell <- c(1:dim(datacells.Latimer2006)[1])
data(neighbors.Latimer2006,package="hSDM")

# Format data
p <- datacells.Latimer2006$y[datacells.Latimer2006$n>0]
t <- datacells.Latimer2006$n[datacells.Latimer2006$n>0]
s <- datacells.Latimer2006$cell[datacells.Latimer2006$n>0]
data.obs <- datacells.Latimer2006[datacells.Latimer2006$n>0,]
```

```
# Model
Start <- Sys.time() # Start the clock
mod.hSDM.Lat2006.iCAR <- hSDM.binomial.iCAR(presences=p,
                         trials=t,
                         suitability=~rough+julmint+pptcv+smdsum+evi+ph1,
                         spatial.entity=s,
                         data=data.obs,
                         n.neighbors=datacells.Latimer2006$num,
                         neighbors=neighbors.Latimer2006,
                         suitability.pred=datacells.Latimer2006,
                         spatial.entity.pred=datacells.Latimer2006$cell,
                         burnin=5000,
                         mcmc=5000, thin=5,
                         beta.start=0,
                         Vrho.start=10,
                         priorVrho="1/Gamma",
                         mubeta=0, Vbeta=1.0E6,
                         shape=0.001, rate=0.001,
                         Vrho.max=1000,
                         seed=1234, verbose=1,
                         save.rho=0,save.p=0)
Time.hSDM <- difftime(Sys.time(),Start,units="sec") # Time difference

# Some outputs
summary(mod.hSDM.Lat2006.iCAR$rho.pred)
summary(mod.hSDM.Lat2006.iCAR$theta.latent)
summary(mod.hSDM.Lat2006.iCAR$theta.pred)
```

```
# Parameter estimates
summary(mod.hSDM.Lat2006.iCAR$mcmc)

##
## Iterations = 5001:9996
## Thinning interval = 5
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                       Mean     SD Naive SE Time-series SE
## beta.(Intercept)  -1.8742  0.187  0.00590         0.0231
## beta.rough         0.0465  0.193  0.00610         0.0314
## beta.julmint      -0.6892  0.191  0.00603         0.0360
## beta.pptcv        -0.5055  0.313  0.00990         0.0750
## beta.smdsum       -0.0582  0.258  0.00816         0.0638
## beta.evi          -0.1212  0.225  0.00712         0.0327
## beta.ph1           1.1842  0.367  0.01159         0.0749
## Vrho              10.0134  1.679  0.05308         0.1426
## Deviance         741.7470 23.254  0.73535         1.3365
##
## 2. Quantiles for each variable:
##
##                      2.5%     25%      50%      75%   97.5%
## beta.(Intercept)  -2.286  -1.985  -1.8575  -1.7508  -1.532
## beta.rough        -0.306  -0.099   0.0495   0.1885   0.404
## beta.julmint      -1.011  -0.812  -0.7064  -0.5993  -0.230
## beta.pptcv        -1.133  -0.702  -0.5067  -0.2927   0.119
## beta.smdsum       -0.574  -0.235  -0.0594   0.1290   0.428
## beta.evi          -0.582  -0.263  -0.1210   0.0171   0.381
## beta.ph1           0.537   0.924   1.1754   1.4102   1.964
## Vrho               7.137   8.797   9.9211  11.0806  13.545
## Deviance         695.120 725.745 742.3363 758.1263 786.873
```

Contrary to the previous example, and due to the higher information content associated to the fact that each site is visited several times, it was easier to reach convergence for the variance of the spatial random effects in this example.

```
# BUGS model
modelBUGS2.txt <-
"model {

# Likelihood
for (i in 1:N_nonzeroy) {
  y[ind[i]] ~ dbin(p[ind[i]], n[ind[i]])
}

for(i in 1:N_LOC){
  logit(p[i]) <- rho[i]+xbeta[i]+mu
  xbeta[i]<-beta[1]*rough[i] + beta[2]*julmint[i] + beta[3]*pptcv[i] +
            beta[4]*smdsum[i] + beta[5]*evi[i] + beta[6]*ph1[i]
}

# CAR prior distribution for spatial random effects:
rho[1:N_LOC] ~ car.normal(adj[], weights[], num[], tau)
for(k in 1:sumNumNeigh) {
  weights[k] <- 1 # set equal weights for all neighbors
}
```

```
# Other priors
mu ~ dnorm(0,0.1)
for (i in 1:6) {
  beta[i] ~ dnorm(0, 0.2)
}
vrho <- 1/tau
tau ~ dgamma(0.001,0.001)

}"

# Create model.txt file in the working directory
system(paste("echo \"",modelBUGS2.txt,"\" > modelBUGS2.txt",sep=""))

# Data for OpenBUGS
y <- datacells.Latimer2006$y
n <- datacells.Latimer2006$n
rough <- datacells.Latimer2006$rough
julmint <- datacells.Latimer2006$julmint
pptcv <- datacells.Latimer2006$pptcv
smdsum <- datacells.Latimer2006$smdsum
evi <- datacells.Latimer2006$evi
ph1 <- datacells.Latimer2006$ph1
num <- datacells.Latimer2006$num
adj <- neighbors.Latimer2006
ind <- which(datacells.Latimer2006$n!=0)
N_LOC <- length(y)
N_nonzeroy <- length(ind)
sumNumNeigh <- length(adj)

data <- list("y","n","rough","julmint","pptcv","smdsum",
             "evi","ph1","num",
             "adj","ind","N_LOC","N_nonzeroy","sumNumNeigh")

# Inits
inits <- list(list(mu=1,beta=rep(1.5,6),rho=rep(0,N_LOC),tau=1))

# OpenBUGS call
library(R2OpenBUGS)
Start <- Sys.time() # Start the clock
Open <- bugs(data,inits,
            model.file="modelBUGS2.txt",
            parameters=c("mu","beta","vrho"),
            n.chains=1,
            OpenBUGS.pgm="/usr/bin/OpenBUGS",
            n.iter=2000,
            n.burnin=1000,
            n.thin=5,
            DIC=TRUE,
            debug=FALSE,
            clearWD=FALSE)
Time.OpenBUGS <- difftime(Sys.time(),Start,units="sec") # Time difference

# Time difference
ratio.time <- as.numeric(Time.OpenBUGS)/as.numeric(Time.hSDM)
```

```
# Parameter estimates with OpenBUGS
print(Open$summary[,c(1,2)])

##               mean      sd
## mu        -1.85206  0.1598
## beta[1]    0.03596  0.1455
## beta[2]   -0.74706  0.2354
## beta[3]   -0.49305  0.2674
## beta[4]   -0.12948  0.3064
## beta[5]   -0.15000  0.1899
## beta[6]    1.15289  0.2608
## vrho       9.59599  1.5356
## deviance 740.95200 21.0887
```

For this example, hSDM and OpenBUGS gave similar estimates for model parameters. For the same number of iterations (10000), and for a relatively low number of grid cells (476), hSDM was more than twice as fast as OpenBUGS.

4.3 ZIB model with data from Latimer *et al.* (2006)

Because sites have been visited several times, the same data-set can be used to fit a ZIB model accounting for imperfect detection. If the observation conditions were different from one visit to another, we would have to use the `hSDM.siteocc()` function which uses a mixture model combining two Bernoulli processes. But in this case, the observation conditions are not specified and can be supposed to be the same so that we can use the `hSDM.ZIB()` function of the **hSDM** package. The `hSDM.ZIB()` function uses a mixture model combining a Binomial process for observability and a Bernoulli process for suitability.

