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

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

Ghislain Vieilledent[⋆,1]

[⋆] **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

Abstract

Work in progess...

Keywords: R

1 Introduction

2 Species distribution models

2.1 Binomial model

2.1.1 Mathematical formulation

Let's consider a random variable $y_{i(jt)}$ (abbreviated y_i) representing the total number of presences of a species after several visits $v_{i(jt)}$ (abbreviated v_i) at a particular site j at time t (t can indicate a day, a year, etc. thus allowing several visits at time t). 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 $\theta_{i(jt)}$ (Eq. 1). $\theta_{i(jt)}$ (abbreviated θ_i) can be interpreted as the probability of presence of the species at site j and time t. Using a logit link function, θ_i can be expressed as a linear model combining explicative variables $X_{i(jt)}$ and parameters β (Eq. 1).

$$\begin{gathered} y_i \sim \mathcal{B}inomial(v_i, \theta_i) \\ \text{logit}(\theta_i) = X_{i(jt)}\beta \end{gathered} \tag{1}$$

Using this statistical model, we aim at representing a "suitability process". Given environmental variables $X_{i(jt)}$, how much is habitat at site j and at time t 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 can generate a virtual data-set on the basis of the Binomial model described above (Eq. 1). In the most general case (presence/absence data or presence/pseudo-absence data), a site j is visited once at time t and $v_i = 1$. Thus, the random variable y_i follows a Bernoulli distribution having parameters θ_i and habitat characteristics $X_{i(jt)}$ are fixed for site j. We will generate a virtual data-set in this particular case. For data generation, we can import virtual altitudinal data in R. Altitude will be used as an explicative variable to determine habitat suitability, i.e. the probability of presence of a virtual species. Altitudinal data are available on the website of the **hSDM** R package hosted on Sourceforge (`http://hsdm.sourceforge.net/altitude.csv`).

These data can be transformed into a raster using function `rasterFromXYZ()` from the **raster** package. The raster has 2500 cells (50 columns and 50 rows) and the altitude is comprised roughly between 100 and 600 m (Fig. 1). For linear models, explicative variables are usually centered and scaled to facilitate inference and interpretation of model parameters.

```
# Import altitudinal data
require(raster)
fname <- "http://hsdm.sourceforge.net/altitude.csv"
alt.df <- read.csv(fname,header=TRUE)
alt.orig <- rasterFromXYZ(alt.df)
plot(alt.orig)
# Center and scale altitudinal data
alt <- scale(alt.orig,center=TRUE,scale=TRUE)
plot(alt)
```

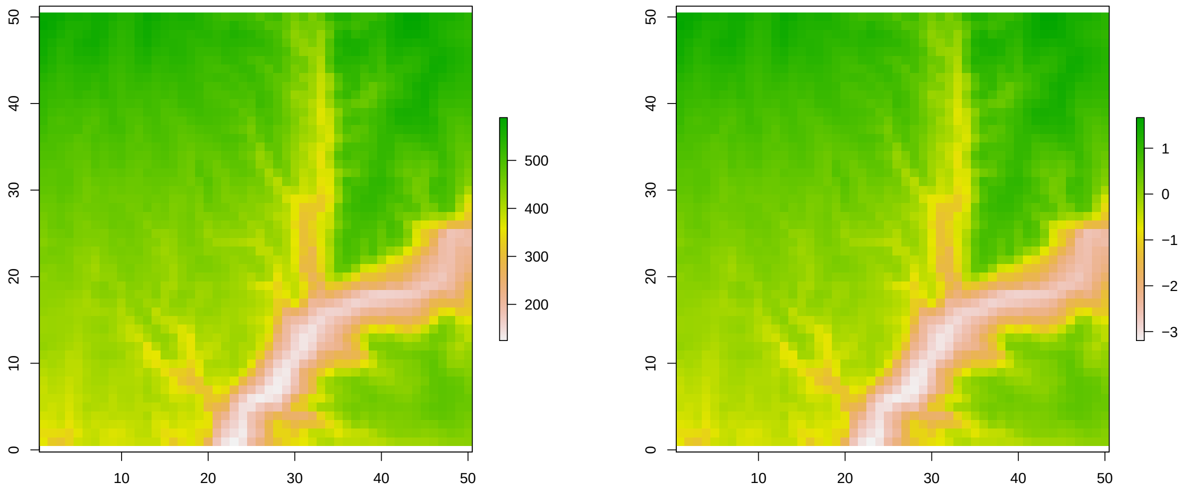

Figure 1: **Altitudinal data**. Original values (in m) on the left. Centered and scaled values on the right.

