

# Package ‘PrevMap’

August 31, 2015

**Type** Package

**Title** Geostatistical Modelling of Spatially Referenced Prevalence Data

**Version** 1.2.2

**Date** 2015-08-31

**Author** Emanuele Giorgi, Peter J. Diggle

**Maintainer** Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

**Imports** splancs, truncnorm, methods

**Depends** geoR, maxLik, raster, pdist

**Description** It provides functions for both likelihood-based and Bayesian analysis of spatially referenced prevalence data, and is also an extension of the 'geoR' package.

**LazyData** true

**License** GPL (>= 2)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2015-08-31 15:17:43

## R topics documented:

adjust.sigma2 . . . . .	2
autocor.plot . . . . .	3
binary.probit.Bayes . . . . .	4
binomial.logistic.Bayes . . . . .	7
binomial.logistic.MCML . . . . .	10
coef.PrevMap . . . . .	14
continuous.sample . . . . .	15
contour.pred.PrevMap . . . . .	16
control.mcmc.Bayes . . . . .	17
control.mcmc.MCML . . . . .	18
control.prior . . . . .	19
control.profile . . . . .	21

create.ID.coords . . . . .	22
data_sim . . . . .	22
dens.plot . . . . .	23
discrete.sample . . . . .	24
Laplace.sampling . . . . .	25
Laplace.sampling.lr . . . . .	27
linear.model.Bayes . . . . .	29
linear.model.MLE . . . . .	32
loaloa . . . . .	34
loglik.ci . . . . .	35
loglik.linear.model . . . . .	36
matern.kernel . . . . .	37
plot.pred.PrevMap . . . . .	38
plot.profile.PrevMap . . . . .	38
plot.shape.matern . . . . .	39
shape.matern . . . . .	40
spatial.pred.binomial.Bayes . . . . .	41
spatial.pred.binomial.MCML . . . . .	42
spatial.pred.linear.Bayes . . . . .	44
spatial.pred.linear.MLE . . . . .	45
summary.Bayes.PrevMap . . . . .	47
summary.PrevMap . . . . .	48
trace.plot . . . . .	49
trace.plot.MCML . . . . .	49

**Index** **51**

---

adjust.sigma2	<i>Adjustment factor for the variance of the convolution of Gaussian noise</i>
---------------	--

---

**Description**

This function computes the multiplicative constant used to adjust the value of sigma2 in the low-rank approximation of a Gaussian process.

**Usage**

```
adjust.sigma2(knots.dist, phi, kappa)
```

**Arguments**

knots.dist	a matrix of the distances between the observed coordinates and the spatial knots.
phi	scale parameter of the Matern covariance function.
kappa	shape parameter of the Matern covariance function.

**Details**

Let  $U$  denote the  $n$  by  $m$  matrix of the distances between the  $n$  observed coordinates and  $m$  pre-defined spatial knots. This function computes the following quantity

$$\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m K(u_{ij}; \phi, \kappa)^2,$$

where  $K(\cdot; \phi, \kappa)$  is the Matern kernel (see [matern.kernel](#)) and  $u_{ij}$  is the distance between the  $i$ -th sampled location and the  $j$ -th spatial knot.

**Value**

A value corresponding to the adjustment factor for sigma2.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

**See Also**

[matern.kernel](#), [pdist](#).

**Examples**

```
set.seed(1234)
# Observed coordinates
n <- 100
coords <- cbind(runif(n),runif(n))

# Spatial knots
knots <- expand.grid(seq(-0.2,1.2,length=5),seq(-0.2,1.2,length=5))

# Distance matrix
knots.dist <- as.matrix(pdist(coords,knots))

adjust.sigma2(knots.dist,0.1,2)
```

---

autocor.plot

*Plot of the autocorrelgram for posterior samples*


---

**Description**

Plots the autocorrelogram for the posterior samples of the model parameters and spatial random effects.