```
# Model
mod.hSDM.Lat2006.ZIB <- hSDM.ZIB(presences=p,
                                 trials=t,
                                 suitability=~rough+julmint+pptcv+smdsum+evi+ph1,
                                 observability=~1,
                                 data=data.obs,
                                 suitability.pred=datacells.Latimer2006,
                                 burnin=1000,
                                 mcmc=1000, thin=1,
                                 beta.start=0,
                                 gamma.start=0,
                                 mubeta=0, Vbeta=1.0E6,
                                 mugamma=0, Vgamma=1.0E6,
                                 seed=1234, verbose=1,
                                 save.p=0)

# Some outputs
summary(mod.hSDM.Lat2006.ZIB$prob.p.pred)
summary(mod.hSDM.Lat2006.ZIB$prob.p.latent)
summary(mod.hSDM.Lat2006.ZIB$prob.q.latent)
```

```
# Parameter estimates
summary(mod.hSDM.Lat2006.ZIB$mcmc)

##
## Iterations = 1001:2000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                         Mean     SD Naive SE Time-series SE
## beta.(Intercept)   3.38e-01 0.2259  0.00714        0.02411
## beta.rough        -7.02e-03 0.2653  0.00839        0.02380
## beta.julmint       8.85e-01 0.3162  0.01000        0.04036
## beta.pptcv        -4.26e-01 0.3363  0.01064        0.04461
## beta.smdsum        6.61e-01 0.3225  0.01020        0.03824
## beta.evi          -9.29e-01 0.3089  0.00977        0.02772
## beta.ph1           1.61e+00 0.3486  0.01102        0.03535
## gamma.(Intercept)  1.26e-01 0.0375  0.00118        0.00364
## Deviance           1.84e+03 4.1585  0.13150        0.40860
##
## 2. Quantiles for each variable:
##
##                         2.5%      25%       50%      75%    97.5%
## beta.(Intercept)    -0.0671    0.179  3.16e-01    0.474    0.767
## beta.rough          -0.5603   -0.180 -2.72e-03    0.163    0.445
## beta.julmint         0.2268    0.693  9.02e-01    1.110    1.488
## beta.pptcv          -1.1054   -0.613 -3.97e-01   -0.215    0.210
## beta.smdsum          0.0588    0.442  6.43e-01    0.875    1.307
## beta.evi            -1.6330   -1.140 -8.93e-01   -0.691   -0.417
## beta.ph1             0.9186    1.383  1.60e+00    1.839    2.321
## gamma.(Intercept)    0.0476    0.099  1.23e-01    0.150    0.199
## Deviance          1831.8041 1834.727  1.84e+03 1839.954 1847.497
```

```
# Detection probability
gamma.hat <- mean(mod.hSDM.Lat2006.ZIB$mcmc[,"gamma.(Intercept)"])
delta.est <- inv.logit(gamma.hat)
delta.est

## [1] 0.5314
```

Using this type of model, we can estimate the detection probability of the species (`delta.est`= 0.53).

4.4 ZIB iCAR model with data from Latimer *et al.* (2006)

```
# Model
mod.hSDM.Lat2006.ZIB.iCAR <- hSDM.ZIB.iCAR(presences=p,
                         trials=t,
                         suitability=~rough+julmint+pptcv+smdsum+evi+ph1,
                         observability=~1,
                         spatial.entity=s,
                         data=data.obs,
                         n.neighbors=datacells.Latimer2006$num,
                         neighbors=neighbors.Latimer2006,
                         suitability.pred=datacells.Latimer2006,
                         spatial.entity.pred=datacells.Latimer2006$cell,
                         burnin=5000,
                         mcmc=5000, thin=5,
                         beta.start=0,
                         gamma.start=0,
                         Vrho.start=10,
                         priorVrho="Uniform",
                         mubeta=0, Vbeta=1.0E6,
                         mugamma=0, Vgamma=1.0E6,
                         shape=2, rate=1,
                         Vrho.max=10,
                         seed=1234, verbose=1,
                         save.rho=0,save.p=0)

# Some outputs
summary(mod.hSDM.Lat2006.ZIB.iCAR$prob.p.pred)
summary(mod.hSDM.Lat2006.ZIB.iCAR$prob.p.latent)
summary(mod.hSDM.Lat2006.ZIB.iCAR$prob.q.latent)
```

```
# Parameter estimates
summary(mod.hSDM.Lat2006.ZIB.iCAR$mcmc)

##
## Iterations = 5001:9996
## Thinning interval = 5
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                        Mean     SD Naive SE Time-series SE
## beta.(Intercept)     0.714  0.267  0.00845        0.01477
## beta.rough           0.564  0.398  0.01259        0.04621
## beta.julmint        -1.186  0.593  0.01876        0.06945
## beta.pptcv          -0.796  0.499  0.01579        0.06185
## beta.smdsum         -0.478  0.621  0.01964        0.08805
## beta.evi            -0.510  0.389  0.01229        0.02395
## beta.ph1             1.384  0.421  0.01331        0.03544
## gamma.(Intercept)    0.130  0.037  0.00117        0.00133
## Vrho                 9.124  0.770  0.02434        0.06126
## Deviance          1729.700 10.985  0.34738        0.84539
##
## 2. Quantiles for each variable:
##
##                        2.5%      25%      50%       75%    97.5%
## beta.(Intercept)     0.2125    0.542    0.708    0.9055    1.229
## beta.rough          -0.1535    0.296    0.556    0.8198    1.417
## beta.julmint        -2.3756   -1.590   -1.158   -0.7720   -0.100
## beta.pptcv          -1.6889   -1.123   -0.848   -0.4659    0.261
## beta.smdsum         -1.5652   -0.898   -0.535   -0.0891    0.845
## beta.evi            -1.2773   -0.764   -0.496   -0.2309    0.180
## beta.ph1             0.6118    1.090    1.357    1.6705    2.171
## gamma.(Intercept)    0.0607    0.104    0.131    0.1559    0.199
## Vrho                 7.1122    8.730    9.344    9.7356    9.990
## Deviance          1710.1820 1721.665 1729.205 1737.0470 1752.443
```

```
# Detection probability
gamma.hat <- mean(mod.hSDM.Lat2006.ZIB.iCAR$mcmc[,"gamma.(Intercept)"])
delta.est <- inv.logit(gamma.hat)
delta.est

## [1] 0.5324
```

4.5 Abundance models with data from Kéry & Andrew Royle (2010)

4.5.1 Presentation of the data

The data-set from Kéry & Andrew Royle (2010) includes repeated count data for the Willow tit (*Poecile montanus*, a pesserine bird, see Fig. 4.5) in Switzerland on the period 1999-2003. Data come from the Swiss national breeding bird survey MHB (Monitoring Haüfige Brutvögel). MHB is based on 264 1-km^2 sampling units (quadrats) laid out as a grid (Fig. 4.6). Since 1999, every quadrat has been surveyed two to three times during most breeding seasons (15 April to 15 July). The Willow tit is a widespread but moderately rare bird species. It has a weak song and elusive behaviour and can be rather difficult to detect.

This data-set is available in the **hSDM** R package. It can be loaded with the `data` command and formated to be used with hSDM functions.