A quadratic effect of the altitude (variable denoted x) is used to compute the probability of presence of the species (Eq. 2).

$$\begin{aligned} y_i &\sim \mathcal{B}ernoulli(\theta_i) \\ \text{logit}(\theta_i) &= \beta_0 + \beta_1 x_i + \beta_2 x_i^2 \end{aligned} \tag{2}$$

We fix the parameters to $\beta_0 = 1$, $\beta_1 = 2$ and $\beta_2 = -4$. The species has a higher probability of presence at intermediate altitudinal values (Fig. 2).

```
# Load hSDM library
library(hSDM)
# Target parameters
beta.target <- matrix(c(1,2,-4),ncol=1)
# Matrix of covariates (including the intercept)
X <- cbind(rep(1,ncell(alt)),values(alt),(values(alt))^2)
# Probability of presence as a quadratic function of altitude
logit.theta <- X %*% beta.target
theta <- inv.logit(logit.theta)
# Transform the probability of presence into a raster
theta <- rasterFromXYZ(cbind(coordinates(alt),theta))
# Plot the probability of presence
## require(fields)
## col.prob <- colorRampPalette(c("transparent","blue","green",
##                                "yellow","orange","red","brown"))
## plot(theta,col=col.prob(255))
plot(theta,col=rev(heat.colors(255)))
```

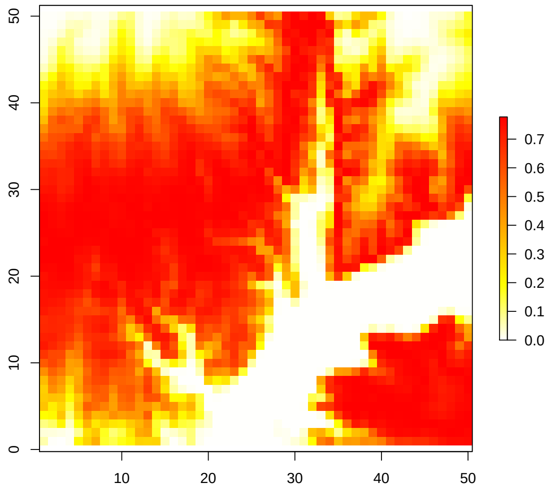

Figure 2: **Probability of presence**.

We can assume a number n of points 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 points given our model (Fig. 3).

```
# Load dismo library
require(dismo) # For randomPoints() function
```

```
# Number of observation points
nobspt <- 200
# Set seed for repeatability
seed <- 1234
# Sample the observations in the landscape
set.seed(seed)
obs <- randomPoints(alt,nobspt)
# Extract altitude data for observations
alt.obs <- extract(alt,obs)
# Compute theta for these observations
X <- cbind(rep(1,nobspt),alt.obs,alt.obs^2)
logit.theta.obs <- X %*% beta.target
theta.obs <- inv.logit(logit.theta.obs)
# Simulate observations
trials <- rep(1,nobspt)
set.seed(seed)
Y <- rbinom(nobspt,trials,theta.obs)
# Group explicative and response variables in a data-frame
data.obs.df <- data.frame(Y,trials,alt=X[,2],alt2=X[,3])
# Transform observations in a spatial object
require(sp)
data.obs <- SpatialPointsDataFrame(coords=coordinates(obs),data=data.obs.df)
# Plot observations
plot(alt.orig)
points(data.obs[data.obs$Y==1,],pch=16)
points(data.obs[data.obs$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" and "alt2" in our example).

```
data.pred <- data.frame(alt=values(alt),alt2=(values(alt))^2)
```

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.

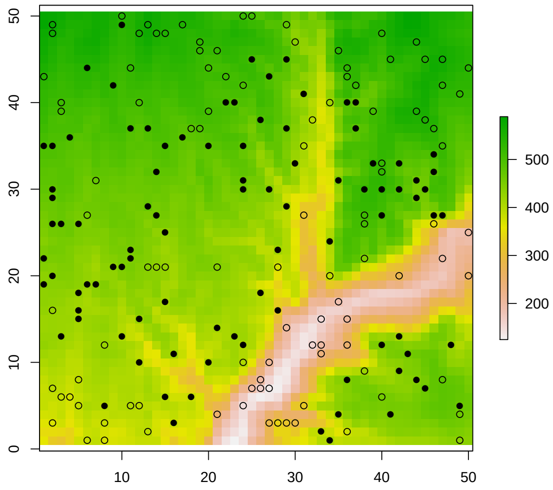

Figure 3: **Observation points**. Presences (full circles) and absences (empty circles) are localized on the altitude map (in m).

