

# Package ‘hSDM’

January 13, 2014

**Type** Package

**Title** hierarchical Bayesian species distribution models

**Version** 1.2

**Date** 2013-10-15

**Depends** coda

**Author** Ghislain Vieilledent, Andrew M. Latimer, Alan E. Gelfand, Cory Merow, Adam M. Wilson, Frederic Mortier and John A. Silander Jr.

**Maintainer** Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

**Description** hSDM is an R package for estimating parameters of hierarchical Bayesian species distribution models. Such models allow interpreting the observations (occurrence and abundance of a species) as a result of several hierarchical processes including ecological processes (habitat suitability, spatial dependence and anthropogenic disturbance) and observation processes (species detectability). Hierarchical species distribution models are essential for accurately characterizing the environmental response of species, predicting their probability of occurrence, and assessing uncertainty in the model results.

**License** GPL-3 | file LICENSE

**URL** <http://hSDM.sf.net>

**LazyLoad** yes

## R topics documented:

hSDM-package	2
hSDM.binomial	2
hSDM.binomial.iCAR	5
hSDM.hierarchical.binomial	10
hSDM.hierarchical.poisson	15
hSDM.Nmixture	20
hSDM.poisson	28
hSDM.poisson.iCAR	31
hSDM.site.occupancy	35
logit	43

**Index****45**

hSDM-package

*hierarchical Bayesian species distribution models***Description**

hSDM is an R package for estimating parameters of hierarchical Bayesian species distribution models. Such models allow interpreting the observations (occurrence and abundance of a species) as a result of several hierarchical processes including ecological processes (habitat suitability, spatial dependence and anthropogenic disturbance) and observation processes (species detectability). Hierarchical species distribution models are essential for accurately characterizing the environmental response of species, predicting their probability of occurrence, and assessing uncertainty in the model results.

**Details**

Package: hSDM  
 Type: Package  
 Version: 1.2  
 Date: 2013-10-15  
 License: GPL-3  
 LazyLoad: yes

**Author(s)**

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>  
 Andrew M. Latimer <amlatimer@ucdavis.edu>  
 Alan E. Gelfand <alan@stat.duke.edu>  
 Cory Merow <cory.merow@gmail.com>  
 Adam M. Wilson <adam.wilson@yale.edu>  
 Frederic Mortier <frederic.mortier@cirad.fr>  
 John A. Silander Jr. <john.silander@uconn.edu>

hSDM.binomial

*The hSDM.binomial function performs a Binomial logistic regression model in a Bayesian framework.*

**Description**

The hSDM.binomial function calls a Gibbs sampler written in C code which uses a Metropolis algorithm to estimate the conditional posterior distribution of model's parameters.

**Usage**

```
hSDM.binomial(presences, trials, suitability, data, burnin =
5000, mcmc = 10000, thin = 10, beta.start, mubeta = 0, Vbeta = 1e+06,
seed = 1234, verbose = 1)
```

**Arguments**

presences	A vector indicating the number of successes (or presences) for each observation.
trials	A vector indicating the number of trials for each observation. $t_n$ should be superior or equal to $y_n$ , the number of successes for observation $n$ . If $t_n = 0$ , then $y_n = 0$ .
suitability	A one-sided formula of the form '~x1+...+xp' with p terms specifying the explicative variables for the suitability process of the model.
data	A data frame containing the model's explicative variables.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
beta.start	Starting values for beta parameters of the suitability process. If beta.start takes a scalar value, then that value will serve for all of the betas.
mubeta	Means of the priors for the $\beta$ parameters of the suitability process. mubeta must be either a scalar or a p-length vector. If mubeta takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value is set to 0 for an uninformative prior.
Vbeta	Variances of the Normal priors for the $\beta$ parameters of the suitability process. Vbeta must be either a scalar or a p-length vector. If Vbeta takes a scalar value, then that value will serve as the prior variance for all of the betas. The default variance is large and set to 1.0E6 for an uninformative flat prior.
seed	The seed for the random number generator. Default to 1234.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.

**Value**

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance $D$ , with $D = -2 \log(\prod_i P(y_i, n_i   \beta))$ , is also provided.
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

prob.p.pred      Predictive posterior mean of the probability associated to the suitability process for each spatial cell.

### Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

### References

Latimer, A. M.; Wu, S. S.; Gelfand, A. E. and Silander, J. A. (2006) Building statistical models to analyze species distributions. *Ecological Applications*, 16, 33-50.

Gelfand, A. E.; Schmidt, A. M.; Wu, S.; Silander, J. A.; Latimer, A. and Rebelo, A. G. (2005) Modelling species diversity through species level hierarchical modelling. *Applied Statistics*, 54, 1-20.

### See Also

[plot.mcmc](#), [summary.mcmc](#)

### Examples

```
## Not run:

#####
# hSDM.binomial()
# Example with simulated data
#####

=====
#== Preamble
library(hSDM)

=====
#== Data simulation

# Set seed for repeatability
set.seed(1234)

# Constants
nobs <- 1000
trials <- rpois(nobs,5) # Number of trial associated to each observation

# Covariates for "suitability" process
X1 <- rnorm(n=nobs,0,1)
X2 <- rnorm(n=nobs,0,1)
X <- cbind(rep(1,nobs),X1,X2)

# Target parameters
beta.target <- matrix(c(0.2,0.1,0.1),ncol=1) # fixed effects
```

```

#== Simulating latent variables

# Suitability
logit.theta <- X %*% beta.target
theta <- inv.logit(logit.theta)

#== Simulating response variable
Y <- rbinom(n=nobs,size=trials,prob=theta)

#== Data-set
Data <- data.frame(Y,trials,X1,X2)
str(Data)
par(mfrow=c(2,1))
plot(Data$X1,logit.theta)
plot(Data$X2,logit.theta)

#=====
#== Statistical modelling

#== glm resolution to compare
mod.glm <- glm(cbind(Y,trials-Y)~X1+X2,family="binomial",data=Data)
summary(mod.glm)
beta.hat.glm <- coef(mod.glm)

#== hSDM
mod.hSDM.binomial <- hSDM.binomial(presences=Data$Y,
                                   trials=Data$trials,
                                   suitability=~X1+X2,
                                   data=Data, burnin=1000,
                                   mcmc=1000, thin=1,
                                   beta.start=beta.hat.glm,
                                   mubeta=0, Vbeta=1.0E6,
                                   seed=1234, verbose=1)

#=====
#== Outputs
summary(mod.hSDM.binomial$mcmc)
pdf(file="Posteriors_hSDM.binomial.pdf")
plot(mod.hSDM.binomial$mcmc)
dev.off()
summary(mod.hSDM.binomial$prob.p.pred)

## End(Not run)

```

hSDM.binomial.iCAR

*The hSDM.binomial.iCAR function performs a Binomial logistic regression model in a hierarchical Bayesian framework. The suitability process includes a spatial correlation process. The spatial correlation is modelled using an intrinsic CAR model.*

**Description**

The `hSDM.binomial.iCAR` function calls a Gibbs sampler written in C code which uses a Metropolis algorithm to estimate the conditional posterior distribution of model's parameters.

**Usage**

```
hSDM.binomial.iCAR(presences, trials, suitability, cells,
n.neighbors, neighbors, data, burnin = 5000, mcmc = 10000, thin = 10,
beta.start, Vrho.start, mubeta = 0, Vbeta = 1e+06, priorVrho =
"1/Gamma", shape = 0.5, rate = 0.0005, Vrho.max = 1000, seed = 1234,
verbose = 1, save.rho = 0)
```

**Arguments**

<code>presences</code>	A vector indicating the number of successes (or presences) for each observation.
<code>trials</code>	A vector indicating the number of trials for each observation. $t_n$ should be superior or equal to $y_n$ , the number of successes for observation $n$ . If $t_n = 0$ , then $y_n = 0$ .
<code>suitability</code>	A one-sided formula of the form ' $\sim x_1 + \dots + x_p$ ' with $p$ terms specifying the explicative variables for the suitability process of the model.
<code>cells</code>	A vector indicating the spatial cell identifier (from 1 to total number of cell) for each observation. Several observations can occur in one spatial cell.
<code>n.neighbors</code>	A vector of integers indicating the number of neighbors (adjacent cells) of each spatial cell.
<code>neighbors</code>	A vector of integers indicating the neighbors (adjacent cells) of each spatial cell. Must be of the form $c(\text{neighbors of cell 1, neighbors of cell 2, } \dots, \text{neighbors of the last cell})$ . Length of the <code>neighbors</code> vector should be equal to <code>sum(data\$num)</code> .
<code>data</code>	A data frame containing the model's explicative variables.
<code>burnin</code>	The number of burnin iterations for the sampler.
<code>mcmc</code>	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to <code>burnin+mcmc</code> . <code>burnin+mcmc</code> must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
<code>thin</code>	The thinning interval used in the simulation. The number of <code>mcmc</code> iterations must be divisible by this value.
<code>beta.start</code>	Starting values for beta parameters.
<code>Vrho.start</code>	Positive scalar indicating the starting value for the variance of the spatial random effects.
<code>mubeta</code>	Means of the priors for the $\beta$ parameters of the suitability process. <code>mubeta</code> must be either a scalar or a $p$ -length vector. If <code>mubeta</code> takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value is set to 0 for an uninformative prior.
<code>Vbeta</code>	Variances of the Normal priors for the $\beta$ parameters of the suitability process. <code>Vbeta</code> must be either a scalar or a $p$ -length vector. If <code>Vbeta</code> takes a scalar value, then that value will serve as the prior variance for all of the betas. The default variance is large and set to $1.0E6$ for an uninformative flat prior.

priorVrho	Type of prior for the variance of the spatial random effects. Can be set to a fixed positive scalar, or to an inverse-gamma distribution ("1/Gamma") with parameters shape and rate, or to a uniform distribution ("Uniform") on the interval [0,Vrho.max]. Default to "1/Gamma".
shape	The shape parameter for the Gamma prior on the precision of the spatial random effects. Default value is shape=0.05 for uninformative prior.
rate	The rate (1/scale) parameter for the Gamma prior on the precision of the spatial random effects. Default value is rate=0.0005 for uninformative prior.
Vrho.max	Upper bound for the uniform prior of the spatial random effect variance. Default to 1000.
seed	The seed for the random number generator. Default to 1234.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.
save.rho	A switch (0,1) which determines whether or not the sampled values for rhos are saved. Default is 0: the posterior mean is computed and returned in the rho.pred vector. Be careful, setting save.rho to 1 might require a large amount of memory.