```
# Load libraries
library(hSDM)
library(sp)
library(raster)

# Load Kéry et al. 2010 data
data(data.Kery2010,package="hSDM")
head(data.Kery2010)

# Normalized variables
elev.mean <- mean(data.Kery2010$elevation)
elev.sd <- sd(data.Kery2010$elevation)
juldate.mean <- mean(c(data.Kery2010$juldate1,
                     data.Kery2010$juldate2,
                     data.Kery2010$juldate3),na.rm=TRUE)
juldate.sd <- sd(c(data.Kery2010$juldate1,
                 data.Kery2010$juldate2,
                 data.Kery2010$juldate3),na.rm=TRUE)
data.Kery2010$elevation <- (data.Kery2010$elevation-elev.mean)/elev.sd
```

Figure 4.5: **Willow tit (*Poecile montanus*).**

```r
data.Kery2010$juldate1 <- (data.Kery2010$juldate1-juldate.mean)/juldate.sd
data.Kery2010$juldate2 <- (data.Kery2010$juldate2-juldate.mean)/juldate.sd
data.Kery2010$juldate3 <- (data.Kery2010$juldate3-juldate.mean)/juldate.sd

# Landscape and observation sites
sites.sp <- SpatialPointsDataFrame(coords=data.Kery2010[c("coordx","coordy")],
                                   data=data.Kery2010[,-c(1,2)])
xmin <- min(data.Kery2010$coordx)
xmax <- max(data.Kery2010$coordx)
ymin <- min(data.Kery2010$coordy)
ymax <- max(data.Kery2010$coordy)
ext <- extent(c(xmin,xmax,ymin,ymax))
ncol <- round((xmax-xmin)/10)
nrow <- round((ymax-ymin)/10)
landscape <- raster(ncols=ncol,nrows=nrow,ext)
values(landscape) <- runif(ncell(landscape),0,1)
plot(landscape,legend=FALSE)
plot(sites.sp,add=TRUE,col="black")
# Neighborhood
# Rasters must be projected to correctly compute the neighborhood
crs(landscape) <- '+proj=utm +zone=1'
# Cell for each site
cells <- extract(landscape,sites.sp,cell=TRUE)[,1]
# Neighborhood matrix
ncells <- ncell(landscape)
neighbors.mat <- adjacent(landscape, cells=c(1:ncells), directions=8,
                          pairs=TRUE, sorted=TRUE)
# Number of neighbors by cell
n.neighbors <- as.data.frame(table(as.factor(neighbors.mat[,1])))[,2]
```

```r
# Adjacent cells
adj <- neighbors.mat[,2]

# Arranging data
# data.obs
nsite <- length(data.Kery2010$coordx)
count <- c(data.Kery2010$count1,data.Kery2010$count2,data.Kery2010$count3)
juldate <- c(data.Kery2010$juldate1,data.Kery2010$juldate2,
             data.Kery2010$juldate3)
site <- rep(1:nsite,3)
data.obs <- data.frame(count,juldate,site)
data.obs <- data.obs[!is.na(data.obs$juldate),]
# data.suit
data.suit <- data.Kery2010[c("coordx","coordy","elevation","forest")]
data.suit$cells <- cells
data.suit <- data.suit[-139,] # Removing site 139 with no juldate
```

4.5.2 Simple Poisson model

```r
# hSDM.poisson
data.pois <- data.obs
data.pois$elevation <- data.suit$elevation[as.numeric(as.factor(data.obs$site))]
mod.Kery2010.pois <- hSDM.poisson(counts=data.pois$count,
                                  suitability=~elevation+I(elevation^2),
                                  data=data.pois,beta.start=0)
```

```r
# Outputs
summary(mod.Kery2010.pois$mcmc)

##
## Iterations = 5001:14991
## Thinning interval = 10
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                          Mean     SD Naive SE Time-series SE
## beta.(Intercept)       0.0281 0.0643  0.00203        0.00373
## beta.elevation         3.0813 0.1535  0.00485        0.01716
## beta.I(elevation^2)   -1.8000 0.1023  0.00324        0.01067
```

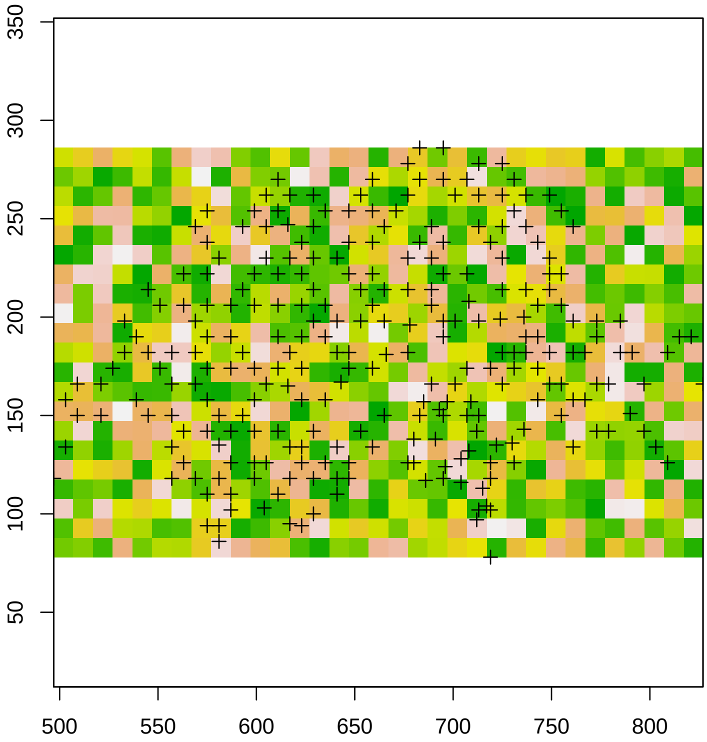


Figure 4.6: **Location of the 264 1-km^2 quadrats of the Swiss national breeding bird survey.** Points are located on a grid of 10-km^2 cells. The grid is covering the geographical extent of the observation points.

```
## Deviance            2157.8806 2.4051  0.07606        0.11616
##
## 2. Quantiles for each variable:
##
##                          2.5%       25%       50%      75%    97.5%
## beta.(Intercept)      -0.0926   -0.0144    0.0276    0.071    0.156
## beta.elevation         2.8072    2.9783    3.0775    3.178    3.382
## beta.I(elevation^2)   -2.0024   -1.8626   -1.7991   -1.731   -1.611
## Deviance            2155.1993 2156.1413 2157.2780 2158.971 2164.380
```

```
# Predictions
npred <- 100
nsamp <- dim(mod.Kery2010.pois$mcmc)[1]
# Abundance-elevation
elev.seq <- seq(500,3000,length.out=npred)
elev.seq.n <- (elev.seq-elev.mean)/elev.sd
beta <- as.matrix(mod.Kery2010.pois$mcmc[,1:3])
tbeta <- t(beta)
X <- matrix(c(rep(1,npred),elev.seq.n,elev.seq.n^2),ncol=3)
N <- matrix(NA,nrow=nsamp,ncol=npred)
for (i in 1:npred) {
    N[,i] <- exp(X[i,] %*% tbeta)
}
N.est.pois <- apply(N,2,mean)
N.q1.pois <- apply(N,2,quantile,0.025)
N.q2.pois <- apply(N,2,quantile,0.975)
```

4.5.3 N-mixture model with imperfect detection

```
# hSDM.Nmixture
mod.Kery2010.Nmix <- hSDM.Nmixture(# Observations
                     counts=data.obs$count,
                     observability=~juldate+I(juldate^2),
                     site=data.obs$site,
                     data.observability=data.obs,
                     # Habitat
                     suitability=~elevation+I(elevation^2),
                     data.suitability=data.suit,
                     # Predictions
                     suitability.pred=NULL,
                     # Chains
                     burnin=10000, mcmc=5000, thin=5,
                     # Starting values
                     beta.start=0,
                     gamma.start=0,
                     # Priors
                     mubeta=0, Vbeta=1.0E6,
                     mugamma=0, Vgamma=1.0E6,
                     # Various
                     seed=1234, verbose=1,
                     save.p=0, save.N=0)
```