```
mod.hSDM.binomial <- hSDM.binomial(presences=data.obs$Y,
                                   trials=data.obs$trials,
                                   suitability=~alt+alt2,
                                   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 $\beta_0 = 1$, $\beta_1 = 2$ and $\beta_2 = -4$ are within the 95% confidence interval of the parameter estimates.

```
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)   1.18 0.231  0.00732         0.0215
## beta.alt           1.57 0.556  0.01758         0.0821
## beta.alt2         -3.89 0.694  0.02194         0.1116
## Deviance         196.27 2.430  0.07685         0.2225
##
## 2. Quantiles for each variable:
##
##                     2.5%    25%    50%    75%  97.5%
## beta.(Intercept)   0.735   1.02   1.16   1.34   1.62
## beta.alt           0.468   1.16   1.57   1.97   2.64
## beta.alt2         -5.200  -4.37  -3.88  -3.37  -2.41
## Deviance         193.428 194.50 195.67 197.53 202.85
```

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

```
#== glm results for comparison
mod.glm <- glm(cbind(Y,trials-Y)~alt+alt2,family="binomial",data=data.obs)
summary(mod.glm)

##
## Call:
## glm(formula = cbind(Y, trials - Y) ~ alt + alt2, family = "binomial",
##     data = data.obs)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.766  -0.797   0.000   0.799   3.112
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.186      0.241    4.92  8.6e-07 ***
## alt            1.475      0.538    2.74   0.0061 **
## alt2          -3.736      0.735   -5.08  3.7e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##     Null deviance: 276.54  on 199  degrees of freedom
## Residual deviance: 193.15  on 197  degrees of freedom
## AIC: 199.1
##
## Number of Fisher Scoring iterations: 8
```

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. 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, `prob.p.latent`, is the predictive posterior mean of the latent variable θ (the probability of presence) for each observation.

```
str(mod.hSDM.binomial$prob.p.latent)

##  num [1:200] 0.17046 0.4498 0.6823 0.00542 0.05014 ...

summary(mod.hSDM.binomial$prob.p.latent)

##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.184   0.572   0.465   0.731   0.789
```

The second one, `prob.p.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 `prob.p.pred` is an `mcmc` object. Values in `prob.p.pred` can be used to plot the predicted probability of presence on the whole landscape and the uncertainty associated to the predictions (Fig 5).

```
# 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$prob.p.pred,2,mean)
theta.pred.2.5[] <- apply(mod.hSDM.binomial$prob.p.pred,2,quantile,0.025)
theta.pred.97.5[] <- apply(mod.hSDM.binomial$prob.p.pred,2,quantile,0.975)
# Plot the predicted probability of presence and uncertainty
plot(theta.pred.mean,col=rev(heat.colors(255)))
plot(theta.pred.2.5,col=rev(heat.colors(255)))
plot(theta.pred.97.5,col=rev(heat.colors(255)))
```