**Value**

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance $D$ , with $D = -2 \log(\prod_i P(y_i, n_i   \beta, \rho_i))$ , is also provided.
rho.pred	If save.rho is set to 0 (default), rho.pred is the predictive posterior mean of the spatial random effect associated to each spatial cell. If save.rho is set to 1, rho.pred is an mcmc object with sampled values for each spatial random effect associated to each spatial cell.
prob.p.pred	Predictive posterior mean of the probability associated to the suitability process for each spatial cell.

**Author(s)**

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

**References**

- Latimer, A. M.; Wu, S. S.; Gelfand, A. E. and Silander, J. A. (2006) Building statistical models to analyze species distributions. *Ecological Applications*, 16, 33-50.
- Gelfand, A. E.; Schmidt, A. M.; Wu, S.; Silander, J. A.; Latimer, A. and Rebelo, A. G. (2005) Modelling species diversity through species level hierarchical modelling. *Applied Statistics*, 54, 1-20.

**See Also**

[plot.mcmc](#), [summary.mcmc](#)

## Examples

```
## Not run:

#####
# hSDM.binomial.iCAR()
# Example with simulated data
#####

=====
#== Preamble
library(mvtnorm)
library(lme4) # To compare with nonspatial random effects model
library(hSDM)

=====
#== Data simulation

# Set seed for repeatability
set.seed(1234)

# Constants
ncell <- 150 # Number of cells
nobs <- 10*ncell # Number of observation for the *binomial* random variable
trials <- rpois(nobs,5) # Number of trial associated to each observation
cell <- rep(c(1:ncell),each=nobs/ncell)

# Covariates for "suitability" process
X1 <- rnorm(n=nobs,0,1)
X2 <- rnorm(n=nobs,0,1)
X <- cbind(rep(1,nobs),X1,X2)

# Target parameters
beta.target <- matrix(c(0.2,0.1,0.1),ncol=1) # fixed effects
Vrho.target <- 10 # Spatial Variance

# Generate symmetric adjacency matrix, A
A <- matrix(0,ncell,ncell)
A[upper.tri(A,diag=F)] <- rbinom(ncell*(ncell-1)/2,1,.05)
A <- A+t(A)
n.neighbors <- apply(A,1,sum)
f.adjacent <- function (x) {
  which(x==1)
}
adj <- unlist(apply(A,1,f.adjacent))

# Spatial effects
d <- 1 # Spatial dependence parameter = 1 for intrinsic CAR
Q <- diag(n.neighbors)-d*A + diag(.0001,ncell) # Add small constant to make Q non-singular
covrho <- Vrho.target*solve(Q) # Covariance of rhos
rho <- c(rmvnorm(1,sigma=covrho)) # Spatial Random Effects
rho <- rho-mean(rho) # Centering rhos on zero
```



```

#== Simulating latent variables

# Suitability
logit.theta <- vector()
for (n in 1:nobs) {
  logit.theta[n] <- X[n,]%*%beta.target+rho[cell[n]]
}
theta <- inv.logit(logit.theta)

#== Simulating response variable
Y <- rbinom(nobs,trials,theta)

#== Data-set
Data <- data.frame(Y,trials,cell,X1,X2)
str(Data)

#=====
#== Statistical modelling

#== glm resolution to compare
mod.glm <- glm(cbind(Y,trials-Y)~X1+X2,data=Data,family="binomial")
summary(mod.glm)

#== glmm (mixed models) resolution to compare
#== cell random effects without spatial correlation
mod.glmm <- lmer(cbind(Y,trials-Y)~X1+X2+(1|cell),data=Data,family="binomial") # Non-spatial random effect fit
summary(mod.glmm)
beta.hat.glmm <- fixef(mod.glmm)

#== hSDM
mod.hSDM.binomial.iCAR <- hSDM.binomial.iCAR(presences=Data$Y,
                                              trials=Data$trials,
                                              suitability=~X1+X2,
                                              cells=Data$cell,
                                              n.neighbors=n.neighbors,
                                              neighbors=adj,
                                              data=Data, burnin=1000,
                                              mcmc=1000, thin=1,
                                              beta.start=beta.hat.glmm,
                                              Vrho.start=10,
                                              priorVrho="1/Gamma",
                                              #priorVrho="Uniform",
                                              #priorVrho=10,
                                              mubeta=0, Vbeta=1.0E6,
                                              shape=0.5, rate=0.0005,
                                              Vrho.max=1000,
                                              seed=1234, verbose=1, save.rho=0)

#=====
#== Outputs
summary(mod.hSDM.binomial.iCAR$mcmc)
pdf(file="Posteriors_hSDM.binomial.iCAR.pdf")

```

```

plot(mod.hSDM.binomial.iCAR$mcmc)
dev.off()
summary(mod.hSDM.binomial.iCAR$prob.p.pred)
summary(mod.hSDM.binomial.iCAR$rho.pred)

## End(Not run)

```

---

#### hSDM.hierarchical.binomial

*The hSDM.hierarchical.binomial function can be used to model species distribution including different processes in a hierarchical Bayesian framework: (i) a Bernoulli suitability process (referring to environmental suitability) which takes into account the spatial dependence of the observations, (ii) an alteration process (referring to anthropogenic disturbances), and (iii) a Binomial observability process (referring to various ecological and methodological issues explaining the species presence). The model reduces to a ZIB (Zero-Inflated Binomial) model with spatial dependence if the alteration is set to zero.*

---

### Description

The hSDM.hierarchical.binomial function calls a Gibbs sampler written in C code which uses a Metropolis algorithm to estimate the conditional posterior distribution of hierarchical model's parameters.

### Usage

```

hSDM.hierarchical.binomial(presences, trials, suitability, cells,
n.neighbors, neighbors, alteration, observability, data, burnin = 5000,
mcmc = 10000, thin = 10, beta.start, gamma.start, Vrho.start, mubeta =
0, Vbeta = 1e+06, mugamma = 0, Vgamma = 1e+06, priorVrho = "1/Gamma",
shape = 0.5, rate = 0.0005, Vrho.max=1000, seed = 1234, verbose = 1,
save.rho = 0)

```

### Arguments

presences	A vector indicating the number of successes (or presences) for each observation.
trials	A vector indicating the number of trials for each observation. $t_n$ should be superior or equal to $y_n$ , the number of successes for observation $n$ . If $t_n = 0$ , then $y_n = 0$ .
suitability	A one-sided formula of the form '~x1+...+xp' with p terms specifying the explicative variables for the suitability process of the model.
cells	A vector indicating the spatial cell identifier (from 1 to total number of cell) for each observation. Several observations can occur in one spatial cell.

n.neighbors	A vector of integers that indicates the number of neighbors (adjacent cells) of each spatial cell.
neighbors	A vector of integers indicating the neighbors (adjacent cells) of each spatial cell. Must be of the form c(neighbors of cell 1, neighbors of cell 2, ... , neighbors of the last cell). Length of the neighbors vector should be equal to sum(data\$num).
alteration	A vector indicating the proportion of area in the spatial cell which is transformed (by anthropogenic activities for example) for each observation. Must be between 0 and 1.
observability	A one-sided formula of the form ' $\sim x_1 + \dots + x_q$ ' with q terms specifying the explicative variables for the observability process of the model.
data	A data frame containing the model's variables.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
beta.start	Starting values for beta parameters.
gamma.start	Starting values for gamma parameters.
Vrho.start	Positive scalar indicating the starting value for the variance of the spatial random effects.
mubeta	Means of the priors for the $\beta$ parameters of the suitability process. mubeta must be either a scalar or a p-length vector. If mubeta takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value is set to 0 for an uninformative prior.
Vbeta	Variances of the Normal priors for the $\beta$ parameters of the suitability process. Vbeta must be either a scalar or a p-length vector. If Vbeta takes a scalar value, then that value will serve as the prior variance for all of the betas. The default variance is large and set to 1.0E6 for an uninformative flat prior.
mugamma	Means of the Normal priors for the $\gamma$ parameters of the observability process. mugamma must be either a scalar or a p-length vector. If mugamma takes a scalar value, then that value will serve as the prior mean for all of the gammas. The default value is set to 0 for an uninformative prior.
Vgamma	Variances of the Normal priors for the $\gamma$ parameters of the observability process. Vgamma must be either a scalar or a p-length vector. If Vgamma takes a scalar value, then that value will serve as the prior variance for all of the gammas. The default variance is large and set to 1.0E6 for an uninformative flat prior.
priorVrho	Type of prior for the variance of the spatial random effects. Can be set to a fixed positive scalar, or to an inverse-gamma distribution ("1/Gamma") with parameters shape and rate, or to a uniform distribution ("Uniform") on the interval [0,Vrho.max]. Default to "1/Gamma".
shape	The shape parameter for the Gamma prior on the precision of the spatial random effects. Default value is shape=0.05 for uninformative prior.

rate	The rate (1/scale) parameter for the Gamma prior on the precision of the spatial random effects. Default value is rate=0.0005 for uninformative prior.
Vrho.max	Upper bound for the uniform prior of the spatial random effect variance. Default to 1000.
seed	The seed for the random number generator. Default to 1234.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.
save.rho	A switch (0,1) which determines whether or not the sampled values for rhos are saved. Default is 0: the posterior mean is computed and returned in the rho.pred vector. Be careful, setting save.rho to 1 might require a large amount of memory.

### Value

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance $D$ , with $D = -2 \log(\prod_i P(y_i, n_i   u_i, \beta, \gamma, \rho_i))$ , is also provided.
rho.pred	If save.rho is set to 0 (default), rho.pred is the predictive posterior mean of the spatial random effect associated to each spatial cell. If save.rho is set to 1, rho.pred is an mcmc object with sampled values for each spatial random effect associated to each spatial cell.
prob.p.pred	Predictive posterior mean of the probability associated to the suitability process for each spatial cell.
prob.q.pred	Predictive posterior mean of the probability associated to the observability process for each spatial cell.

### Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

### References

- Latimer, A. M.; Wu, S. S.; Gelfand, A. E. and Silander, J. A. (2006) Building statistical models to analyze species distributions. *Ecological Applications*, 16, 33-50.
- Gelfand, A. E.; Schmidt, A. M.; Wu, S.; Silander, J. A.; Latimer, A. and Rebelo, A. G. (2005) Modelling species diversity through species level hierarchical modelling. *Applied Statistics*, 54, 1-20.
- Gilks, W. R., Best, N. G. and Tan, K. K. C. (1995) Adaptive rejection Metropolis sampling. *Applied Statistics*, 44, 455-472.