**Usage**

```
autocor.plot(object, param, component.beta = NULL, component.S = NULL)
```

**Arguments**

object	an object of class 'Bayes.PrevMap'.
param	a character indicating for which component of the model the autocorrelation plot is required: param="beta" for the regression coefficients; param="sigma2" for the variance of the spatial random effect; param="phi" for the scale parameter of the Matern correlation function; param="tau2" for the variance of the nugget effect; param="S" for the spatial random effect.
component.beta	if param="beta", component.beta is a numeric value indicating the component of the regression coefficients; default is NULL.
component.S	if param="S", component.S can be a numeric value indicating the component of the spatial random effect, or set equal to "all" if the autocorrelgram should be plotted for all the components. Default is NULL.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

binary.probit.Bayes     *Bayesian estimation for the two-levels binary probit model*

---

**Description**

This function performs Bayesian estimation for a geostatistical binary probit model. It also allows to specify a two-levels model so as to include individual-level and household-level (or any other unit comprising a group of individuals, e.g. village, school, compound, etc...) variables.

**Usage**

```
binary.probit.Bayes(formula, coords, data, ID.coords, control.prior,
  control.mcmc, kappa, low.rank = FALSE, knots = NULL, messages = TRUE)
```

**Arguments**

formula	an object of class <a href="#">formula</a> (or one that can be coerced to that class): a symbolic description of the model to be fitted.
coords	an object of class <a href="#">formula</a> indicating the geographic coordinates.
data	a data frame containing the variables in the model.
ID.coords	vector of ID values for the unique set of spatial coordinates obtained from <a href="#">create.ID.coords</a> . These must be provided in order to specify spatial random effects at household-level. <b>Warning:</b> the household coordinates must all be distinct otherwise see <a href="#">jitterDupCoords</a> . Default is NULL.
control.prior	output from <a href="#">control.prior</a> .
control.mcmc	output from <a href="#">control.mcmc.Bayes</a> .

kappa	value for the shape parameter of the Matern covariance function.
low.rank	logical; if low.rank=TRUE a low-rank approximation is required. Default is low.rank=FALSE.
knots	if low.rank=TRUE, knots is a matrix of spatial knots used in the low-rank approximation. Default is knots=NULL.
messages	logical; if messages=TRUE then status messages are printed on the screen (or output device) while the function is running. Default is messages=TRUE.

## Details

This function performs Bayesian estimation for the parameters of the geostatistical binary probit model. Let  $i$  and  $j$  denote the indices of the  $i$ -th household and  $j$ -th individual within that household. The response variable  $Y_{ij}$  is a binary indicator taking value 1 if the individual has been tested positive for the disease of interest and 0 otherwise. Conditionally on a zero-mean stationary Gaussian process  $S(x_i)$ ,  $Y_{ij}$  are mutually independent Bernoulli variables with probit link function  $\Phi^{-1}(\cdot)$ , i.e.

$$\Phi^{-1}(p_{ij}) = d'_{ij}\beta + S(x_i),$$

where  $d_{ij}$  is a vector of covariates, both at individual- and household-level, with associated regression coefficients  $\beta$ . The Gaussian process  $S(x)$  has isotropic Matern covariance function (see [matern](#)) with variance `sigma2`, scale parameter `phi` and shape parameter `kappa`.

**Priors definition.** Priors can be defined through the function [control.prior](#). The hierarchical structure of the priors is the following. Let  $\theta$  be the vector of the covariance parameters `c(sigma2, phi)`; each component of  $\theta$  has independent priors that can be freely defined by the user. However, in [control.prior](#) uniform and log-normal priors are also available as default priors for each of the covariance parameters. The vector of regression coefficients  $\beta$  has a multivariate Gaussian prior with mean `beta.mean` and covariance matrix `beta.covar`.

**Updating regression coefficients and random effects using auxiliary variables.** To update  $\beta$  and  $S(x_i)$ , we use an auxiliary variable technique based on Rue and Held (2005). Let  $V_{ij}$  denote a set of random variables that conditionally on  $\beta$  and  $S(x_i)$ , are mutually independent Gaussian with mean  $d'_{ij}\beta + S(x_i)$  and unit variance. Then,  $Y_{ij} = 1$  if  $V_{ij} > 0$  and  $Y_{ij} = 0$  otherwise. Using this representation of the model, we use a Gibbs sampler to simulate from the full conditionals of  $\beta$ ,  $S(x_i)$  and  $V_{ij}$ . See Section 4.3 of Rue and Held (2005) for more details.