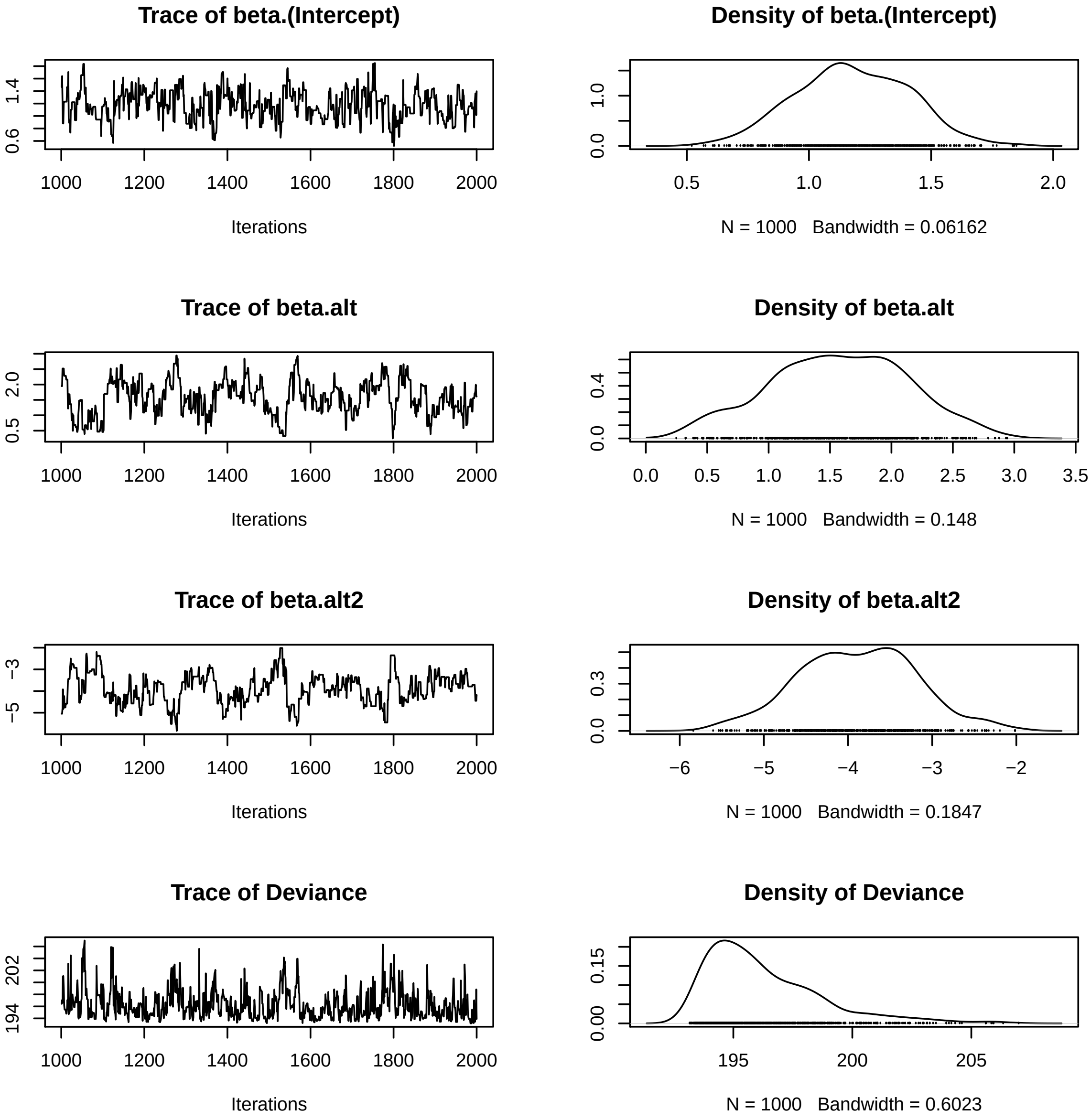

Figure 4: **Trace and density estimate for each variable of the MCMC.**

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. 6).