```
# Outputs
summary(mod.Kery2010.Nmix$mcmc)

##
## Iterations = 10001:14996
## Thinning interval = 5
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                          Mean      SD Naive SE Time-series SE
## beta.(Intercept)        0.673  0.0866  0.00274        0.00734
## beta.elevation          2.788  0.1961  0.00620        0.02310
## beta.I(elevation^2)    -1.790  0.1444  0.00456        0.01834
## gamma.(Intercept)       0.254  0.1035  0.00327        0.00833
## gamma.juldate          -0.225  0.0849  0.00269        0.00490
## gamma.I(juldate^2)      0.266  0.0820  0.00259        0.00567
## Deviance             1887.583 27.7300  0.87690        2.19556
##
## 2. Quantiles for each variable:
##
##                          2.5%      25%      50%      75%     97.5%
## beta.(Intercept)        0.5018    0.614    0.674    0.730    0.8379
## beta.elevation          2.4165    2.644    2.793    2.923    3.1701
## beta.I(elevation^2)    -2.0726   -1.888   -1.786   -1.689   -1.5142
## gamma.(Intercept)       0.0538    0.177    0.260    0.328    0.4439
## gamma.juldate          -0.3892   -0.282   -0.223   -0.168   -0.0571
## gamma.I(juldate^2)      0.0952    0.209    0.267    0.321    0.4189
## Deviance             1837.0699 1868.213 1885.680 1906.769 1942.4545
```

```
# Predictions
nsamp <- dim(mod.Kery2010.Nmix$mcmc)[1]
# Abundance-elevation
beta <- as.matrix(mod.Kery2010.Nmix$mcmc[,1:3])
tbeta <- t(beta)
N <- matrix(NA,nrow=nsamp,ncol=npred)
for (i in 1:npred) {
    N[,i] <- exp(X[i,] %*% tbeta)
}
N.est.Nmix <- apply(N,2,mean)
N.q1.Nmix <- apply(N,2,quantile,0.025)
N.q2.Nmix <- apply(N,2,quantile,0.975)
# Detection-Julian date
juldate.seq <- seq(100,200,length.out=npred)
juldate.seq.n <- (juldate.seq-juldate.mean)/juldate.sd
gamma <- as.matrix(mod.Kery2010.Nmix$mcmc[,4:6])
tgamma <- t(gamma)
W <- matrix(c(rep(1,npred),juldate.seq.n,juldate.seq.n^2),ncol=3)
delta <- matrix(NA,nrow=nsamp,ncol=npred)
for (i in 1:npred) {
    delta[,i] <- inv.logit(X[i,] %*% tgamma)
}
delta.est.Nmix <- apply(delta,2,mean)
delta.q1.Nmix <- apply(delta,2,quantile,0.025)
delta.q2.Nmix <- apply(delta,2,quantile,0.975)
```

4.5.4 Nmixture model with iCAR process

```
# hSDM.Nmixture.iCAR
mod.Kery2010.Nmix.iCAR <- hSDM.Nmixture.iCAR(# Observations
                            counts=data.obs$count,
                            observability=~juldate+I(juldate^2),
                            site=data.obs$site,
                            data.observability=data.obs,
                            # Habitat
                            suitability=~elevation+I(elevation^2),
                            data.suitability=data.suit,
                            # Spatial structure
                            spatial.entity=data.suit$cells,
                            n.neighbors=n.neighbors, neighbors=adj,
                            # Chains
                            burnin=20000, mcmc=10000, thin=10,
                            # Starting values
                            beta.start=0,
                            gamma.start=0,
                            Vrho.start=1,
                            # Priors
                            mubeta=0, Vbeta=1.0E6,
                            mugamma=0, Vgamma=1.0E6,
                            priorVrho="1/Gamma",
                            shape=1, rate=1,
                            # Various
                            seed=1234, verbose=1,
                            save.rho=0, save.p=0, save.N=0)
```

```
# Outputs
summary(mod.Kery2010.Nmix.iCAR$mcmc)

##
## Iterations = 20001:29991
## Thinning interval = 10
## Number of chains = 1
## Sample size per chain = 1000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                          Mean     SD Naive SE Time-series SE
## beta.(Intercept)       0.304  0.293  0.00926        0.03355
## beta.elevation         2.186  0.526  0.01663        0.10901
## beta.I(elevation^2)   -1.944  0.316  0.00999        0.05168
## gamma.(Intercept)     -0.798  0.173  0.00548        0.04808
## gamma.juldate         -0.165  0.069  0.00218        0.00486
## gamma.I(juldate^2)     0.145  0.056  0.00177        0.00355
## Vrho                  15.281  3.517  0.11122        0.36227
## Deviance            1383.345 45.305  1.43266        7.77491
##
## 2. Quantiles for each variable:
##
##                          2.5%      25%      50%      75%     97.5%
## beta.(Intercept)      -0.2697    0.113    0.299    0.497    0.8584
## beta.elevation         1.2885    1.828    2.126    2.472    3.5030
## beta.I(elevation^2)   -2.6399   -2.139   -1.920   -1.727   -1.3687
## gamma.(Intercept)     -1.1013   -0.918   -0.820   -0.672   -0.4640
## gamma.juldate         -0.2974   -0.209   -0.167   -0.121   -0.0243
## gamma.I(juldate^2)     0.0405    0.107    0.146    0.181    0.2670
## Vrho                   9.6937   12.807   14.837   17.277   23.6666
## Deviance            1300.6648 1351.539 1381.030 1411.153 1479.4193
```

```r
# Spatial random effects
rho.pred <- mod.Kery2010.Nmix.iCAR$rho.pred
r.rho.pred <- rasterFromXYZ(cbind(coordinates(landscape),rho.pred))
plot(r.rho.pred)
# Mean abundance by site
ma <- apply(sites.sp@data[,3:5],1,mean,na.rm=TRUE)
points(sites.sp,pch=".",cex=2)
points(sites.sp,pch=1,cex=ma/2)
```

```r
# Predictions
nsamp <- dim(mod.Kery2010.Nmix.iCAR$mcmc)[1]
# Abundance-elevation
beta <- as.matrix(mod.Kery2010.Nmix.iCAR$mcmc[,1:3])
tbeta <- t(beta)
N <- matrix(NA,nrow=nsamp,ncol=npred)
# Simplified way of obtaining samples for rho
rho.samp <- sample(rho.pred,nsamp,replace=TRUE)
for (i in 1:npred) {
    N[,i] <- exp(X[i,] %*% tbeta + rho.samp)
}
N.est.Nmix.iCAR <- apply(N,2,mean)
N.q1.Nmix.iCAR <- apply(N,2,quantile,0.025)
N.q2.Nmix.iCAR <- apply(N,2,quantile,0.975)

# Detection-Julian date
gamma <- as.matrix(mod.Kery2010.Nmix.iCAR$mcmc[,4:6])
tgamma <- t(gamma)
delta <- matrix(NA,nrow=nsamp,ncol=npred)
for (i in 1:npred) {
    delta[,i] <- inv.logit(X[i,] %*% tgamma)
}
delta.est.Nmix.iCAR <- apply(delta,2,mean)
delta.q1.Nmix.iCAR <- apply(delta,2,quantile,0.025)
delta.q2.Nmix.iCAR <- apply(delta,2,quantile,0.975)
```

4.5.5 Comparing predictions from the three different models