### See Also

[plot.mcmc](#), [summary.mcmc](#)

**Examples**

```
## Not run:

#####
# hSDM.hierarchical.binomial()
# Example with simulated data
#####

=====
#== Preamble
library(mvtnorm)
library(hSDM)

=====
#== Data simulation

# Set seed for repeatability
set.seed(1234)

# Constants
ncell <- 150 # Number of cells
nobs <- 100*ncell # Number of observation for the *binomial* random variable
trials <- rpois(nobs,5) # Number of trial associated to each observation
cell <- rep(c(1:ncell),each=nobs/ncell)

# Covariates for "suitability" process
X1 <- rnorm(n=nobs,0,1)
X2 <- rnorm(n=nobs,0,1)
X <- cbind(rep(1,nobs),X1,X2)

# Alteration
U <- runif(n=nobs,min=0,max=1)

# Covariates for "observability" process
W1 <- rnorm(n=nobs,0,1)
W2 <- rnorm(n=nobs,0,1)
W <- cbind(rep(1,nobs),W1,W2)

# Target parameters
beta.target <- matrix(c(0.2,0.1,0.1),ncol=1) # fixed effects
gamma.target <- matrix(c(0.3,0.1,0.1),ncol=1) # fixed effects
Vrho.target <- 10 # Spatial Variance

# Generate symmetric adjacency matrix, A
A <- matrix(0,ncell,ncell)
A[upper.tri(A,diag=F)] <- rbinom(ncell*(ncell-1)/2,1,.05)
A <- A+t(A)
n.neighbors <- apply(A,1,sum)
f.adjacent <- function (x) {
  which(x==1)
}
```

```

adj <- unlist(apply(A,1,f.adjacent))

# Spatial effects
d <- 1 # Spatial dependence parameter = 1 for intrinsic CAR
Q <- diag(n.neighbors)-d*A + diag(.0001,ncell) # Add small constant to make Q non-singular
covrho <- Vrho.target*solve(Q) # Covariance of rhos
rho <- c(rmvnorm(1,sigma=covrho)) # Spatial Random Effects
rho <- rho-mean(rho) # Centering rhos on zero

#== Simulating latent variables

# Suitability
logit.theta.1 <- vector()
for (n in 1:nobs) {
  logit.theta.1[n] <- X[n,]%*%beta.target+rho[cell[n]]
}
theta.1 <- inv.logit(logit.theta.1)
y.1 <- rbinom(nobs,1,theta.1)

# Alteration
u <- rbinom(nobs,1,U)

# Observability
logit.theta.2 <- W%*%gamma.target
theta.2 <- inv.logit(logit.theta.2)
y.2 <- rbinom(nobs,trials,theta.2)

#== Simulating response variable
Y <- y.2*(1-u)*y.1

#== Data-set
Data <- data.frame(Y,trials,cell,X1,X2,W1,W2,U)
str(Data)

#=====
#== Statistical modelling with hSDM

model <- hSDM.hierarchical.binomial(presences=Data$Y,
                                     trials=Data$trials,
                                     suitability=~X1+X2,
                                     cells=Data$cell,
                                     n.neighbors=n.neighbors,
                                     neighbors=adj,
                                     alteration=Data$U,
                                     observability=~W1+W2,
                                     data=Data, burnin=500,
                                     mcmc=500, thin=1,
                                     beta.start=0,
                                     gamma.start=0,
                                     Vrho.start=1,
                                     priorVrho="1/Gamma",
                                     #priorVrho="Uniform",
                                     #priorVrho=10,

```

```

                                mubeta=0, Vbeta=1.0E6,
                                mugamma=0, Vgamma=1.0E6,
                                shape=0.5, rate=0.0005,
                                Vrho.max=1000,
                                seed=1234, verbose=1, save.rho=0)

#####
#== Outputs
summary(model$mcmc)
pdf(file="Posteriors_hSDM.hierarchical.binomial.pdf")
plot(model$mcmc)
dev.off()
summary(model$prob.p.pred)
summary(model$rho.pred)

## End(Not run)

```

---

hSDM.hierarchical.poisson

*The hSDM.hierarchical.poisson function can be used to model species distribution including different processes in a hierarchical Bayesian framework: (i) a Bernoulli suitability process (referring to environmental suitability) which takes into account the spatial dependence of the observations, (ii) an alteration process (referring to anthropogenic disturbances), and (iii) a Poisson observability process (referring to various ecological and methodological issues explaining the species abundance). The model reduces to a ZIP (Zero-Inflated Poisson) model with spatial dependence if the alteration is set to zero.*

---

## Description

The hSDM.hierarchical.poisson function calls a Gibbs sampler written in C code which uses a Metropolis algorithm to estimate the conditional posterior distribution of model's parameters.

## Usage

```

hSDM.hierarchical.poisson(counts, visits, suitability, cells,
n.neighbors, neighbors, alteration, observability, data, burnin = 5000,
mcmc = 10000, thin = 10, beta.start, gamma.start, Vrho.start, mubeta =
0, Vbeta = 1e+06, mugamma = 0, Vgamma = 1e+06, priorVrho = "1/Gamma",
shape = 0.5, rate = 0.0005, Vrho.max=1000, seed = 1234, verbose = 1,
save.rho = 0)

```

## Arguments

counts                      A vector indicating the count (or abundance) for each observation.

<code>visits</code>	A vector indicating if the observation point $n$ has been visited ( <code>visits[n]=1</code> ) or not ( <code>visits[n]=0</code> ). If <code>visits[n]=0</code> , then <code>counts[n]=0</code> .
<code>suitability</code>	A one-sided formula of the form ' $\sim x_1 + \dots + x_p$ ' with $p$ terms specifying the explicative variables for the suitability process of the model.
<code>cells</code>	A vector indicating the spatial cell identifier (from 1 to total number of cell) for each observation. Several observations can occur in one spatial cell.
<code>n.neighbors</code>	A vector of integers that indicates the number of neighbors (adjacent cells) of each spatial cell.
<code>neighbors</code>	A vector of integers indicating the neighbors (adjacent cells) of each spatial cell. Must be of the form <code>c(neighbors of cell 1, neighbors of cell 2, ... , neighbors of the last cell)</code> . Length of the <code>neighbors</code> vector should be equal to <code>sum(data\$num)</code> .
<code>alteration</code>	A vector indicating the proportion of area in the spatial cell which is transformed (by anthropogenic activities for example) for each observation. Must be between 0 and 1.
<code>observability</code>	A one-sided formula of the form ' $\sim x_1 + \dots + x_q$ ' with $q$ terms specifying the explicative variables for the observability process of the model.
<code>data</code>	A data frame containing the model's explicative variables.
<code>burnin</code>	The number of burnin iterations for the sampler.
<code>mcmc</code>	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to <code>burnin+mcmc</code> . <code>burnin+mcmc</code> must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
<code>thin</code>	The thinning interval used in the simulation. The number of <code>mcmc</code> iterations must be divisible by this value.
<code>beta.start</code>	Starting values for beta parameters.
<code>gamma.start</code>	Starting values for gamma parameters.
<code>Vrho.start</code>	Positive scalar indicating the starting value for the variance of the spatial random effects.
<code>mubeta</code>	Means of the priors for the $\beta$ parameters of the suitability process. <code>mubeta</code> must be either a scalar or a $p$ -length vector. If <code>mubeta</code> takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value is set to 0 for an uninformative prior.
<code>Vbeta</code>	Variances of the Normal priors for the $\beta$ parameters of the suitability process. <code>Vbeta</code> must be either a scalar or a $p$ -length vector. If <code>Vbeta</code> takes a scalar value, then that value will serve as the prior variance for all of the betas. The default variance is large and set to $1.0E6$ for an uninformative flat prior.
<code>mugamma</code>	Means of the Normal priors for the $\gamma$ parameters of the observability process. <code>mugamma</code> must be either a scalar or a $p$ -length vector. If <code>mugamma</code> takes a scalar value, then that value will serve as the prior mean for all of the gammas. The default value is set to 0 for an uninformative prior.
<code>Vgamma</code>	Variances of the Normal priors for the $\gamma$ parameters of the observability process. <code>Vgamma</code> must be either a scalar or a $p$ -length vector. If <code>Vgamma</code> takes a scalar value, then that value will serve as the prior variance for all of the gammas. The default variance is large and set to $1.0E6$ for an uninformative flat prior.



priorVrho	Type of prior for the variance of the spatial random effects. Can be set to a fixed positive scalar, or to an inverse-gamma distribution ("1/Gamma") with parameters shape and rate, or to a uniform distribution ("Uniform") on the interval [0,Vrho.max]. Default to "1/Gamma".
shape	The shape parameter for the Gamma prior on the precision of the spatial random effects. Default value is shape=0.05 for uninformative prior.
rate	The rate (1/scale) parameter for the Gamma prior on the precision of the spatial random effects. Default value is rate=0.0005 for uninformative prior.
Vrho.max	Upper bound for the uniform prior of the spatial random effect variance. Default to 1000.
seed	The seed for the random number generator. Default to 1234.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.
save.rho	A switch (0,1) which determines whether or not the sampled values for rhos are saved. Default is 0: the posterior mean is computed and returned in the rho.pred vector. Be careful, setting save.rho to 1 might require a large amount of memory.

**Value**

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance $D$ , with $D = -2 \log(\prod_i P(y_i, n_i   u_i, \beta, \gamma, \rho_i))$ , is also provided.
rho.pred	If save.rho is set to 0 (default), rho.pred is the predictive posterior mean of the spatial random effect associated to each spatial cell. If save.rho is set to 1, rho.pred is an mcmc object with sampled values for each spatial random effect associated to each spatial cell.
prob.p.pred	Predictive posterior mean of the probability associated to the suitability process for each spatial cell.
prob.q.pred	Predictive posterior mean of the probability associated to the observability process for each spatial cell.

**Author(s)**

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

**References**

- Latimer, A. M.; Wu, S. S.; Gelfand, A. E. and Silander, J. A. (2006) Building statistical models to analyze species distributions. *Ecological Applications*, 16, 33-50.
- Gelfand, A. E.; Schmidt, A. M.; Wu, S.; Silander, J. A.; Latimer, A. and Rebelo, A. G. (2005) Modelling species diversity through species level hierarchical modelling. *Applied Statistics*, 54, 1-20.
- Gilks, W. R., Best, N. G. and Tan, K. K. C. (1995) Adaptive rejection Metropolis sampling. *Applied Statistics*, 44, 455-472.