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

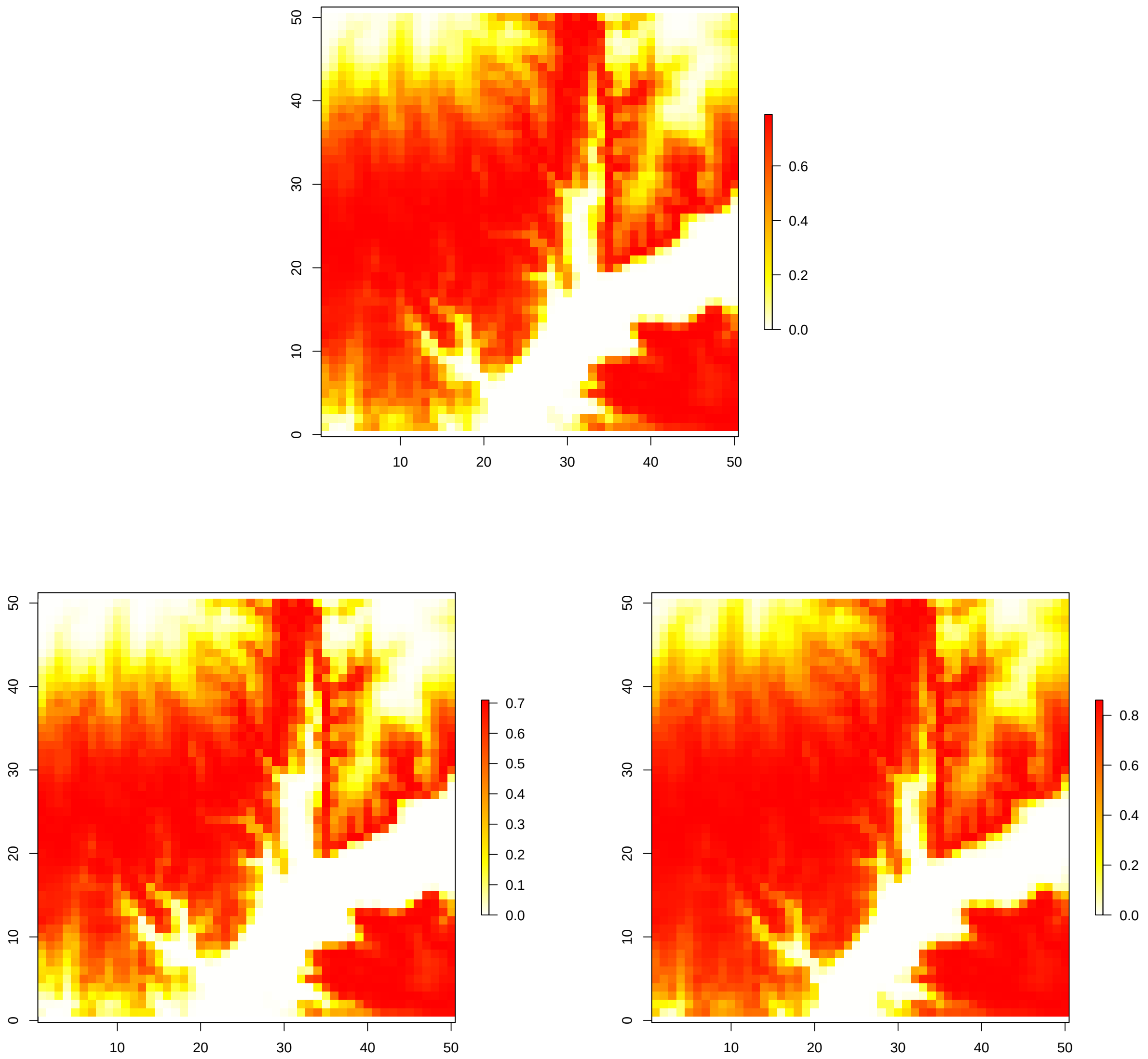

Figure 5: **Predicted probability of presence and uncertainty of predictions**. Mean probability of presence (top), predictions at 2.5% quantile (bottom left) and 97.5% quantile (bottom right) can be plotted from the `mcmc` object `plot.p.pred` returned by function `hSDM.binomial()`.

2.2 Site-occupancy model

2.2.1 Mathematical formulation

Occurrence of a species is typically not observed perfectly. Species traits, survey-specific conditions and site-specific characteristics may influence species detection probability which

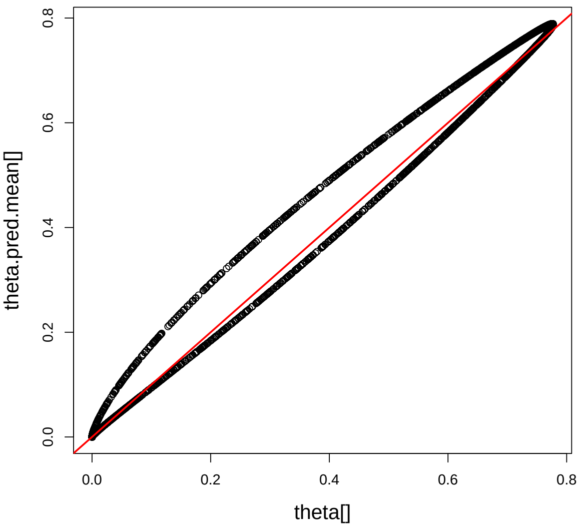

Figure 6: **Predicted vs. initial probabilities of presence**. Initial probabilities of presence are computed from the Binomial logistic regression model with fixed parameters.

is often < 1 (Chen *et al.*, 2013). Thus, observations might include false absences. For example, the habitat can be suitable and the species 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).

New classes of models, called site-occupancy models, were developed to solve the problems created by imperfect detection (MacKenzie *et al.*, 2002). These models combine two processes, an ecological process which describes habitat suitability and an observation process which takes into account imperfect detection. Site-occupancy models use information from repeated observations at each site to estimate detectability. Detectability may vary with site characteristics (e.g., habitat variables) or survey characteristics (e.g., weather conditions), whereas occupancy relates only to site characteristics.

Let's consider the random variable $z_{i(jt)}$ (abbreviated z_i) describing habitat suitability for observation i at site j and at time t. The random variable z_i can take value 1 or 0 depending on the fact that the habitat at site j and at time t is suitable ($z_i = 1$) or not ($z_i = 0$). Habitat at site j and time t is described by environmental variables $X_{i(jt)}$. Random variable z_i can be assumed to follow a Bernoulli distribution of parameter θ_i (Eq. 3). In this case, θ_i is the probability that the habitat is suitable. Let's consider also the random variable $y_{i(jt)}$ (abbreviated y_i) representing the total number of presences of a species after several visits $v_{i(jt)}$ (abbreviated v_i) at a particular site j at time t. Again, t can indicate a day, a year, etc. thus allowing several visits at time t. The species is observed ($y_i \geq 1$) only if the habitat is suitable ($z = 1$). The species is unobserved ($y_i = 0$) if the habitat is not suitable ($z_i = 0$) or if the habitat is suitable ($z_i = 1$) but the detection probability $\delta_{i(jt)}$ (abbreviated δ_i) is inferior to 1. Thus, the total number of presences y_i is assumed to follow a Binomial distribution with parameters $z_i\delta_i$ and v_i. Using a logit link function, δ_i can be expressed as a linear model combining explicative variables $W_{i(jt)}$ and parameters γ (Eq. 3). Typically, explicative variables $W_{i(jt)}$ 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.

$$\begin{array}{c}\textbf{Ecological process:}\\ z_i \sim \mathcal{B}ernoulli(\theta_i)\\ \text{logit}(\theta_i) = X_{i(jt)}\beta\\ \\ \textbf{Observation process:}\\ y_i \sim \mathcal{B}inomial(z_i\delta_i, v_i)\\ \text{logit}(\delta_i) = W_{i(jt)}\gamma \end{array} \tag{3}$$

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. 3). In most general cases, protocol experiments would include severals visits with varying survey conditions (e.g. weather conditions) to several sites with fixed sites characteristics (e.g. habitat variables). We will generate virtual data following this protocole and using the altitudinal data used in the previous example for the Binomial model (Sec. 2.1).