```r
# Expected abundance - Elevation
par(mar=c(4,4,1,1),cex=1.4,tcl=+0.5)
plot(elev.seq,N.est.pois,type="l",
     xlim=c(500,3000),
     ylim=c(0,7),
```

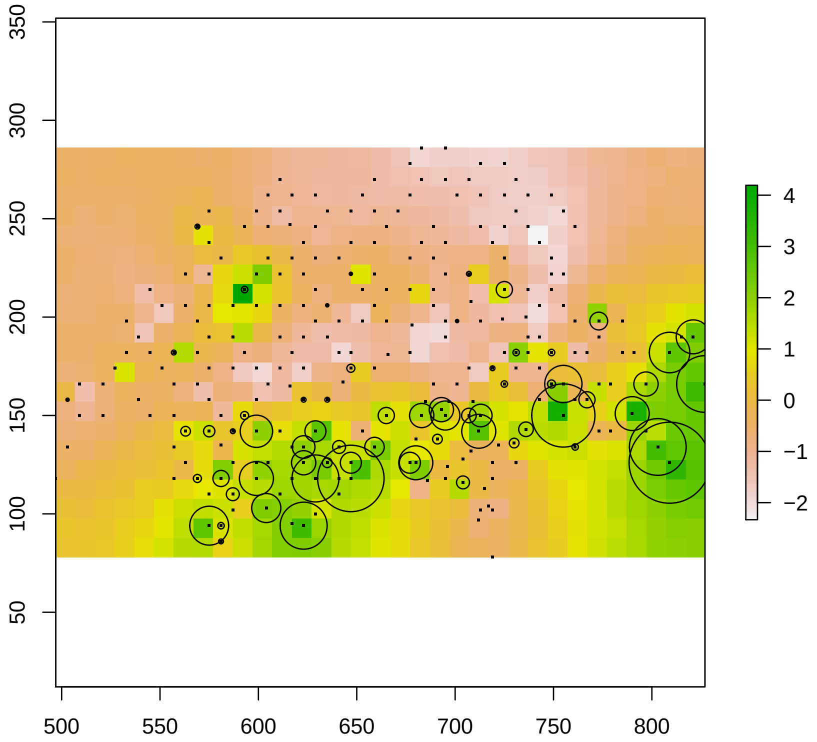


Figure 4.7: **Estimated spatial random effects.** Locations of observation quadrats are represented by dots. The mean abundance on each quadrat is represented by a circle of size proportional to abundance.

```
     lwd=2,
     xlab="Elevation (m a.s.l.)",
     ylab="Expected abundance",
     axes=FALSE)
#lines(elev.seq,N.q1.pois,lty=3,lwd=1)
#lines(elev.seq,N.q2.pois,lty=3,lwd=1)
axis(1,at=seq(500,3000,by=500),labels=seq(500,3000,by=500))
axis(2,at=seq(0,7,by=1),labels=seq(0,7,by=1))
# Nmix
lines(elev.seq,N.est.Nmix,lwd=2,col="red")
#lines(elev.seq,N.q1.Nmix,lty=3,lwd=1,col="red")
#lines(elev.seq,N.q2.Nmix,lty=3,lwd=1,col="red")
# Nmix.iCAR
lines(elev.seq,N.est.Nmix.iCAR,lwd=2,col="dark green")
#lines(elev.seq,N.q1.Nmix.iCAR,lty=3,lwd=1,col="dark green")
#lines(elev.seq,N.q2.Nmix.iCAR,lty=3,lwd=1,col="dark green")
```