**See Also**

[plot.mcmc](#), [summary.mcmc](#)

**Examples**

```
## Not run:

#####
# hSDM.hierarchical.poisson()
# Example with simulated data
#####

=====
#== Preamble
library(mvtnorm)
library(hSDM)

=====
#== Data simulation

# Set seed for repeatability
set.seed(1234)

# Constants
ncell <- 150 # Number of cells
nobs <- 100*ncell # Number of observation for the *binomial* random variable
cell <- rep(c(1:ncell),each=nobs/ncell)

# Covariates for "suitability" process
X1 <- rnorm(n=nobs,0,1)
X2 <- rnorm(n=nobs,0,1)
X <- cbind(rep(1,nobs),X1,X2)

# Alteration
U <- runif(n=nobs,min=0,max=1)

# Covariates for "observability" process
W1 <- rnorm(n=nobs,0,1)
W2 <- rnorm(n=nobs,0,1)
W <- cbind(rep(1,nobs),W1,W2)

# Target parameters
beta.target <- matrix(c(0.2,0.1,0.1),ncol=1) # fixed effects
gamma.target <- matrix(c(0.3,0.1,0.1),ncol=1) # fixed effects
Vrho.target <- 10 # Spatial Variance

# Generate symmetric adjacency matrix, A
A <- matrix(0,ncell,ncell)
A[upper.tri(A,diag=F)] <- rbinom(ncell*(ncell-1)/2,1,.05)
A <- A+t(A)
n.neighbors <- apply(A,1,sum)
```

```

f.adjacent <- function (x) {
  which(x==1)
}
adj <- unlist(apply(A,1,f.adjacent))

# Spatial effects, non-intrinsic CAR with rho ~ 1
d <- 1 # Spatial dependence parameter = 1 for intrinsic CAR
Q <- diag(n.neighbors)-d*A + diag(.0001,ncell) # Add small constant to make Q non-singular
covrho <- Vrho.target*solve(Q) # Covariance of rhos
rho <- c(rmvnorm(1,sigma=covrho)) # Spatial Random Effects
rho <- rho-mean(rho) # Centering rhos on zero

#== Simulating latent variables

# Suitability
logit.theta.1 <- vector()
for (n in 1:nobs) {
  logit.theta.1[n] <- X[n,]%*%beta.target+rho[cell[n]]
}
theta.1 <- inv.logit(logit.theta.1)
y.1 <- rbinom(nobs,1,theta.1)

# Alteration
u <- rbinom(nobs,1,U)

# Observability
log.theta.2 <- W%*%gamma.target
theta.2 <- exp(log.theta.2)
y.2 <- rpois(nobs,theta.2)

#== Simulating response variable
Y <- y.2*(1-u)*y.1
V <- rep(1,nobs)

#== Data-set
Data <- data.frame(Y,V,cell,X1,X2,W1,W2,U)
str(Data)

#=====
#== Statistical modelling with hSDM

model <- hSDM.hierarchical.poisson(counts=Data$Y,
                                   visits=Data$V,
                                   suitability=~X1+X2,
                                   cells=Data$cell,
                                   n.neighbors=n.neighbors,
                                   neighbors=adj,
                                   alteration=Data$U,
                                   observability=~W1+W2,
                                   data=Data, burnin=500,
                                   mcmc=500, thin=1,
                                   beta.start=0,
                                   gamma.start=0,

```

```

Vrho.start=1,
priorVrho="1/Gamma",
#priorVrho="Uniform",
#priorVrho=10,
mubeta=0, Vbeta=1.0E6,
mugamma=0, Vgamma=1.0E6,
shape=0.5, rate=0.0005,
Vrho.max=1000,
seed=1234, verbose=1, save.rho=0)

#=====
#== Outputs
summary(model$mcmc)
pdf(file="Posteriors_hSDM.hierarchical.binomial.pdf")
plot(model$mcmc)
dev.off()
summary(model$prob.p.pred)
summary(model$rho.pred)

## End(Not run)

```

---

hSDM.Nmixture

*The hSDM.Nmixture function can be used to model species distribution including different processes in a hierarchical Bayesian framework: a Poisson suitability process (referring to environmental suitability) which takes into account the spatial dependence of the observations, and a Binomial observability process (referring to various ecological and methodological issues explaining the species detection).*

---

## Description

The hSDM.Nmixture function calls a Gibbs sampler written in C code which uses a Metropolis algorithm to estimate the conditional posterior distribution of hierarchical model's parameters.

## Usage

```

hSDM.Nmixture(# Observations
               counts, observability, spatial.entity, data.observability,
               # Habitat
               suitability, data.suitability,
               # Spatial structure
               n.neighbors, neighbors,
               # Predictions
               suitability.pred = NULL, spatial.entity.pred = NULL,
               # Chains
               burnin = 5000, mcmc = 10000, thin = 10,

```

```

# Starting values
beta.start,
gamma.start,
Vrho.start,
# Priors
mubeta = 0, Vbeta = 1.0E6,
mugamma = 0, Vgamma = 1.0E6,
priorVrho = "1/Gamma",
shape = 0.5, rate = 0.0005,
Vrho.max = 1000,
# Various
seed = 1234, verbose = 1,
save.rho = 0, save.p = 0, save.N = 0)

```

### Arguments

counts	A vector indicating the count (or abundance) for each observation.
observability	A one-sided formula of the form $\sim w_1 + \dots + w_q$ with $q$ terms specifying the explicative variables for the observability process.
spatial.entity	A vector indicating the spatial entity identifier (from one to the total number of entities) for each observation. Several observations can occur in one spatial entity. A spatial entity can be a raster cell for example.
data.observability	A data frame containing the model's variables for the observability process.
suitability	A one-sided formula of the form $\sim x_1 + \dots + x_p$ with $p$ terms specifying the explicative variables for the suitability process.
data.suitability	A data frame containing the model's variables for the suitability process.
n.neighbors	A vector of integers that indicates the number of neighbors (adjacent entities) of each spatial entity. <code>length(n.neighbors)</code> indicates the total number of spatial entities.
neighbors	A vector of integers indicating the neighbors (adjacent entities) of each spatial entity. Must be of the form <code>c(neighbors of entity 1, neighbors of entity 2, ... , neighbors of the last entity)</code> . Length of the neighbors vector should be equal to <code>sum(n.neighbors)</code> .
suitability.pred	An optional data frame in which to look for variables with which to predict. If NULL, the observations are used.
spatial.entity.pred	An optional vector indicating the spatial entity identifier (from one to the total number of entities) for predictions. If NULL, the vector <code>spatial.entity</code> for observations is used.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to <code>burnin+mcmc</code> . <code>burnin+mcmc</code> must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.

thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
beta.start	Starting values for $\beta$ parameters of the suitability process. This can either be a scalar or a $p$ -length vector.
gamma.start	Starting values for $\beta$ parameters of the observability process. This can either be a scalar or a $q$ -length vector.
Vrho.start	Positive scalar indicating the starting value for the variance of the spatial random effects.
mubeta	Means of the priors for the $\beta$ parameters of the suitability process. mubeta must be either a scalar or a $p$ -length vector. If mubeta takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value is set to 0 for an uninformative prior.
Vbeta	Variances of the Normal priors for the $\beta$ parameters of the suitability process. Vbeta must be either a scalar or a $p$ -length vector. If Vbeta takes a scalar value, then that value will serve as the prior variance for all of the betas. The default variance is large and set to 1.0E6 for an uninformative flat prior.
mugamma	Means of the Normal priors for the $\gamma$ parameters of the observability process. mugamma must be either a scalar or a $p$ -length vector. If mugamma takes a scalar value, then that value will serve as the prior mean for all of the gammas. The default value is set to 0 for an uninformative prior.
Vgamma	Variances of the Normal priors for the $\gamma$ parameters of the observability process. Vgamma must be either a scalar or a $p$ -length vector. If Vgamma takes a scalar value, then that value will serve as the prior variance for all of the gammas. The default variance is large and set to 1.0E6 for an uninformative flat prior.
priorVrho	Type of prior for the variance of the spatial random effects. Can be set to a fixed positive scalar, or to an inverse-gamma distribution ("1/Gamma") with parameters shape and rate, or to a uniform distribution ("Uniform") on the interval [0,Vrho.max]. Default set to "1/Gamma".
shape	The shape parameter for the Gamma prior on the precision of the spatial random effects. Default value is shape=0.05 for uninformative prior.
rate	The rate (1/scale) parameter for the Gamma prior on the precision of the spatial random effects. Default value is rate=0.0005 for uninformative prior.
Vrho.max	Upper bound for the uniform prior of the spatial random effect variance. Default set to 1000.
seed	The seed for the random number generator. Default set to 1234.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.
save.rho	A switch (0,1) which determines whether or not the sampled values for rhos are saved. Default is 0: the posterior mean is computed and returned in the rho.pred vector. Be careful, setting save.rho to 1 might require a large amount of memory.
save.p	A switch (0,1) which determines whether or not the sampled values for predictions are saved. Default is 0: the posterior mean is computed and returned in

	the prob.p.pred vector. Be careful, setting save.p to 1 might require a large amount of memory.
save.N	A switch (0,1) which determines whether or not the sampled values for the latent count variable N for each observed cells are saved. Default is 0: the mean (rounded to the closest integer) is computed and returned in the N.pred vector. Be careful, setting save.N to 1 might require a large amount of memory.

## Details

The model integrates two processes, an ecological process associated to the presence or absence of the species due to habitat suitability and an observation process that takes into account the fact that the probability of detection of the species is inferior to one. The ecological process includes an intrinsic conditional autoregressive model (iCAR) model for spatial autocorrelation between observations, assuming that the probability of presence of the species at one site depends on the probability of presence of the species on neighboring sites.

### Ecological process:

$$N_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = X_i\beta + \rho_i$$

$\rho_i$ : spatial random effect

### Spatial autocorrelation:

An intrinsic conditional autoregressive model (iCAR) is assumed:

$$\rho_i \sim \text{Normal}(\mu_i, V_\rho/n_i)$$

$\mu_i$ : mean of  $\rho_{i'}$  in the neighborhood of  $i$ .

$V_\rho$ : variance of the spatial random effects.

$n_i$ : number of neighbors for spatial entity  $i$ .

### Observation process:

$$y_{it} \sim \text{Binomial}(N_i, \delta_{it})$$

$$\text{logit}(\delta_{it}) = W_{it}\gamma$$

## Value

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance $D$ , with $D = -2 \log(\prod_{it} P(y_{it}, N_i   \dots))$ , is also provided.
rho.pred	If save.rho is set to 0 (default), rho.pred is the predictive posterior mean of the spatial random effect associated to each spatial entity. If save.rho is set to 1, rho.pred is an mcmc object with sampled values for each spatial random effect associated to each spatial entity.
prob.p.pred	If save.p is set to 0 (default), prob.p.pred is the predictive posterior mean of the probability associated to the suitability process for each prediction. If save.p is set to 1, prob.p.pred is an mcmc object with sampled values of the probability associated to the suitability process for each prediction.

N.pred	If save.N is set to 0 (default), N.pred is the posterior mean (rounded to the closest integer) of the latent count variable N for each observed cell. If save.N is set to 1, N.pred is an mcmc object with sampled values of the latent count variable N for each observed cell.
prob.p.latent	Predictive posterior mean of the probability associated to the suitability process for each observation.
prob.q.latent	Predictive posterior mean of the probability associated to the observability process for each observation.

### Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

### References

- Gelfand, A. E.; Schmidt, A. M.; Wu, S.; Silander, J. A.; Latimer, A. and Rebelo, A. G. (2005) Modelling species diversity through species level hierarchical modelling. *Applied Statistics*, 54, 1-20.
- Gilks, W. R., Best, N. G. and Tan, K. K. C. (1995) Adaptive rejection Metropolis sampling. *Applied Statistics*, 44, 455-472.
- Latimer, A. M.; Wu, S. S.; Gelfand, A. E. and Silander, J. A. (2006) Building statistical models to analyze species distributions. *Ecological Applications*, 16, 33-50.
- MacKenzie, D. I.; Nichols, J. D.; Lachman, G. B.; Droege, S.; Andrew Royle, J. and Langtimm, C. A. (2002) Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83, 2248-2255.

### See Also

[plot.mcmc](#), [summary.mcmc](#)

### Examples

```
## Not run:

#####
# hSDM.Nmixture()
# Example with simulated data
#####

=====
#== Preamble
library(hSDM)
library(raster)
library(sp)
library(mvtnorm)

=====
#== Data simulation
```



```

# Set seed for repeatability
seed <- 1234

# Target parameters
beta.target <- matrix(c(0.2,0.5,0.5),ncol=1)
gamma.target <- matrix(c(0.2),ncol=1) #= This means that theta.2~0.5: 50% of chance to detect the species
## Uncomment if you want covariates on the observability process
## gamma.target <- matrix(c(0.2,0.5,0.5),ncol=1)
Vrho.target <- 1 # Spatial Variance

# Landscape
Landscape <- raster(ncol=20,nrow=20,crs='+proj=utm +zone=1')
ncell <- ncell(Landscape)

# Neighbors
neighbors.mat <- adjacent(Landscape, cells=c(1:ncell), directions=8, pairs=TRUE, sorted=TRUE)
n.neighbors <- as.data.frame(table(as.factor(neighbors.mat[,1])))[,2]
adj <- neighbors.mat[,2]

# Generate symmetric adjacency matrix, A
A <- matrix(0,ncell,ncell)
index.start <- 1
for (i in 1:ncell) {
  index.end <- index.start+n.neighbors[i]-1
  A[i,adj[c(index.start:index.end)]] <- 1
  index.start <- index.end+1
}

# Spatial effects
d <- 1 # Spatial dependence parameter = 1 for intrinsic CAR
Q <- diag(n.neighbors)-d*A + diag(.0001,ncell) # Add small constant to make Q non-singular
covrho <- Vrho.target*solve(Q) # Covariance of rhos
set.seed(seed)
rho <- c(rmvnorm(1,sigma=covrho)) # Spatial Random Effects
rho <- rho-mean(rho) # Centering rhos on zero

# Visited cells
n.visited <- 150 # Compare with 400, 350 and 100 for example
set.seed(seed)
visited.cells <- sort(sample(1:ncell,n.visited,replace=FALSE)) # Draw visited cells at random
notvisited.cells <- c(1:ncell)[-visited.cells]

# Number of observations
nobs <- 300

# Cell vector
set.seed(seed)
cells <- c(visited.cells,sample(visited.cells,nobs-n.visited,replace=TRUE))
coords <- xyFromCell(Landscape,cells) # Get coordinates

# Covariates for "suitability" process
set.seed(seed)

```

```

X1 <- rnorm(n=ncell,0,1) # Here, we decided to define one habitat by spatial entity (raster cell)
set.seed(2*seed)
X2 <- rnorm(n=ncell,0,1)
X <- cbind(rep(1,ncell),X1,X2)

# Covariates for "observability" process
## Uncomment if you want covariates on the observability process
## set.seed(3*seed)
## W1 <- rnorm(n=nobs,0,1)
## set.seed(4*seed)
## W2 <- rnorm(n=nobs,0,1)
## W <- cbind(rep(1,nobs),W1,W2)
W <- cbind(rep(1,nobs))

#== Simulating latent variables

# Suitability
log.lambda <- vector()
log.lambda <- X%*%beta.target+rho
lambda <- exp(log.lambda)
set.seed(seed)
N <- rpois(ncell,lambda)

# Observability
logit.theta <- W%*%gamma.target
theta <- inv.logit(logit.theta)

#== Simulating response variable
set.seed(seed)
Y <- rbinom(nobs,N[cells],theta)

#== Data-sets
Data.obs <- data.frame(Y,cells)
## Uncomment if you want covariates on the observability process
## Data.obs <- data.frame(Y,cells,W1,W2)
Data.suit <- data.frame(cells=c(1:ncell),X1,X2)
Data.obs <- SpatialPointsDataFrame(coords=coords,data=Data.obs)
plot(Data.obs)
Data.pred <- Data.suit

#=====
#== N-mixture model

mod.hSDM.Nmixture <- hSDM.Nmixture(# Observations
                                   counts=Data.obs$Y, observability=~1,
                                   spatial.entity=Data.obs$cells, data.observability=Data.obs,
                                   # Habitat
                                   suitability=~X1+X2, data.suitability=Data.suit,
                                   # Spatial structure
                                   n.neighbors=n.neighbors, neighbors=adj,
                                   # Predictions
                                   suitability.pred=Data.pred, spatial.entity.pred=Data.pred$cells,
                                   # Chains

```

```

burnin=10000, mcmc=10000, thin=10,
# Starting values
beta.start=0,
gamma.start=0,
Vrho.start=10,
# Priors
mubeta=0, Vbeta=1.0E6,
mugamma=0, Vgamma=1.0E6,
priorVrho="1/Gamma",
#priorVrho="Uniform",
#priorVrho=10,
shape=0.5, rate=0.0005,
#Vrho.max=1000,
# Various
seed=1234, verbose=1,
save.rho=0, save.p=0, save.N=1)

#=====
#== Outputs

# Parameter estimates
summary(mod.hSDM.Nmixture$mcmc)

# MCMC and posteriors
pdf(file="Posteriors_hSDM.Nmixture.pdf")
plot(mod.hSDM.Nmixture$mcmc)
dev.off()

pdf(file="Posteriors.rho_hSDM.Nmixture.pdf")
plot(mod.hSDM.Nmixture$rho.pred)
dev.off()

pdf(file="Posteriors.N_hSDM.Nmixture.pdf")
plot(mod.hSDM.Nmixture$N.pred)
dev.off()

# Check that Ns are correctly estimated
M <- as.matrix(mod.hSDM.Nmixture$N.pred)
N.est <- apply(M,2,mean)
Y.by.cell <- tapply(Data.obs$Y,Data.obs$cells,mean)
pdf(file="Check_N.pdf",width=10,height=5)
par(mfrow=c(1,2))
plot(Y.by.cell, N.est) ## More individuals are expected in reality (N > Y) due to detection process
abline(a=0,b=1,col="red")
plot(N[visited.cells], N.est) ## N are well estimated
abline(a=0,b=1,col="red")
cor(N[visited.cells], N.est) ## Close to 1
dev.off()

# Summary plots

# rho
r.rho <- r.rho.pred <- r.visited <- Landscape

```

```

r.rho[] <- rho
rho.pred <- mod.hSDM.Nmixture$rho.pred # apply(mod.hSDM.Nmixture$rho.pred,2,mean)
r.rho.pred[] <- rho.pred
r.visited[] <- 0
r.visited[visited.cells] <- 1
# prob.p
r.prob.p <- Landscape
r.prob.p[] <- mod.hSDM.Nmixture$prob.p.pred

pdf(file="Summary_hSDM.Nmixture.pdf")
par(mfrow=c(3,2))
plot(r.rho, main="rho target")
plot(r.visited,main="Visited cells and presences")
plot(Data.obs[Y>0,],add=TRUE,pch=16,cex=0.5)
plot(r.rho.pred, main="rho estimated")
plot(rho[visited.cells],rho.pred[visited.cells],
      xlab="rho target",
      ylab="rho estimated")
points(rho[notvisited.cells],rho.pred[notvisited.cells],pch=16,col="blue")
legend(x=-4,y=3.5,legend="Unvisited cells",col="blue",pch=16,bty="n")
abline(a=0,b=1,col="red")
plot(r.prob.p,main="Proba of presence")
plot(Data.obs[Y>0,],add=TRUE,pch=16,cex=0.5)
dev.off()

## End(Not run)

```

hSDM.poisson

*The hSDM.poisson function performs a Poisson regression in a Bayesian framework.*

## Description

The hSDM.poisson function calls a Gibbs sampler written in C code which uses a Metropolis algorithm to estimate the conditional posterior distribution of model's parameters.

## Usage

```
hSDM.poisson(counts, visits, suitability, data, burnin = 5000, mcmc =
10000, thin = 10, beta.start, mubeta = 0, Vbeta = 1e+06, seed = 1234,
verbose = 1)
```

## Arguments

counts	A vector indicating the count (or abundance) for each observation.
visits	A vector indicating if the observation point n has been visited (visits[n]=1) or not (visits[n]=0). If visits[n]=0, then counts[n]=0.

suitability	A one-sided formula of the form ' $\sim x_1 + \dots + x_p$ ' with $p$ terms specifying the explicative covariates for the suitability process of the model.
data	A data frame containing the model's explicative variables.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
beta.start	Starting values for beta parameters of the suitability process. If beta.start takes a scalar value, then that value will serve for all of the betas.
mubeta	Means of the priors for the $\beta$ parameters of the suitability process. mubeta must be either a scalar or a $p$ -length vector. If mubeta takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value is set to 0 for an uninformative prior.
Vbeta	Variances of the Normal priors for the $\beta$ parameters of the suitability process. Vbeta must be either a scalar or a $p$ -length vector. If Vbeta takes a scalar value, then that value will serve as the prior variance for all of the betas. The default variance is large and set to 1.0E6 for an uninformative flat prior.
seed	The seed for the random number generator. Default to 1234.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.

**Value**

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance $D$ , with $D = -2 \log(\prod_i P(y_i, n_i   \beta))$ , is also provided.
prob.p.pred	Predictive posterior mean of the probability associated to the suitability process for each spatial cell.

**Author(s)**

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

**References**

- Latimer, A. M.; Wu, S. S.; Gelfand, A. E. and Silander, J. A. (2006) Building statistical models to analyze species distributions. *Ecological Applications*, 16, 33-50.
- Gelfand, A. E.; Schmidt, A. M.; Wu, S.; Silander, J. A.; Latimer, A. and Rebelo, A. G. (2005) Modelling species diversity through species level hierarchical modelling. *Applied Statistics*, 54, 1-20.

