Package 'hSDM'

March 4, 2013

Type Package

Title hierarchical Bayesian species distribution models

Version 1.1

Date 2012-11-19

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

URL http://hSDM.sf.net

LazyLoad yes

R topics documented:

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:	1.0
Date:	2012-11-19
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 β 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 β 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.pred.p` 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
#=============================================

#============
#== Preambule
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)
```

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 indicating 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).
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.
Vrho.start	Positive scalar indicating the starting value for the variance of the spatial random effects.
mubeta	Means of the priors for the β 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 β 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/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.

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	Predictive posterior mean of the spatial random effect associated to each spatial cell.
prob.pred.p	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
#=============================================

#============
#== Preambule
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 fi
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)

#==========
#== 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)
```

```
## 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 (refering to environmental suitability) which takes into account the spatial dependence of the observations, (ii) an alteration process (refering to anthropogenic disturbances), and (iii) a Binomial observability process (refering 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)
```

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 '~x1+...+xq' 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 β 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 β 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 γ 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 γ 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.

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` Predictive posterior mean of the spatial random effect associated to each spatial cell.

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

`prob.pred.q` 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
#=============================================

#============
#== Preambule
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)

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

```
## 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 (refering to environmental suitability) which takes into account the spatial dependence of the observations, (ii) an alteration process (refering to anthropogenic disturbances), and (iii) a Poisson observability process (refering 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)
```

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 '~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 '~x1+...+xq' with q terms specifying the explicative variables for the observability 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.
`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 β 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 β 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 γ 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 γ 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.

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` Predictive posterior mean of the spatial random effect associated to each spatial cell.

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

`prob.pred.q` 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
#=============================================

#============
#== Preambule
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)

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

```
## 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 '~x1+...+xp' 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 β 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 β 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.pred.p	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
#=============================================

#== Preambule
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 *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.*

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)
```

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 '~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).
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.
Vrho.start	Positive scalar indicating the starting value for the variance of the spatial random effects.

`mubeta`	Means of the priors for the β 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 β 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/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.

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`	Predictive posterior mean of the spatial random effect associated to each spatial cell.
`prob.pred.p`	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
#=============================================

#== Preambule
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)

#==========
```

```
#== 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)


## 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:

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

where

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

The generized 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