```
# Detection probability - Julian date
par(mar=c(4,4,1,1),cex=1.4,tcl=+0.5)
plot(juldate.seq,delta.est.Nmix,type="l",
     xlim=c(100,200),
     ylim=c(0,1),
     lwd=2,
     col="red",
     xlab="Julian date",
     ylab="Detection probability",
     axes=FALSE)
lines(juldate.seq,delta.q1.Nmix,lty=3,lwd=1,col="red")
lines(juldate.seq,delta.q2.Nmix,lty=3,lwd=1,col="red")
axis(1,at=seq(100,200,by=20),labels=seq(100,200,by=20))
axis(2,at=seq(0,1,by=0.2),labels=seq(0,1,by=0.2))
# Nmix.iCAR
lines(juldate.seq,delta.est.Nmix.iCAR,lwd=2,col="dark green")
lines(juldate.seq,delta.q1.Nmix.iCAR,lty=3,lwd=1,col="dark green")
lines(juldate.seq,delta.q2.Nmix.iCAR,lty=3,lwd=1,col="dark green")
```

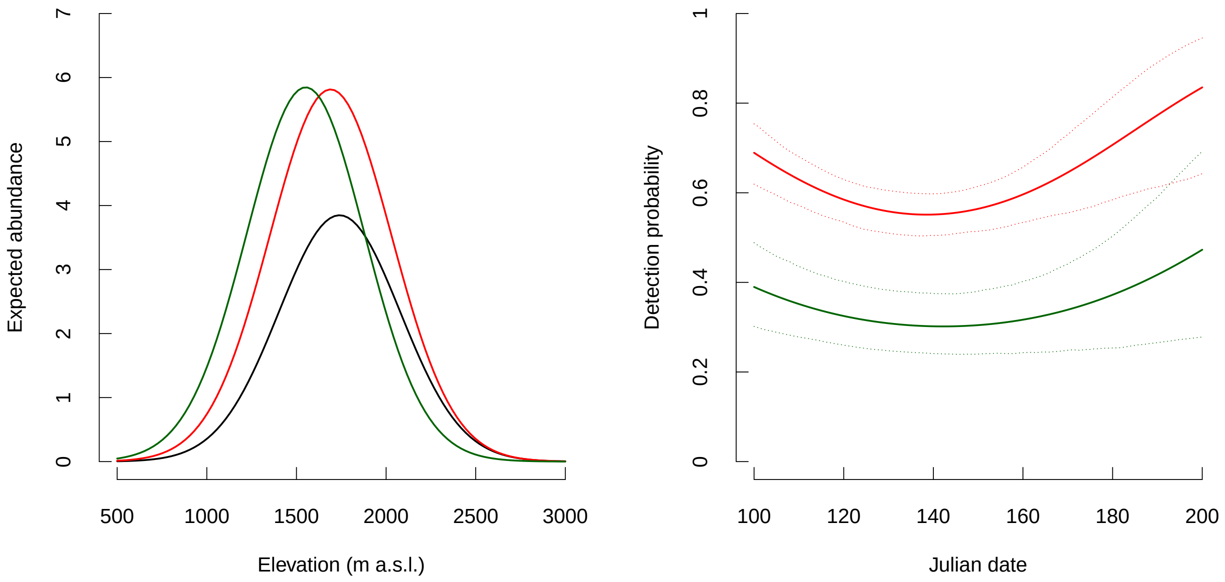


Figure 4.8: **Comparing predictions from the three different models.** The three different models are: Poisson (black), N-mixture (red) and N-mixture with iCAR process (green). The plain lines represent the predictive posterior mean of the abundance or of the probability of detection while the dashed lines represent the quantiles at 95% of the predictive posterior given parameter uncertainty.

CHAPTER 5

Some technical aspects of parameter inference

5.1 Likelihood for site-occupancy models

As previously detailed in the mathematical formulation of the site-occupancy model, let's consider the random variable z_i describing habitat suitability at site i. The random variable z_i can take value 1 or 0 depending on the fact that the habitat is suitable ($z_i = 1$) or not ($z_i = 0$). Random variable z_i can be assumed to follow a Bernoulli distribution of parameter θ_i. In this case, θ_i is the probability that the habitat is suitable. Several visits at time t_1, t_2, etc., can occur at site i. Let's consider the random variable y_{it} representing the presence of the species at site i and time t. The species is observed at site i ($\sum_t y_{it} \geq 1$) only if the habitat is suitable ($z_i = 1$). The species is unobserved at site i ($\sum_t y_{it} = 0$) if the habitat is not suitable ($z_i = 0$), or if the habitat is suitable ($z_i = 1$) but the probability δ_{it} of detecting the species at site i and time t is inferior to 1. Given H_i the set of observations (list of presence/absence) at site i, the likelihood L for site-occupancy models can be computed as follow (Eq. 5.1).

$$
\begin{aligned}
& L = \textstyle\prod_i \mathrm{p}(H_i) \\
& \text{if } \textstyle\sum_t y_{it} \geq 1 \quad && \mathrm{p}(H_i) = \mathrm{p}(z_i = 1) \textstyle\prod_t \mathrm{p}(y_{it}) \\
& && \mathrm{p}(H_i) = \theta_i \textstyle\prod_{t=1} \mathrm{p}(y_{it}) \\
& && \text{with } \mathrm{p}(y_{it} = 1) = \delta_{it} \text{ and } \mathrm{p}(y_{it} = 0) = 1 - \delta_{it} \\
& \text{if } \textstyle\sum_t y_{it} = 0 \quad && \mathrm{p}(H_i) = \mathrm{p}(z_i = 0) + \mathrm{p}(z_i = 1) \textstyle\prod_t \mathrm{p}(y_{it} = 0) \\
& && \mathrm{p}(H_i) = (1 - \theta_i) + \theta_i \textstyle\prod_t (1 - \delta_{it})
\end{aligned}
\tag{5.1}
$$

For site-occupancy models, there is a strong advantage of visiting a site several times.

When a site is visited several times for observation, if the species has been observed at least once during the different visits, we can assert that the habitat at this site is suitable. And the fact that the species can be unobserved at this site is only due to imperfect detection. For more details, please refer to the original paper by MacKenzie *et al.* (2002) and the very pedagogical note by Bailey & Adams (2005).

5.2 Random walk to estimate latent variables in N-mixture models

Section to be written...

5.3 Adaptive Metropolis within Gibbs

Except for the variance of the spatial random effects of the iCAR models, for which we proposed conjugate priors, we used an adaptive Metropolis algorithm (Metropolis *et al.*, 1953; Robert & Casella, 2004) within Gibbs sampler (Casella & George, 1992; Gelfand & Smith, 1990) to draw the samples of the posterior distribution for model's parameters.

The proposal distribution in the Metropolis algorithm is a Normal distribution centered on the current parameter value and with standard deviation σ. The standard deviation σ is set to 1 at the beginning of the MCMC and is continuously adjusted so that the acceptance rate is 0.44 for non-hierarchical models (`hSDM.binomial()` and `hSDM.poisson()` functions) and 0.234 for hierarchical models (other hSDM functions). These values of acceptance rate (0.44 for low-dimensional models and 0.234 for high-dimensional models) ensure a better efficiency of the Metropolis algorithm and a faster MCMC convergence (Roberts *et al.*, 1997; Roberts & Rosenthal, 2009; Roberts *et al.*, 2001).

The actualized value $\sigma^\star$ of the standard deviation of the proposal distribution is computed from the current acceptance rate A, the optimal acceptance rate r (0.44 or 0.234) and the current standard deviation σ (Eq. 5.2).

$$\begin{aligned} \text{if } A \geq r \quad & \sigma^\star = \sigma(2-(1-A)/(1-r)) \\ \text{else} \quad & \sigma^\star = \sigma/(2-A/r) \end{aligned} \tag{5.2}$$

The tuning of the proposal is only done during the burnin period. After the burnin period, the standard deviation of the proposal distribution is fixed at the current value. The adaptive Metropolis within Gibbs is written in C code and compiled to optimize computation efficiency.

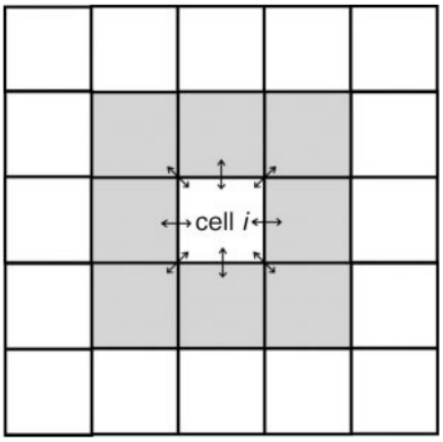


Figure 5.1: **Diagram of the grid cell neighborhood used in the intrinsic conditional autoregressive (iCAR) models**

5.4 Intrinsic conditional autoregressive (iCAR) model

To capture the spatial autocorrelation, we employ a Gaussian intrinsic conditional autoregressive (iCAR) model (Besag, 1974). To specify this model, we assume that the conditional distribution of the spatial random effect ρ_j in cell j, given values for the spatial random effect in all other cells $j' \neq j$, depends only on the spatial random effect of the neighbouring cells of j. Here, we specify that cell j' is a neighbor of j if their boundaries intersect (Fig. 5.1). In the actual version of the iCAR process used in the **hSDM** R package, the spatial effect for any given cell depends only on the values of ρ for the cells in its neighborhood, and the neighborhood encompasses only the height immediately adjacent cells ("king movement" in chess). The neighborhood could alternatively be defined to be larger, and different weights could be assigned to cells at different distances. Formally, the Gaussian iCAR model for the spatial random effect at cell i can be presented by a conditional distribution (Eq. 5.3).

$$\text{p}(\rho_j|\rho_{j'}) \sim \mathcal{N}ormal(\mu_j, V_\rho/n_j)$$

(5.3) μ_j: mean of $\rho_{j'}$ in the neighborhood of j.
V_ρ: variance of the spatial random effects.
n_j: number of neighbors for cell j.

The variance of the spatial random effects V_ρ is also a parameter to be estimated. We use a conjugate prior to infer V_ρ and we propose two prior distributions: an Inverse-Gamma distribution with shape and rate parameters or a Uniform distribution with zero for the lower bound of the interval and one parameter for the upper bound.

5.5 Difference between site-occupancy and ZIB models

Both site-occupancy or ZIB models (with `hSDM.siteocc()` or `hSDM.ZIB()` functions respectively) can be used to model the presence-absence of a species taking into account imperfect detection. The site-occupancy model can be used in all cases but can be less convenient and slower to fit when the repeated visits at each site are made under the exact same observation conditions. In this particular case, a Binomial distribution can be used for the observation process and we suggest the use of a ZIB model for computational efficiency (see example in Section 4.3).

On the contrary, when the data-set includes several visits at each site under different observation conditions, a Bernoulli distribution must be used for the observation process (not a Binomial distribution). In this case, the ZIB models must not be used. For `hSDM.ZIB()` functions, the fact that the observations are done on a same site is implicitely assumed by the data structure (see `presences` and `trials` arguments for each observation/site). Thus, for `hSDM.ZIB()` functions, there is no `site` argument to specify the site for each observation such as for `hSDM.siteocc()` functions.