**See Also**

[plot.mcmc](#), [summary.mcmc](#)

## Examples

```
## Not run:

#####
# hSDM.hierarchical.poisson()
# Example with simulated data
#####

#== Preamble
library(hSDM)

#####
#== Data simulation

# Set seed for repeatability
set.seed(1234)

# Constants
nobs <- 1000

# Covariates for "suitability" process
X1 <- rnorm(n=nobs,0,1)
X2 <- rnorm(n=nobs,0,1)
X <- cbind(rep(1,nobs),X1,X2)

# Target parameters
beta.target <- matrix(c(0.2,0.1,0.1),ncol=1) # fixed effects

#== Simulating latent variables

# Suitability
log.theta <- X %*% beta.target
theta <- exp(log.theta)

#== Simulating response variable
Y <- rpois(n=nobs,lambda=theta)
V <- rep(1,nobs)

#== Data-set
Data <- data.frame(Y,V,X1,X2)
str(Data)
par(mfrow=c(2,1))
plot(Data$X1,log.theta)
plot(Data$X2,log.theta)

#####
#== Statistical modelling

#== glm resolution to compare
mod.glm <- glm(Y~X1+X2,family="poisson",data=Data)
summary(mod.glm)
```

```

beta.hat.glm <- coef(mod.glm)

#== hSDM
mod.hSDM.poisson <- hSDM.poisson(counts=Data$Y,
                                visits=Data$V,
                                suitability=~X1+X2,
                                data=Data, burnin=1000,
                                mcmc=1000, thin=1,
                                beta.start=beta.hat.glm,
                                mubeta=0, Vbeta=1.0E6,
                                seed=1234, verbose=1)

#=====
#== Outputs
summary(mod.hSDM.poisson$mcmc)
pdf(file="Posteriors_hSDM.poisson.pdf")
plot(mod.hSDM.poisson$mcmc)
dev.off()
summary(mod.hSDM.poisson$prob.p.pred)

## End(Not run)

```

---

hSDM.poisson.iCAR	<i>The hSDM.poisson.iCAR function performs a Poisson regression in a hierarchical Bayesian framework. The suitability process includes a spatial correlation process. The spatial correlation is modelled using an intrinsic CAR model.</i>
-------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

---

## Description

The hSDM.poisson.iCAR function calls a Gibbs sampler written in C code which uses a Metropolis algorithm to estimate the conditional posterior distribution of model's parameters.

## Usage

```

hSDM.poisson.iCAR(counts, visits, suitability, cells,
n.neighbors, neighbors, data, burnin = 5000, mcmc = 10000, thin = 10,
beta.start, Vrho.start, mubeta = 0, Vbeta = 1e+06, priorVrho =
"1/Gamma", shape = 0.5, rate = 0.0005, Vrho.max = 1000, seed = 1234,
verbose = 1, save.rho = 0)

```

## Arguments

counts	A vector indicating the count (or abundance) for each observation.
visits	A vector indicating if the observation point n has been visited (visits[n]=1) or not (visits[n]=0). If visits[n]=0, then counts[n]=0.

suitability	A one-sided formula of the form ' $\sim x_1 + \dots + x_p$ ' with $p$ terms specifying the explicative variables for the suitability process of the model.
cells	A vector indicating the spatial cell identifier (from 1 to total number of cell) for each observation. Several observations can occur in one spatial cell.
n.neighbors	A vector of integers that indicates the number of neighbors (adjacent cells) of each spatial cell.
neighbors	A vector of integers indicating the neighbors (adjacent cells) of each spatial cell. Must be of the form $c(\text{neighbors of cell 1, neighbors of cell 2, } \dots, \text{neighbors of the last cell})$ . Length of the neighbors vector should be equal to $\text{sum}(\text{data}\$num)$ .
data	A data frame containing the model's explicative variables.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to $\text{burnin} + \text{mcmc}$ . $\text{burnin} + \text{mcmc}$ must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
beta.start	Starting values for beta parameters.
Vrho.start	Positive scalar indicating the starting value for the variance of the spatial random effects.
mubeta	Means of the priors for the $\beta$ parameters of the suitability process. mubeta must be either a scalar or a $p$ -length vector. If mubeta takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value is set to 0 for an uninformative prior.
Vbeta	Variances of the Normal priors for the $\beta$ parameters of the suitability process. Vbeta must be either a scalar or a $p$ -length vector. If Vbeta takes a scalar value, then that value will serve as the prior variance for all of the betas. The default variance is large and set to $1.0E6$ for an uninformative flat prior.
priorVrho	Type of prior for the variance of the spatial random effects. Can be set to a fixed positive scalar, or to an inverse-gamma distribution (" $1/\text{Gamma}$ ") with parameters shape and rate, or to a uniform distribution ("Uniform") on the interval $[0, Vrho.max]$ . Default to " $1/\text{Gamma}$ ".
shape	The shape parameter for the Gamma prior on the precision of the spatial random effects. Default value is $\text{shape} = 0.05$ for uninformative prior.
rate	The rate ( $1/\text{scale}$ ) parameter for the Gamma prior on the precision of the spatial random effects. Default value is $\text{rate} = 0.0005$ for uninformative prior.
Vrho.max	Upper bound for the uniform prior of the spatial random effect variance. Default to 1000.
seed	The seed for the random number generator. Default to 1234.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.



`save.rho` A switch (0,1) which determines whether or not the sampled values for rhos are saved. Default is 0: the posterior mean is computed and returned in the `rho.pred` vector. Be careful, setting `save.rho` to 1 might require a large amount of memory.

### Value

`mcmc` An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance  $D$ , with  $D = -2 \log(\prod_i P(y_i, n_i | \beta, \rho_i))$ , is also provided.

`rho.pred` If `save.rho` is set to 0 (default), `rho.pred` is the predictive posterior mean of the spatial random effect associated to each spatial cell. If `save.rho` is set to 1, `rho.pred` is an mcmc object with sampled values for each spatial random effect associated to each spatial cell.

`prob.p.pred` Predictive posterior mean of the probability associated to the suitability process for each spatial cell.

### Author(s)

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

### References

Latimer, A. M.; Wu, S. S.; Gelfand, A. E. and Silander, J. A. (2006) Building statistical models to analyze species distributions. *Ecological Applications*, 16, 33-50.

Gelfand, A. E.; Schmidt, A. M.; Wu, S.; Silander, J. A.; Latimer, A. and Rebelo, A. G. (2005) Modelling species diversity through species level hierarchical modelling. *Applied Statistics*, 54, 1-20.

### See Also

[plot.mcmc](#), [summary.mcmc](#)

### Examples

```
## Not run:

#####
# hSDM.poisson.iCAR()
# Example with simulated data
#####

#== Preamble
library(mvtnorm)
library(lme4) # To compare with nonspatial random effects model
library(hSDM)

#####
#== Data simulation
```

```

# Set seed for repeatability
set.seed(1234)

# Constants
ncell <- 150 # Number of cells
nobs <- 10*ncell # Number of observation for the *poisson* random variable
cell <- rep(c(1:ncell),each=nobs/ncell)

# Covariates for "suitability" process
X1 <- rnorm(n=nobs,0,1)
X2 <- rnorm(n=nobs,0,1)
X <- cbind(rep(1,nobs),X1,X2)

# Target parameters
beta.target <- matrix(c(0.2,0.1,0.1),ncol=1) # fixed effects
Vrho.target <- 10 # Spatial Variance

# Generate symmetric adjacency matrix, A
A <- matrix(0,ncell,ncell)
A[upper.tri(A,diag=F)] <- rbinom(ncell*(ncell-1)/2,1,.05)
A <- A+t(A)
n.neighbors <- apply(A,1,sum)
f.adjacent <- function(x) {
  which(x==1)
}
adj <- unlist(apply(A,1,f.adjacent))

# Spatial effects, phi
d <- 1 # Spatial dependence parameter = 1 for intrinsic CAR
Q <- diag(n.neighbors)-d*A + diag(.0001,ncell) # Add small constant to make Q non-singular
covrho <- Vrho.target*solve(Q) # Covariance of rhos
rho <- c(rmvnorm(1,sigma=covrho)) # Spatial Random Effects
rho <- rho-mean(rho) # Centering rhos on zero

#== Simulating latent variables

# Suitability
log.theta <- vector()
for (n in 1:nobs) {
  log.theta[n] <- X[n,]%*%beta.target+rho[cell[n]]
}
theta <- exp(log.theta)

#== Simulating response variable
Y <- rpois(nobs,theta)
V <- rep(1,nobs)

#== Data-set
Data <- data.frame(Y,V,cell,X1,X2)
str(Data)

#=====

```

```

#== Statistical modelling

#== glm resolution to compare
mod.glm <- glm(Y~X1+X2,data=Data,family="poisson")
summary(mod.glm)

#== glmm (mixed models) resolution to compare
#== cell random effects without spatial correlation
mod.glmm <- lmer(Y~X1+X2+(1|cell),data=Data,family="poisson") # Non-spatial random effect fit
summary(mod.glmm)
beta.hat.glmm <- fixef(mod.glmm)

#== hSDM
mod.hSDM.poisson.iCAR <- hSDM.poisson.iCAR(counts=Data$Y,
                                           visits=Data$V,
                                           suitability=~X1+X2,
                                           cells=Data$cell,
                                           n.neighbors=n.neighbors,
                                           neighbors=adj,
                                           data=Data, burnin=1000,
                                           mcmc=1000, thin=1,
                                           beta.start=beta.hat.glmm,
                                           Vrho.start=10,
                                           priorVrho="1/Gamma",
                                           #priorVrho="Uniform",
                                           #priorVrho=10,
                                           mubeta=0, Vbeta=1.0E6,
                                           shape=0.5, rate=0.0005,
                                           Vrho.max=1000,
                                           seed=1234, verbose=1, save.rho=0)

#=====
#== Outputs
summary(mod.hSDM.poisson.iCAR$mcmc)
pdf(file="Posteriors_hSDM.poisson.iCAR.pdf")
plot(mod.hSDM.poisson.iCAR$mcmc)
dev.off()
summary(mod.hSDM.poisson.iCAR$prob.p.pred)
summary(mod.hSDM.poisson.iCAR$rho.pred)

## End(Not run)

```

---

hSDM.site.occupancy     *The hSDM.site.occupancy function can be used to model species distribution including different processes in a hierarchical Bayesian framework: a Bernoulli suitability process (referring to environmental suitability) which takes into account the spatial dependence of the observations, and a Binomial observability process (referring to various ecological and methodological issues explaining the species detection).*

---

## Description

The hSDM.site.occupancy function calls a Gibbs sampler written in C code which uses a Metropolis algorithm to estimate the conditional posterior distribution of hierarchical model's parameters.