We draw at random the number of visits at each observation point used in the previous example (see Fig. 3 of Sec. 2.1).

```r
# Number of visits associated to each observation point
set.seed(seed)
v <- rpois(nobspt,2)
v[v==0] <- 1
# Vector of observation points
obspt <- vector()
for (i in 1:nobspt) {
    obspt <- c(obspt,rep(i,v[i]))
}
```

We fix the survey conditions for each visit depending on two explicative variables w_1 and w_2 which will explain the observability of the species. We also fix the intercept and the effects of these two variables: $\gamma_0 = 0.2$, $\gamma_1 = 0.5$ and $\gamma_2 = 0.5$ for determining the detection probability.

```
# Explicative variables for observation process
nobs <- sum(v)
set.seed(seed)
w1 <- rnorm(n=nobs,0,1)
set.seed(2*seed)
w2 <- rnorm(n=nobs,0,1)
W <- cbind(rep(1,nobs),w1,w2)
# Target parameters for observation process
gamma.target <- matrix(c(0.2,0.5,0.5),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.obspt <- X %*% beta.target
theta.obspt <- inv.logit(logit.theta.obspt)
set.seed(seed)
Z <- rbinom(nobspt,1,theta.obspt)

# 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[obspt])
Y <- rbinom(nobs,1,0.8*Z[obspt])

# Spatial entity (raster cells) associated to each observation
cell.obspt <- extract(alt,obs,cellnumbers=TRUE)[,"cells"]
cell.obs <- cell.obspt[obspt]
# cell.obs <- as.numeric(as.factor(cell.obs)) # from 1 to ncell, not necessary

# Data-sets
data.obs <- data.frame(Y,w1,w2,cell=cell.obs)
data.suit <- data.frame(alt=X[,2],alt2=X[,3])
```

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
                                 presences=data.obs$Y,
                                 #observability=~w1+w2,
                                 observability=~1,
                                 spatial.entity=data.obs$cell,
                                 data.observability=data.obs,
                                 # Habitat
                                 suitability=~alt+alt2,
                                 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=0)
```

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)   -0.219 0.195  0.00617         0.0200
## beta.alt            0.151 0.182  0.00575         0.0166
## beta.alt2          -0.122 0.120  0.00378         0.0156
## gamma.(Intercept)   1.918 0.285  0.00901         0.0185
## Deviance          364.457 2.656  0.08399         0.2087
##
## 2. Quantiles for each variable:
##
##                      2.5%     25%     50%      75%    97.5%
## beta.(Intercept)   -0.562  -0.361  -0.213  -0.0661   0.1284
## beta.alt             -0.236   0.023   0.153   0.2689   0.4740
## beta.alt2            -0.410  -0.196  -0.111  -0.0249   0.0621
## gamma.(Intercept)     1.348   1.735   1.937   2.1174   2.4523
## Deviance            360.818 362.343 364.009 365.8364 371.0877
```