5.6 Difference between N-mixture and ZIP models

For counts data with imperfect detection, both N-mixture and ZIP models can be used (with `hSDM.Nmixture()` or `hSDM.ZIP()` functions respectively). But the interpretation of the underlying processes and the structure of the data that can be used differ between the two models.

For the N-mixture model, the suitability process is modelled by a Poisson distribution. In this case, we interpret the number of individuals at one site as a function of environmental variables and we assume that there is more individuals when the habitat is more suitable. In a second step, the observability process is modelled by a Binomial distribution. We only see a fraction of the individuals present at one site due to observation conditions (Eq. 5.4).

For the N-mixture model, several visits can occur at one site under different observation conditions (see response variable y, explicative variables W and probability δ indexed on both i and t).

$$
\begin{aligned}
&\textbf{Ecological process:} && \\
& && N_i \sim \mathcal{P}oisson(\lambda_i) \\
& && \log(\lambda_i) = X_i\beta \\
&\textbf{Observation process:} && \\
& && y_{it} \sim \mathcal{B}inomial(N_i, \delta_{it}) \\
& && \text{logit}(\delta_{it}) = W_{it}\gamma
\end{aligned}
\tag{5.4}
$$

For the ZIP model, the suitability process is modelled by a Bernoulli distribution. In this case, we interpret the habitat at a particular site to be suitable for the species ($z_i = 1$) or not ($z_i = 0$). Then, the process determining the number of individuals observed at suitable sites (the abundance) is modelled by a Poisson distribution. Thus, this second process can include both ecological or detection factors explaining the abundance of the species at suitable sites (Eq. 5.5). Flores *et al.* (2009) provide a good example of the application of a ZIP model to the distribution of tree saplings.

$$
\begin{aligned}
&\textbf{Suitability process:} && \\
& && z_i \sim \mathcal{B}ernoulli(\theta_i) \\
& && \text{logit}(\theta_i) = X_i\beta \\
&\textbf{Abundance process:} && \\
& && y_i \sim \mathcal{P}oisson(z_i, \lambda_i) \\
& && \log(\lambda_i) = W_i\gamma
\end{aligned}
\tag{5.5}
$$

Note that ZIP models cannot be used when the data-set includes several visits by site. The likelihood of the ZIP models does not account for the fact that if the species is observed at least once at one site during the visits, then the habitat at this site is obviously suitable. Thus, such as for `hSDM.ZIB()` functions, `hSDM.ZIP()` functions do not have a `site` argument to specify the site for each observation (which is the case for `hSDM.Nmixture()` functions).

5.7 Difference between `site` and `spatial.entity`

For site-occupancy and N-mixture models taking into account both imperfect detection and spatial correlation, the user must make the difference between the `site` argument which indicates the site where the repeated observations have been made, and the `spatial.entity` argument which indicates the spatial entity for the spatial correlation process. These two spatial levels are clearly distinct. Thus, several sites (places visited) can be located in the same spatial entity (region, state, etc.).

Of course, in some particular cases, the site and the spatial entity can be confounded.

Nonetheless, it is recommended to choose a resonable spatial scale (not too fine) for the spatial correlation process. With a limited number of spatial entities, there is a possibility to have more observations in each spatial entity. This should increase the amount of information for estimating spatial random effects and also speed up the computation with fewer spatial random effects to estimate. But the number of spatial entities should also be large enough to be able to estimate the variance of the spatial random effects. For example, Maas & Hox (2005) suggest a minimum of 50 levels for a random effect factor.

5.8 Computing the neighborhood for iCAR model

Section to be written...

- **raster** package
- The landscape raster must be projected (otherwise, torus system)
- function `adjacent()`

5.9 Forecasting species distribution under future climate change

Section to be written...

- How to obtain predictions
- What about the spatial random effects, do we include them ?

5.10 Computation time

When comparing OpenBUGS and hSDM outputs, computation times are given for guidance. The computer used for performing the statistical analysis had 4 processors of 2.5 GHz and 4Go of RAM. There is no parallelization implemented when running the Gibbs sampler, so that only one processor is used. The operating system installed on the computer was Linux Debian 7.0.

5.11 Package development, git and Sourceforge

Section to be written...

- Git repository on Sourceforge: git://git.code.sf.net/p/hsdm/code hsdm-code
- Web site on Sourceforge: `http://hSDM.sf.net`
- Number of line of code

Development work to be done:

- Analytically estimate the latent variables in N-mixture models
- Probit link function for Binomial model
- Random site effect for observability process
- Multispecies approach

CHAPTER 6

Conclusion

Section to be written...

- Advantages of hSDM
  - User friendly
  - Speed
  - Can handle large data-sets
- Recommendations
  - Fitting complex models imply the use of data-sets providing sufficient information (in number of observations, in number of repetitions, etc.).
  - Users must be careful especially with non-identifiable over-parametrized model.
  - Using hierarchical Bayesian species distribution models is only an option. Be careful with "statistical machismo" (see `http://dynamicecology.wordpress.com/2012/09/11/statistical-machismo/` and Hodges & Reich (2010) for example).

CHAPTER 7

Acknowledgements

Support was provided by Cirad and FRB (Fondation pour la Recherche sur la Biodiversité) through the BioSceneMada project (project agreement AAP-SCEN-2013 I).

Bibliography

Araujo MB, Guisan A (2006) Five (or so) challenges for species distribution modelling. *Journal of Biogeography*, **33**, 1677–1688.

Bailey L, Adams MJ (2005) *Occupancy Models to Study Wildlife*. 2005-3096. U.S. Geological Survey. URL http://fresc.usgs.gov/products/fs/fs2005-3096.pdf.

Bailey LL, Simons TR, Pollock KH (2004) Estimating site occupancy and species detection probability parameters for terrestrial salamanders. *Ecological Applications*, **14**, 692–702.

Besag J (1974) Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 192–236.

Besag J, York J, Mollié A (1991) Bayesian image restoration, with two applications in spatial statistics. *Annals of the Institute of Statistical Mathematics*, **43**, 1–20.

Brezger A, Kneib T, Lang S (2005) Bayesx: Analyzing bayesian structural additive regression models. *Journal of Statistical Software*, **14**, 1–22. URL http://www.jstatsoft.org/v14/i11.

Casella G, George EI (1992) Explaining the Gibbs Sampler. *American Statistician*, **46**, 167–174.

Chelgren ND, Adams MJ, Bailey LL, Bury RB (2011) Using multilevel spatial models to understand salamander site occupancy patterns after wildfire. *Ecology*, **92**, 408–421.

Chen G, Kéry M, Plattner M, Ma K, Gardner B (2013) Imperfect detection is the rule rather than the exception in plant distribution studies. *Journal of Ecology*, **101**, 183–191.

Choquet R, Rouan L, Pradel R (2009) Program e-surge: a software application for fitting multievent models. In: *Modeling demographic processes in marked populations*, pp. 845–865. Springer.

Cressie NA, Cassie NA (1993) *Statistics for spatial data*, vol. 900. Wiley New York.

Dorazio RM, Royle JA, Soderstrom B, Glimskar A (2006) Estimating species richness and accumulation by modeling species occurrence and detectability. *Ecology*, **87**, 842–854.