## Usage

```
hSDM.site.occupancy(presences, trials, suitability,
observability, spatial.entity, data, n.neighbors, neighbors,
suitability.pred=NULL, spatial.entity.pred=NULL, burnin = 5000, mcmc =
10000, thin = 10, beta.start, gamma.start, Vrho.start, mubeta = 0, Vbeta
= 1e+06, mugamma = 0, Vgamma = 1e+06, priorVrho = "1/Gamma", shape =
0.5, rate = 0.0005, Vrho.max=1000, seed = 1234, verbose = 1, save.rho =
0, save.p = 0)
```

## Arguments

presences	A vector indicating the number of successes (or presences) for each observation.
trials	A vector indicating the number of trials for each observation. $t_i$ should be superior to zero and superior or equal to $y_i$ , the number of successes for observation $i$ .
suitability	A one-sided formula of the form $\sim x_1 + \dots + x_p$ with $p$ terms specifying the explicative variables for the suitability process.
observability	A one-sided formula of the form $\sim w_1 + \dots + w_q$ with $q$ terms specifying the explicative variables for the observability process.
spatial.entity	A vector indicating the spatial entity identifier (from one to the total number of entities) for each observation. Several observations can occur in one spatial entity. A spatial entity can be a raster cell for example.
data	A data frame containing the model's variables.
n.neighbors	A vector of integers that indicates the number of neighbors (adjacent entities) of each spatial entity. length(n.neighbors) indicates the total number of spatial entities.
neighbors	A vector of integers indicating the neighbors (adjacent entities) of each spatial entity. Must be of the form c(neighbors of entity 1, neighbors of entity 2, ... , neighbors of the last entity). Length of the neighbors vector should be equal to sum(n.neighbors).

suitability.pred	An optional data frame in which to look for variables with which to predict. If NULL, the observations are used.
spatial.entity.pred	An optional vector indicating the spatial entity identifier (from one to the total number of entities) for predictions. If NULL, the vector spatial.entity for observations is used.
burnin	The number of burnin iterations for the sampler.
mcmc	The number of Gibbs iterations for the sampler. Total number of Gibbs iterations is equal to burnin+mcmc. burnin+mcmc must be divisible by 10 and superior or equal to 100 so that the progress bar can be displayed.
thin	The thinning interval used in the simulation. The number of mcmc iterations must be divisible by this value.
beta.start	Starting values for $\beta$ parameters of the suitability process. This can either be a scalar or a $p$ -length vector.
gamma.start	Starting values for $\beta$ parameters of the observability process. This can either be a scalar or a $q$ -length vector.
Vrho.start	Positive scalar indicating the starting value for the variance of the spatial random effects.
mubeta	Means of the priors for the $\beta$ parameters of the suitability process. mubeta must be either a scalar or a $p$ -length vector. If mubeta takes a scalar value, then that value will serve as the prior mean for all of the betas. The default value is set to 0 for an uninformative prior.
Vbeta	Variances of the Normal priors for the $\beta$ parameters of the suitability process. Vbeta must be either a scalar or a $p$ -length vector. If Vbeta takes a scalar value, then that value will serve as the prior variance for all of the betas. The default variance is large and set to 1.0E6 for an uninformative flat prior.
mugamma	Means of the Normal priors for the $\gamma$ parameters of the observability process. mugamma must be either a scalar or a $p$ -length vector. If mugamma takes a scalar value, then that value will serve as the prior mean for all of the gammas. The default value is set to 0 for an uninformative prior.
Vgamma	Variances of the Normal priors for the $\gamma$ parameters of the observability process. Vgamma must be either a scalar or a $p$ -length vector. If Vgamma takes a scalar value, then that value will serve as the prior variance for all of the gammas. The default variance is large and set to 1.0E6 for an uninformative flat prior.
priorVrho	Type of prior for the variance of the spatial random effects. Can be set to a fixed positive scalar, or to an inverse-gamma distribution ("1/Gamma") with parameters shape and rate, or to a uniform distribution ("Uniform") on the interval $[0, Vrho.max]$ . Default set to "1/Gamma".
shape	The shape parameter for the Gamma prior on the precision of the spatial random effects. Default value is shape=0.05 for uninformative prior.
rate	The rate (1/scale) parameter for the Gamma prior on the precision of the spatial random effects. Default value is rate=0.0005 for uninformative prior.
Vrho.max	Upper bound for the uniform prior of the spatial random effect variance. Default set to 1000.

seed	The seed for the random number generator. Default set to 1234.
verbose	A switch (0,1) which determines whether or not the progress of the sampler is printed to the screen. Default is 1: a progress bar is printed, indicating the step (in %) reached by the Gibbs sampler.
save.rho	A switch (0,1) which determines whether or not the sampled values for rhos are saved. Default is 0: the posterior mean is computed and returned in the rho.pred vector. Be careful, setting save.rho to 1 might require a large amount of memory.
save.p	A switch (0,1) which determines whether or not the sampled values for predictions are saved. Default is 0: the posterior mean is computed and returned in the prob.p.pred vector. Be careful, setting save.p to 1 might require a large amount of memory.

### Details

The model integrates two processes, an ecological process associated to the presence or absence of the species due to habitat suitability and an observation process that takes into account the fact that the probability of detection of the species is inferior to one. The ecological process includes an intrinsic conditional autoregressive model (iCAR) model for spatial autocorrelation between observations, assuming that the probability of presence of the species at one site depends on the probability of presence of the species on neighboring sites.

#### Ecological process:

$$z_i \sim \text{Bernoulli}(\theta_i)$$

$$\text{logit}(\theta_i) = X_i\beta + \rho_{j(i)}$$

$\rho_j$ : spatial random effect

$j(i)$ : index of the spatial entity for observation  $i$ .

#### Spatial autocorrelation:

An intrinsic conditional autoregressive model (iCAR) is assumed:

$$\rho_j \sim \text{Normal}(\mu_j, V_\rho/n_j)$$

$\mu_j$ : mean of  $\rho_{j'}$  in the neighborhood of  $j$ .

$V_\rho$ : variance of the spatial random effects.

$n_j$ : number of neighbors for spatial entity  $j$ .

#### Observation process:

$$y_i \sim \text{Binomial}(z_i * \delta_i, t_i)$$

$$\text{logit}(\delta_i) = W_i\gamma$$

### Value

mcmc	An mcmc object that contains the posterior sample. This object can be summarized by functions provided by the coda package. The posterior sample of the deviance $D$ , with $D = -2 \log(\prod_i P(y_i, z_i   \dots))$ , is also provided.
------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

rho.pred	If save.rho is set to 0 (default), rho.pred is the predictive posterior mean of the spatial random effect associated to each spatial entity. If save.rho is set to 1, rho.pred is an mcmc object with sampled values for each spatial random effect associated to each spatial entity.
prob.p.pred	If save.p is set to 0 (default), prob.p.pred is the predictive posterior mean of the probability associated to the suitability process for each prediction. If save.p is set to 1, prob.p.pred is an mcmc object with sampled values of the probability associated to the suitability process for each prediction.
prob.p.latent	Predictive posterior mean of the probability associated to the suitability process for each observation.
prob.q.latent	Predictive posterior mean of the probability associated to the observability process for each observation.

**Author(s)**

Ghislain Vieilledent <ghislain.vieilledent@cirad.fr>

**References**

- Gelfand, A. E.; Schmidt, A. M.; Wu, S.; Silander, J. A.; Latimer, A. and Rebelo, A. G. (2005) Modelling species diversity through species level hierarchical modelling. *Applied Statistics*, 54, 1-20.
- Gilks, W. R., Best, N. G. and Tan, K. K. C. (1995) Adaptive rejection Metropolis sampling. *Applied Statistics*, 44, 455-472.
- Latimer, A. M.; Wu, S. S.; Gelfand, A. E. and Silander, J. A. (2006) Building statistical models to analyze species distributions. *Ecological Applications*, 16, 33-50.
- MacKenzie, D. I.; Nichols, J. D.; Lachman, G. B.; Droege, S.; Andrew Royle, J. and Langtimm, C. A. (2002) Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83, 2248-2255.

**See Also**

[plot.mcmc](#), [summary.mcmc](#)

**Examples**

```
## Not run:

#####
# hSDM.binomial.iCAR()
# Example with simulated data
#####

=====
#== Preamble
library(hSDM)
library(raster)
library(sp)
```

```

library(mvtnorm)

#####
#== Data simulation

# Set seed for repeatability
seed <- 1234

# Target parameters
beta.target <- matrix(c(0.2,0.5,0.5),ncol=1)
gamma.target <- matrix(c(0.2),ncol=1) #= This means that theta.2~0.5: 50% of chance to detect the species
## Uncomment if you want covariates on the observability process
## gamma.target <- matrix(c(0.2,0.5,0.5),ncol=1)
Vrho.target <- 1 # Spatial Variance

# Landscape
Landscape <- raster(ncol=20,nrow=20,crs='+proj=utm +zone=1')
ncell <- ncell(Landscape)

# Neighbors
neighbors.mat <- adjacent(Landscape, cells=c(1:ncell), directions=8, pairs=TRUE, sorted=TRUE)
n.neighbors <- as.data.frame(table(as.factor(neighbors.mat[,1])))[,2]
adj <- neighbors.mat[,2]

# Generate symmetric adjacency matrix, A
A <- matrix(0,ncell,ncell)
index.start <- 1
for (i in 1:ncell) {
  index.end <- index.start+n.neighbors[i]-1
  A[i,adj[c(index.start:index.end)]] <- 1
  index.start <- index.end+1
}

# Spatial effects
d <- 1 # Spatial dependence parameter = 1 for intrinsic CAR
Q <- diag(n.neighbors)-d*A + diag(.0001,ncell) # Add small constant to make Q non-singular
covrho <- Vrho.target*solve(Q) # Covariance of rhos
set.seed(seed)
rho <- c(rmvnorm(1,sigma=covrho)) # Spatial Random Effects
rho <- rho-mean(rho) # Centering rhos on zero

# Visited cells
n.visited <- 150 # Compare with 400, 350 and 100 for example
set.seed(seed)
visited.cells <- sort(sample(1:ncell,n.visited,replace=FALSE)) # Draw visited cells at random
notvisited.cells <- c(1:ncell)[-visited.cells]

# Number of observations
nobs <- 300

# Cell vector
set.seed(seed)
cells <- c(visited.cells,sample(visited.cells,nobs-n.visited,replace=TRUE))