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

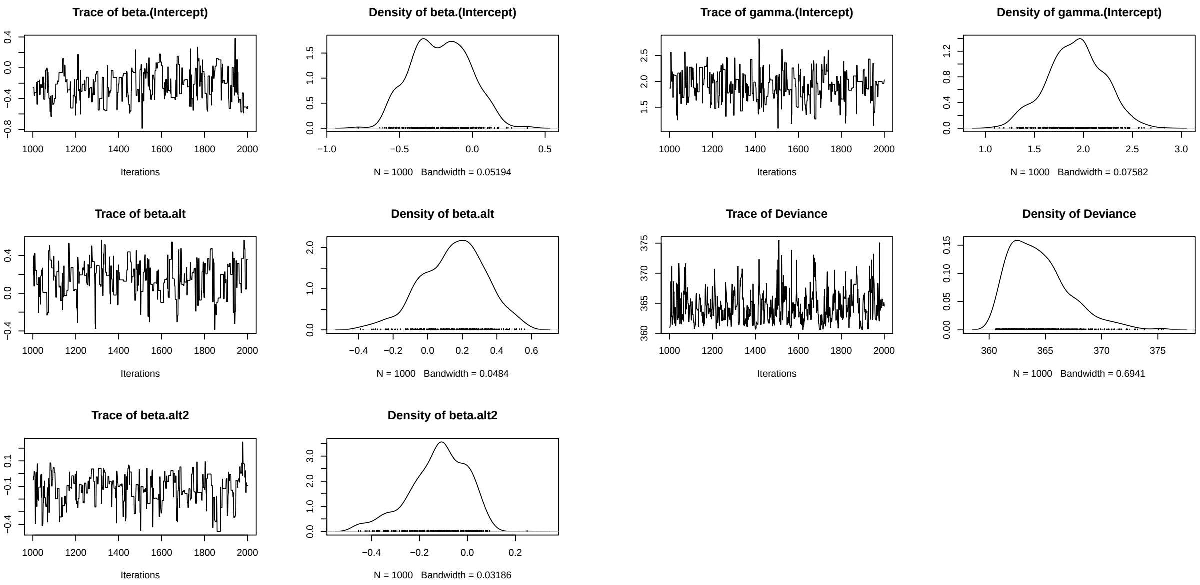

Figure 7: **Trace and density estimate for each variable of the MCMC**.

```
# Create a raster for predictions
theta.pred.mean <- raster(theta)
# Attribute predicted values to raster cells
theta.pred.mean[] <- mod.hSDM.siteocc$theta.pred
# Plot the predicted probability of presence
plot(theta.pred.mean,col=rev(heat.colors(255)))
```

```
# Comparing predictions to initial values
plot(theta[],theta.pred.mean[],cex.lab=1.4)
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.

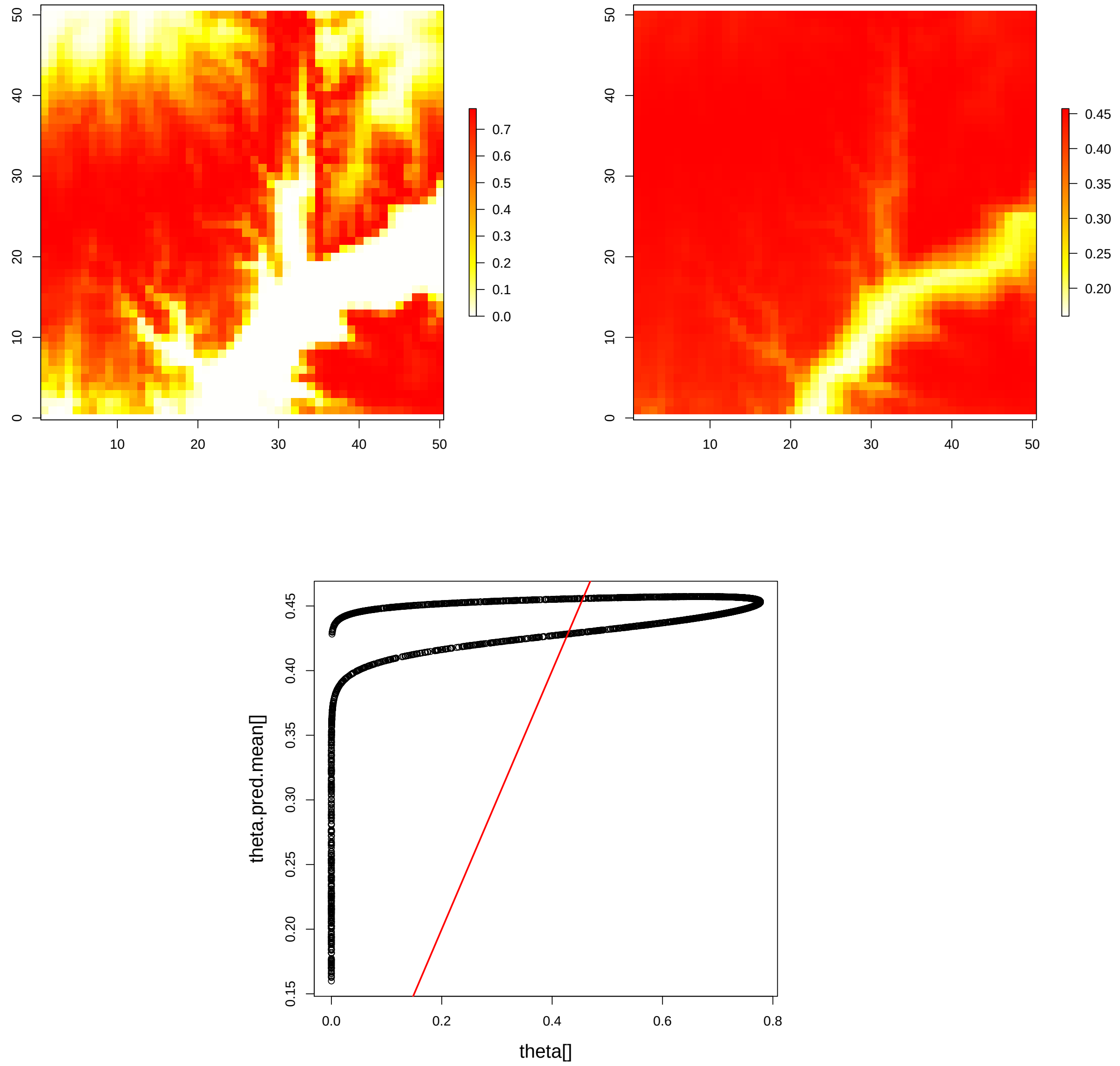

Figure 8: **Comparing predicted probability of presence with initial probabilities**