**Updating the covariance parameters with a Metropolis-Hastings algorithm.** In the MCMC algorithm implemented in `binary.probit.Bayes`, the transformed parameters

$$(\theta_1, \theta_2) = (\log(\sigma^2)/2, \log(\sigma^2/\phi^{2\kappa}))$$

are independently updated using a Metropolis Hastings algorithm. At the  $i$ -th iteration, a new value is proposed for each parameter from a univariate Gaussian distribution with variance  $h_i^2$ . This is tuned using the following adaptive scheme

$$h_i = h_{i-1} + c_1 i^{-c_2} (\alpha_i - 0.45),$$

where  $\alpha_i$  is the acceptance rate at the  $i$ -th iteration, 0.45 is the optimal acceptance rate for a univariate Gaussian distribution, whilst  $c_1 > 0$  and  $0 < c_2 < 1$  are pre-defined constants. The starting values  $h_1$  for each of the parameters  $\theta_1$  and  $\theta_2$  can be set using the function [control.mcmc.Bayes](#) through the arguments `h.theta1`, `h.theta2` and `h.theta3`. To define values for  $c_1$  and  $c_2$ , see the documentation of [control.mcmc.Bayes](#).

**Low-rank approximation.** In the case of very large spatial data-sets, a low-rank approximation of the Gaussian spatial process  $S(x)$  might be computationally beneficial. Let  $(x_1, \dots, x_m)$  and  $(t_1, \dots, t_m)$  denote the set of sampling locations and a grid of spatial knots covering the area of interest, respectively. Then  $S(x)$  is approximated as  $\sum_{i=1}^m K(\|x - t_i\|; \phi, \kappa) U_i$ , where  $U_i$  are zero-mean mutually independent Gaussian variables with variance `sigma2` and  $K(\cdot; \phi, \kappa)$  is the isotropic Matern kernel (see `matern.kernel`). Since the resulting approximation is no longer a stationary process (but only approximately), `sigma2` may take very different values from the actual variance of the Gaussian process to approximate. The function `adjust.sigma2` can then be used to (approximately) explore the range for `sigma2`. For example if the variance of the Gaussian process is 0.5, then an approximate value for `sigma2` is `0.5/const.sigma2`, where `const.sigma2` is the value obtained with `adjust.sigma2`.

### Value

An object of class "Bayes.PrevMap". The function `summary.Bayes.PrevMap` is used to print a summary of the fitted model. The object is a list with the following components:

`estimate`: matrix of the posterior samples of the model parameters.

`S`: matrix of the posterior samples for each component of the random effect.

`const.sigma2`: vector of the values of the multiplicative factor used to adjust the values of `sigma2` in the low-rank approximation.

`y`: binary observations.

`D`: matrix of covariates.

`coords`: matrix of the observed sampling locations.

`kappa`: shape parameter of the Matern function.

`ID.coords`: set of ID values defined through the argument `ID.coords`.

`knots`: matrix of spatial knots used in the low-rank approximation.

`h1`: vector of values taken by the tuning parameter `h.theta1` at each iteration.

`h2`: vector of values taken by the tuning parameter `h.theta2` at each iteration.

`call`: the matched call.

### Author(s)

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

### References

Rue, H., Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall, London.

Higdon, D. (1998). *A process-convolution approach to modeling temperatures in the North Atlantic Ocean*. Environmental and Ecological Statistics 5, 173-190.

### See Also

`control.mcmc.Bayes`, `control.prior`, `summary.Bayes.PrevMap`, `matern`, `matern.kernel`, `create.ID.coords`.

---

binomial.logistic.Bayes

*Bayesian estimation for the binomial logistic model*


---

## Description

This function performs Bayesian estimation for a geostatistical binomial logistic model.