```



```

coords <- xyFromCell(Landscape,cells) # Get coordinates

# Covariates for "suitability" process
set.seed(seed)
X1.cell <- rnorm(n=ncell,0,1) # Here, we decided to define one habitat by spatial entity (raster cell)
set.seed(2*seed)
X2.cell <- rnorm(n=ncell,0,1)
X1 <- X1.cell[cells]
X2 <- X2.cell[cells]
X <- cbind(rep(1,nobs),X1,X2)

# Covariates for "observability" process
## Uncomment if you want covariates on the observability process
## set.seed(3*seed)
## W1 <- rnorm(n=nobs,0,1)
## set.seed(4*seed)
## W2 <- rnorm(n=nobs,0,1)
## W <- cbind(rep(1,nobs),W1,W2)
W <- cbind(rep(1,nobs))

#== Simulating latent variables

# Suitability
logit.theta.1 <- vector()
for (n in 1:nobs) {
  logit.theta.1[n] <- X[n,]%*%beta.target+rho[cells[n]]
}
theta.1 <- inv.logit(logit.theta.1)
set.seed(seed)
y.1 <- rbinom(nobs,1,theta.1)

# Observability
set.seed(seed)
trials <- rpois(nobs,5) # Number of trial associated to each observation
trials[trials==0] <- 1
logit.theta.2 <- W*%gamma.target
theta.2 <- inv.logit(logit.theta.2)
set.seed(seed)
y.2 <- rbinom(nobs,trials,theta.2)

#== Simulating response variable
Y <- y.2*y.1

#== Data-set
Data <- data.frame(Y,trials,cells,X1,X2)
## Uncomment if you want covariates on the observability process
## Data <- data.frame(Y,trials,cells,X1,X2,W1,W2)
Data <- SpatialPointsDataFrame(coords=coords,data=Data)
plot(Data)

#== Data-set for predictions (suitability on each spatial cell)
Data.pred <- data.frame(X1=X1.cell,X2=X2.cell,cells=c(1:ncell))

```

```

#####
#== Site-occupancy model

mod.hSDM.site.occupancy <- hSDM.site.occupancy(presences=Data$Y,
                                              trials=Data$trials,
                                              suitability=~X1+X2,
                                              observability=~1,
                                              spatial.entity=Data$cells,
                                              data=Data,
                                              n.neighbors=n.neighbors,
                                              neighbors=adj,
                                              ## suitability.pred=NULL,
                                              ## spatial.entity.pred=NULL,
                                              suitability.pred=Data.pred,
                                              spatial.entity.pred=Data.pred$cells,
                                              burnin=5000, mcmc=5000, thin=5,
                                              beta.start=0,
                                              gamma.start=0,
                                              Vrho.start=10,
                                              priorVrho="1/Gamma",
                                              #priorVrho="Uniform",
                                              #priorVrho=10,
                                              mubeta=0, Vbeta=1.0E6,
                                              shape=0.5, rate=0.0005,
                                              #Vrho.max=1000,
                                              seed=1234, verbose=1,
                                              save.rho=1, save.p=0)

#####
#== Outputs

# Parameter estimates
summary(mod.hSDM.site.occupancy$mcmc)

# MCMC and posteriors
pdf(file="Posteriors_hSDM.site.occupancy.pdf")
plot(mod.hSDM.site.occupancy$mcmc)
dev.off()

pdf(file="Posteriors.rho_hSDM.site.occupancy.pdf")
plot(mod.hSDM.site.occupancy$rho.pred)
dev.off()

# Summary plots

# rho
r.rho <- r.rho.pred <- r.visited <- Landscape
r.rho[] <- rho
rho.pred <- apply(mod.hSDM.site.occupancy$rho.pred,2,mean)
r.rho.pred[] <- rho.pred
r.visited[] <- 0
r.visited[visited.cells] <- 1
# prob.p

```

```

r.prob.p <- Landscape
r.prob.p[] <- mod.hSDM.site.occupancy$prob.p.pred

pdf(file="Summary_hSDM.site.occupancy.pdf")
par(mfrow=c(3,2))
plot(r.rho, main="rho target")
plot(r.visited,main="Visited cells and presences")
plot(Data[Y>0,],add=TRUE,pch=16,cex=0.5)
plot(r.rho.pred, main="rho estimated")
plot(rho[visited.cells],rho.pred[visited.cells],
      xlab="rho target",
      ylab="rho estimated")
points(rho[notvisited.cells],rho.pred[notvisited.cells],pch=16,col="blue")
legend(x=-4,y=3.5,legend="Unvisited cells",col="blue",pch=16,bty="n")
abline(a=0,b=1,col="red")
plot(r.prob.p,main="Proba of presence")
plot(Data[Y>0,],add=TRUE,pch=16,cex=0.5)
dev.off()

## End(Not run)

```

---

logit

---

*Generalized logit and inverse logit function*


---

## Description

Compute generalized logit and generalized inverse logit functions.

## Usage

```

logit(x, min = 0, max = 1)
inv.logit(x, min = 0, max = 1)

```

## Arguments

x	value(s) to be transformed
min	Lower end of logit interval
max	Upper end of logit interval

## Details

The generalized logit function takes values on [min, max] and transforms them to span [-Inf,Inf] it is defined as:

$$y = \log\left(\frac{p}{(1-p)}\right)$$

where

$$p = \frac{(x - \min)}{(\max - \min)}$$

The generalized inverse logit function provides the inverse transformation:

$$x = p'(\max - \min) + \min$$

where

$$p' = \frac{\exp(y)}{(1 + \exp(y))}$$

**Value**

Transformed value(s).

**Author(s)**

Gregory R. Warnes <greg@warnes.net>

**Examples**

```
## Not run:
x <- seq(0,10, by=0.25)
xt <- logit(x, min=0, max=10)
cbind(x,xt)

y <- inv.logit(xt, min=0, max=10)
cbind(x,xt,y)

## End(Not run)
```

# Index

## \*Topic **Binomial logistic regression**

hSDM.binomial, [2](#)  
hSDM.binomial.iCAR, [5](#)

## \*Topic **MCMC**

hSDM-package, [2](#)  
hSDM.binomial, [2](#)  
hSDM.binomial.iCAR, [5](#)  
hSDM.hierarchical.binomial, [10](#)  
hSDM.hierarchical.poisson, [15](#)  
hSDM.Nmixture, [20](#)  
hSDM.poisson, [28](#)  
hSDM.poisson.iCAR, [31](#)  
hSDM.site.occupancy, [36](#)

## \*Topic **Markov Chains Monte Carlo**

hSDM-package, [2](#)  
hSDM.binomial, [2](#)  
hSDM.binomial.iCAR, [5](#)  
hSDM.hierarchical.binomial, [10](#)  
hSDM.hierarchical.poisson, [15](#)  
hSDM.Nmixture, [20](#)  
hSDM.poisson, [28](#)  
hSDM.poisson.iCAR, [31](#)  
hSDM.site.occupancy, [36](#)

## \*Topic **Metropolis algorithm**

hSDM-package, [2](#)  
hSDM.binomial, [2](#)  
hSDM.binomial.iCAR, [5](#)  
hSDM.hierarchical.binomial, [10](#)  
hSDM.hierarchical.poisson, [15](#)  
hSDM.Nmixture, [20](#)  
hSDM.poisson, [28](#)  
hSDM.poisson.iCAR, [31](#)  
hSDM.site.occupancy, [36](#)

## \*Topic **N-mixture models**

hSDM.Nmixture, [20](#)

## \*Topic **Poisson regression**

hSDM.poisson, [28](#)  
hSDM.poisson.iCAR, [31](#)

## \*Topic **ZIB models**

hSDM.hierarchical.binomial, [10](#)  
hSDM.Nmixture, [20](#)  
hSDM.site.occupancy, [36](#)

## \*Topic **ZIP models**

hSDM.hierarchical.poisson, [15](#)

## \*Topic **biodiversity**

hSDM-package, [2](#)  
hSDM.binomial, [2](#)  
hSDM.binomial.iCAR, [5](#)  
hSDM.hierarchical.binomial, [10](#)  
hSDM.hierarchical.poisson, [15](#)  
hSDM.Nmixture, [20](#)  
hSDM.poisson, [28](#)  
hSDM.poisson.iCAR, [31](#)  
hSDM.site.occupancy, [36](#)

## \*Topic **conditional autoregressive model**

hSDM-package, [2](#)  
hSDM.binomial.iCAR, [5](#)  
hSDM.hierarchical.binomial, [10](#)  
hSDM.hierarchical.poisson, [15](#)  
hSDM.Nmixture, [20](#)  
hSDM.poisson.iCAR, [31](#)  
hSDM.site.occupancy, [36](#)

## \*Topic **hierarchical Bayesian models**

hSDM-package, [2](#)  
hSDM.binomial.iCAR, [5](#)  
hSDM.hierarchical.binomial, [10](#)  
hSDM.hierarchical.poisson, [15](#)  
hSDM.Nmixture, [20](#)  
hSDM.poisson.iCAR, [31](#)  
hSDM.site.occupancy, [36](#)

## \*Topic **intrinsic CAR model**

hSDM-package, [2](#)  
hSDM.binomial.iCAR, [5](#)  
hSDM.hierarchical.binomial, [10](#)  
hSDM.hierarchical.poisson, [15](#)  
hSDM.Nmixture, [20](#)  
hSDM.poisson.iCAR, [31](#)

- hSDM.site.occupancy, [36](#)
- \*Topic **math**
  - logit, [43](#)
- \*Topic **spatial correlation**
  - hSDM-package, [2](#)
  - hSDM.binomial.iCAR, [5](#)
  - hSDM.hierarchical.binomial, [10](#)
  - hSDM.hierarchical.poisson, [15](#)
  - hSDM.Nmixture, [20](#)
  - hSDM.poisson.iCAR, [31](#)
  - hSDM.site.occupancy, [36](#)
- \*Topic **species distribution models**
  - hSDM-package, [2](#)
  - hSDM.binomial, [2](#)
  - hSDM.binomial.iCAR, [5](#)
  - hSDM.hierarchical.binomial, [10](#)
  - hSDM.hierarchical.poisson, [15](#)
  - hSDM.Nmixture, [20](#)
  - hSDM.poisson, [28](#)
  - hSDM.poisson.iCAR, [31](#)
  - hSDM.site.occupancy, [36](#)
- hSDM (hSDM-package), [2](#)
- hSDM-package, [2](#)
- hSDM.binomial, [2](#)
- hSDM.binomial.iCAR, [5](#)
- hSDM.hierarchical.binomial, [10](#)
- hSDM.hierarchical.poisson, [15](#)
- hSDM.Nmixture, [20](#)
- hSDM.poisson, [28](#)
- hSDM.poisson.iCAR, [31](#)
- hSDM.site.occupancy, [35](#)
- inv.logit (logit), [43](#)
- logit, [43](#)
- plot.mcmc, [4](#), [7](#), [12](#), [18](#), [24](#), [29](#), [33](#), [39](#)
- summary.mcmc, [4](#), [7](#), [12](#), [18](#), [24](#), [29](#), [33](#), [39](#)