Dormann CF, McPherson JM, Araujo M, *et al.* (2007) Methods to account for spatial autocorrelation in the analysis of species distributional data: a review. *Ecography*, **30**, 609–628. URL http://dx.doi.org/10.1111/j.2007.0906-7590.05171.x.

Elith J, Leathwick JR (2009) Species distribution models: Ecological explanation and prediction across space and time. *Annu. Rev. Ecol. Evol. Syst.*, **40**, 677–697. URL http://dx.doi.org/10.1146/annurev.ecolsys.110308.120159.

Fiske I, Chandler R (2011) unmarked: An R package for fitting hierarchical models of wildlife occurrence and abundance. *Journal of Statistical Software*, **43**, 1–23. URL http://www.jstatsoft.org/v43/i10/.

Flores O, Rossi V, Mortier F (2009) Autocorrelation offsets zero-inflation in models of tropical saplings density. *Ecological Modelling*, **220**, 1797–1809.

Gelfand AE, Schmidt AM, Wu S, Silander JA, Latimer A, Rebelo AG (2005) Modelling species diversity through species level hierarchical modelling. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **54**, 1–20.

Gelfand AE, Smith AFM (1990) Sampling-Based Approaches to Calculating Marginal Densities. *Journal of American Statistical Association*, **85**, 398–409.

Gray TN (2012) Studying large mammals with imperfect detection: Status and habitat preferences of wild cattle and large carnivores in eastern cambodia. *Biotropica*, **44**, 531–536.

Guisan A, Thuiller W (2005) Predicting species distribution: offering more than simple habitat models. *Ecology Letters*, **8**, 993–1009.

Guisan A, Zimmermann NE (2000) Predictive habitat distribution models in ecology. *Ecological Modelling*, **135**, 147–186.

Hodges JS, Reich BJ (2010) Adding spatially-correlated errors can mess up the fixed effect you love. *The American Statistician*, **64**, 325–334. doi:10.1198/tast.2010.10052. URL http://dx.doi.org/10.1198/tast.2010.10052.

Johnson DS, Conn PB, Hooten MB, Ray JC, Pond BA (2013) Spatial occupancy models for large data sets. *Ecology*, **94**, 801–808.

Keitt TH, Bjørnstad ON, Dixon PM, Citron-Pousty S (2002) Accounting for spatial pattern when modeling organism-environment interactions. *Ecography*, **25**, 616–625.

Kühn I, Bierman SM, Durka W, Klotz S (2006) Relating geographical variation in pollination types to environmental and spatial factors using novel statistical methods. *New Phytologist*, **172**, 127–139.

Kéry M, Andrew Royle J (2010) Hierarchical modelling and estimation of abundance and population trends in metapopulation designs. *Journal of Animal Ecology*, **79**, 453–461.

Kéry M, Gardner B, Monnerat C (2010) Predicting species distributions from checklist data using site-occupancy models. *Journal of Biogeography*, **37**, 1851–1862.

Kéry M, Royle JA, Schmid H (2005) Modeling avian abundance from replicated counts using binomial mixture models. *Ecological applications*, **15**, 1450–1461.

Kéry M, Schaub M (2012) *Bayesian population analysis using WinBUGS: a hierarchical perspective*. Academic Press.

Kéry M, Schmidt BR (2008) Imperfect detection and its consequences for monitoring for conservation. *Community Ecology*, **9**, 207–216.

Lahoz-Monfort JJ, Guillera-Arroita G, Wintle BA (2014) Imperfect detection impacts the performance of species distribution models. *Global Ecology and Biogeography*, **23**, 504–515. doi:10.1111/geb.12138. URL http://dx.doi.org/10.1111/geb.12138.

Latimer AM, Wu SS, Gelfand AE, Silander JA (2006) Building statistical models to analyze species distributions. *Ecological Applications*, **16**, 33–50.

Lee D (2013) Carbayes: An r package for bayesian spatial modeling with conditional autoregressive priors. *Journal of Statistical Software*, **55**. URL http://www.jstatsoft.org/v55/i13.

Legendre P (1993) Spatial autocorrelation: trouble or new paradigm? *Ecology*, **74**, 1659–1673.

Lichstein JW, Simons TR, Shriner SA, Franzreb KE (2002) Spatial autocorrelation and autoregressive models in ecology. *Ecological Monographs*, **72**, 445–463.

Lunn D, Spiegelhalter D, Thomas A, Best N (2009) The bugs project: Evolution, critique and future directions. *Statistics in medicine*, **28**, 3049–3067.

Maas CJ, Hox JJ (2005) Sufficient sample sizes for multilevel modeling. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, **1**, 86.

MacKenzie DI (2006) *Occupancy estimation and modeling: inferring patterns and dynamics of species occurrence*. Academic Press.

MacKenzie DI, Nichols JD, Lachman GB, Droege S, Andrew Royle J, Langtimm CA (2002) Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, **83**, 2248–2255.

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH, Teller E (1953) Equation of state calculations by fast computing machines. *The journal of chemical physics*, **21**, 1087–1092.

Miller J, Franklin J, Aspinall R (2007) Incorporating spatial dependence in predictive vegetation models. *Ecological Modelling*, **202**, 225–242.

Monk J (2014) How long should we ignore imperfect detection of species in the marine environment when modelling their distribution? *Fish and Fisheries*, **15**, 352–358.

Nichols JD (1992) Capture-recapture models. *BioScience*, pp. 94–102.

Poley LG, Pond BA, Schaefer JA, Brown GS, Ray JC, Johnson DS (2014) Occupancy patterns of large mammals in the far north of ontario under imperfect detection and spatial autocorrelation. *Journal of Biogeography*, **41**, 122–132.

R Core Team (2014) *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org.

Robert CP, Casella G (2004) *Monte Carlo statistical methods*, vol. 319. Citeseer.

Roberts GO, Gelman A, Gilks WR, *et al.* (1997) Weak convergence and optimal scaling of random walk metropolis algorithms. *The annals of applied probability*, **7**, 110–120.

Roberts GO, Rosenthal JS (2009) Examples of adaptive mcmc. *Journal of Computational and Graphical Statistics*, **18**, 349–367.

Roberts GO, Rosenthal JS, *et al.* (2001) Optimal scaling for various metropolis-hastings algorithms. *Statistical science*, **16**, 351–367.

Rota CT, Fletcher RJ, Evans JM, Hutto RL (2011) Does accounting for imperfect detection improve species distribution models? *Ecography*, **34**, 659–670.

Royle JA (2004) N-mixture models for estimating population size from spatially replicated counts. *Biometrics*, **60**, 108–115.

Royle JA, Dorazio RM (2008) *Hierarchical modeling and inference in ecology: the analysis of data from populations, metapopulations and communities.* Academic Press.

Royle JA, Dorazio RM, Link WA (2007) Analysis of multinomial models with unknown index using data augmentation. *Journal of Computational and Graphical Statistics*, **16**, 67–85.

Rue H, Martino S, Chopin N (2009) Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology)*, **71**, 319–392.

Sinclair SJ, White MD, Newell GR (2010) How useful are species distribution models for managing biodiversity under future climates? *Ecology and Society*, **15**, 8.

Smith SI (1868) The geographical distribution of animals. *The American Naturalist*, **2**, pp. 124–131. URL http://www.jstor.org/stable/2447129.

Sokal RR, Oden NL (1978) Spatial autocorrelation in biology: 2. some biological implications and four ap- plications of evolutionary and ecological interest. *Biological Journal of the Linnean Society*, **10**, 229–249.

Stan Development Team (2014) *Stan Modeling Language Users Guide and Reference Manual, Version 2.2*. URL http://mc-stan.org/.

Thuiller W, Guéguen M, Georges D, *et al.* (2014) Are different facets of plant diversity well protected against climate and land cover changes? a test study in the french alps. *Ecography*.

Wallace AR (1876) *The geographical distribution of animals: with a study of the relations of living and extinct faunas as elucidating the past changes of the earth's surface*. Macmillan & Co., London.

White GC, Burnham KP (1999) Program mark: survival estimation from populations of marked animals. *Bird study*, **46**, S120–S139.

Williams BK, Nichols JD, Conroy MJ (2002) *Analysis and management of animal populations: modeling, estimation, and decision making*. Academic Press.