## Usage

```
binomial.logistic.Bayes(formula, units.m, coords, data, ID.coords = NULL,
  control.prior, control.mcmc, kappa, low.rank = FALSE, knots = NULL,
  messages = TRUE)
```

## Arguments

formula	an object of class <a href="#">formula</a> (or one that can be coerced to that class): a symbolic description of the model to be fitted.
units.m	an object of class <a href="#">formula</a> indicating the binomial denominators.
coords	an object of class <a href="#">formula</a> indicating the geographic coordinates.
data	a data frame containing the variables in the model.
ID.coords	vector of ID values for the unique set of spatial coordinates obtained from <a href="#">create.ID.coords</a> . These must be provided if, for example, spatial random effects are defined at household level but some of the covariates are at individual level. <b>Warning:</b> the household coordinates must all be distinct otherwise see <a href="#">jitterDupCoords</a> . Default is NULL.
control.prior	output from <a href="#">control.prior</a> .
control.mcmc	output from <a href="#">control.mcmc.Bayes</a> .
kappa	value for the shape parameter of the Matern covariance function.
low.rank	logical; if low.rank=TRUE a low-rank approximation is required. Default is low.rank=FALSE.
knots	if low.rank=TRUE, knots is a matrix of spatial knots used in the low-rank approximation. Default is knots=NULL.
messages	logical; if messages=TRUE then status messages are printed on the screen (or output device) while the function is running. Default is messages=TRUE.

## Details

This function performs Bayesian estimation for the parameters of the geostatistical binomial logistic model. Conditionally on a zero-mean stationary Gaussian process  $S(x)$  and mutually independent zero-mean Gaussian variables  $Z$  with variance  $\tau_2$ , the linear predictor assumes the form

$$\log(p/(1-p)) = d'\beta + S(x) + Z,$$

where  $d$  is a vector of covariates with associated regression coefficients  $\beta$ . The Gaussian process  $S(x)$  has isotropic Matern covariance function (see `matern`) with variance `sigma2`, scale parameter `phi` and shape parameter `kappa`.

**Priors definition.** Priors can be defined through the function `control.prior`. The hierarchical structure of the priors is the following. Let  $\theta$  be the vector of the covariance parameters `c(sigma2, phi, tau2)`; then each component of  $\theta$  has independent priors freely defined by the user. However, in `control.prior` uniform and log-normal priors are also available as default priors for each of the covariance parameters. To remove the nugget effect  $Z$ , no prior should be defined for `tau2`. Conditionally on `sigma2`, the vector of regression coefficients  $\beta$  has a multivariate Gaussian prior with mean `beta.mean` and covariance matrix `sigma2*beta.covar`, while in the low-rank approximation the covariance matrix is simply `beta.covar`.

**Updating the covariance parameters with a Metropolis-Hastings algorithm.** In the MCMC algorithm implemented in `binomial.logistic.Bayes`, the transformed parameters

$$(\theta_1, \theta_2, \theta_3) = (\log(\sigma^2)/2, \log(\sigma^2/\phi^{2\kappa}), \log(\tau^2))$$

are independently updated using a Metropolis Hastings algorithm. At the  $i$ -th iteration, a new value is proposed for each from a univariate Gaussian distribution with variance  $h_i^2$  that is tuned using the following adaptive scheme

$$h_i = h_{i-1} + c_1 i^{-c_2} (\alpha_i - 0.45),$$

where  $\alpha_i$  is the acceptance rate at the  $i$ -th iteration, 0.45 is the optimal acceptance rate for a univariate Gaussian distribution, whilst  $c_1 > 0$  and  $0 < c_2 < 1$  are pre-defined constants. The starting values  $h_1$  for each of the parameters  $\theta_1$ ,  $\theta_2$  and  $\theta_3$  can be set using the function `control.mcmc.Bayes` through the arguments `h.theta1`, `h.theta2` and `h.theta3`. To define values for  $c_1$  and  $c_2$ , see the documentation of `control.mcmc.Bayes`.