```
#== glm results for comparison
data.obs.glm <- data.frame(Y,alt=X[obspt,2],alt2=X[obspt,3])
mod.glm <- glm(Y~alt+alt2,family="binomial",data=data.obs.glm)
summary(mod.glm)

##
## Call:
## glm(formula = Y ~ alt + alt2, family = "binomial", data = data.obs.glm)
```

```
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.146  -1.046  -0.579   1.234   1.818
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.0760     0.1372   -0.55     0.58
## alt           0.0544     0.1943    0.28     0.78
## alt2         -1.0477     0.2186   -4.79  1.6e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 534.91  on 416  degrees of freedom
## Residual deviance: 480.33  on 414  degrees of freedom
## AIC: 486.3
##
## Number of Fisher Scoring iterations: 6
```

```
# 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,col=rev(heat.colors(255)))
```

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

2.3 Binomial iCAR model

2.4 Site-occupancy iCAR model

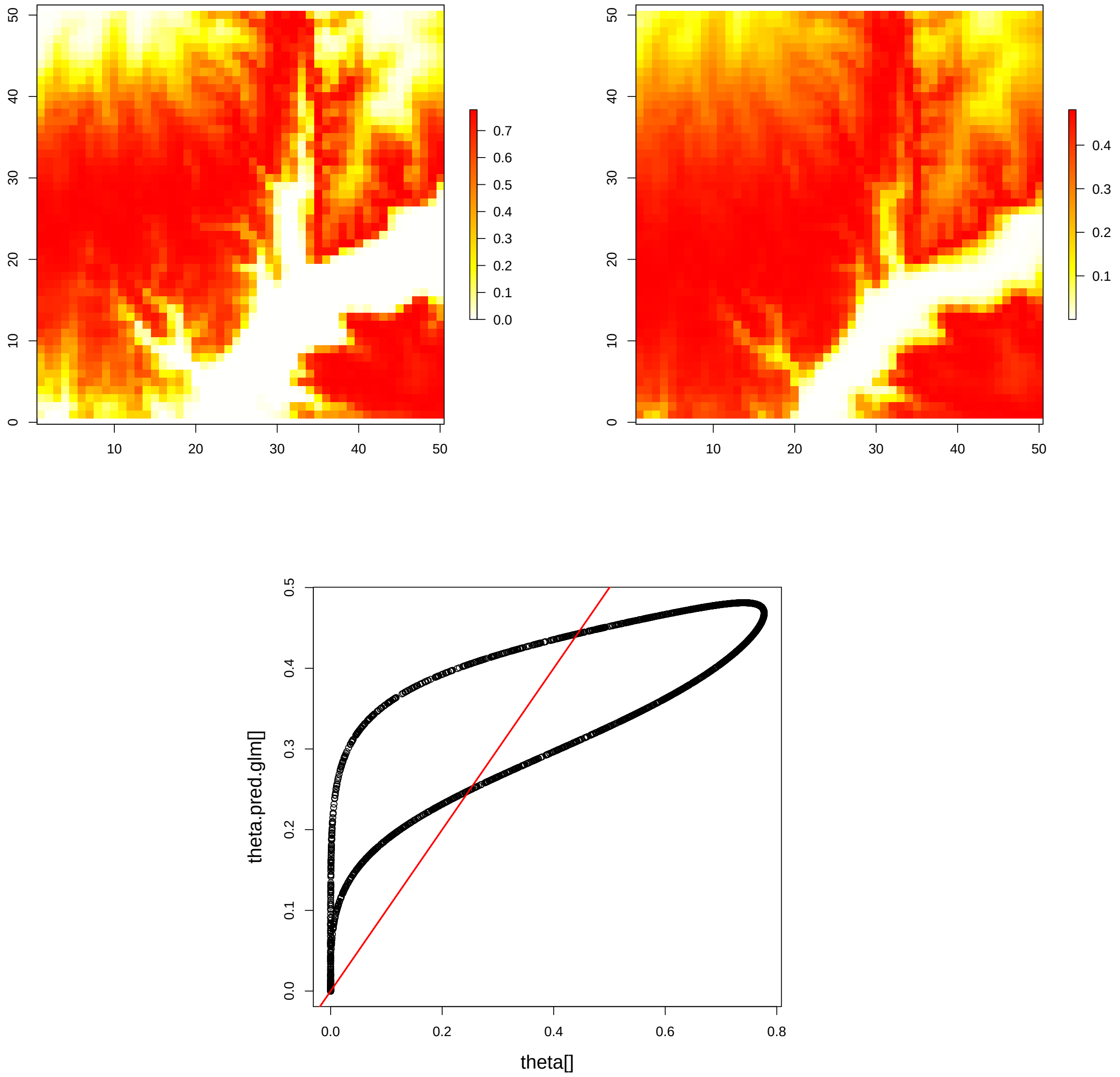

Figure 9: **Comparing predicted probability of presence using glm with initial probabilities**

3 Acknowledgements

References

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.

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

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.

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.