**Hamiltonian Monte Carlo.** The MCMC algorithm in `binomial.logistic.Bayes` uses a Hamiltonian Monte Carlo (HMC) procedure to update the random effect  $T = d'\beta + S(x) + Z$ ; see Neal (2011) for an introduction to HMC. HMC makes use of a position vector, say  $t$ , representing the random effect  $T$ , and a momentum vector, say  $q$ , of the same length of the position vector, say  $n$ . Hamiltonian dynamics also have a physical interpretation where the states of the system are described by the position of a puck and its momentum (its mass times its velocity). The Hamiltonian function is then defined as a function of  $t$  and  $q$ , having the form  $H(t, q) = -\log\{f(t|y, \beta, \theta)\} + q'q/2$ , where  $f(t|y, \beta, \theta)$  is the conditional distribution of  $T$  given the data  $y$ , the regression parameters  $\beta$  and covariance parameters  $\theta$ . The system of Hamiltonian equations then defines the evolution of the system in time, which can be used to implement an algorithm for simulation from the posterior distribution of  $T$ . In order to implement the Hamiltonian dynamic on a computer, the Hamiltonian equations must be discretised. The *leapfrog method* is then used for this purpose, where two tuning parameters should be defined: the stepsize  $\epsilon$  and the number of steps  $L$ . These respectively correspond to `epsilon.S.lim` and `L.S.lim` in the `control.mcmc.Bayes` function. However, it is advisable to let *epsilon* and  $L$  take different random values at each iteration of the HCM algorithm so as to account for the different variances amongst the components of the posterior of  $T$ . This can be done in `control.mcmc.Bayes` by defining `epsilon.S.lim` and `L.S.lim` as vectors of two elements, each of which represents the lower and upper limit of a uniform distribution used to generate values for `epsilon.S.lim` and `L.S.lim`, respectively.

**Using a two-level model to include household-level and individual-level information.** When analysing data from household surveys, some of the available information might be at household-level (e.g. material of house, temperature) and some at individual-level (e.g. age,



gender). In this case, the Gaussian spatial process  $S(x)$  and the nugget effect  $Z$  are defined at household-level in order to account for extra-binomial variation between and within households, respectively.

**Low-rank approximation.** In the case of very large spatial data-sets, a low-rank approximation of the Gaussian spatial process  $S(x)$  might be computationally beneficial. Let  $(x_1, \dots, x_m)$  and  $(t_1, \dots, t_m)$  denote the set of sampling locations and a grid of spatial knots covering the area of interest, respectively. Then  $S(x)$  is approximated as  $\sum_{i=1}^m K(\|x - t_i\|; \phi, \kappa) U_i$ , where  $U_i$  are zero-mean mutually independent Gaussian variables with variance `sigma2` and  $K(\cdot; \phi, \kappa)$  is the isotropic Matern kernel (see [matern.kernel](#)). Since the resulting approximation is no longer a stationary process (but only approximately), `sigma2` may take very different values from the actual variance of the Gaussian process to approximate. The function `adjust.sigma2` can then be used to (approximately) explore the range for `sigma2`. For example if the variance of the Gaussian process is 0.5, then an approximate value for `sigma2` is `0.5/const.sigma2`, where `const.sigma2` is the value obtained with `adjust.sigma2`.

## Value

An object of class "Bayes.PrevMap". The function `summary.Bayes.PrevMap` is used to print a summary of the fitted model. The object is a list with the following components:

`estimate`: matrix of the posterior samples of the model parameters.

`S`: matrix of the posterior samples for each component of the random effect.

`const.sigma2`: vector of the values of the multiplicative factor used to adjust the values of `sigma2` in the low-rank approximation.

`y`: binomial observations.

`units.m`: binomial denominators.

`D`: matrix of covariates.

`coords`: matrix of the observed sampling locations.

`kappa`: shape parameter of the Matern function.

`ID.coords`: set of ID values defined through the argument `ID.coords`.

`knots`: matrix of spatial knots used in the low-rank approximation.

`h1`: vector of values taken by the tuning parameter `h.theta1` at each iteration.

`h2`: vector of values taken by the tuning parameter `h.theta2` at each iteration.

`h3`: vector of values taken by the tuning parameter `h.theta3` at each iteration.

`call`: the matched call.

## Author(s)

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

## References

Neal, R. M. (2011) *MCMC using Hamiltonian Dynamics*, In: Handbook of Markov Chain Monte Carlo (Chapter 5), Edited by Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng Chapman & Hall / CRC Press.

Higdon, D. (1998). *A process-convolution approach to modeling temperatures in the North Atlantic Ocean*. Environmental and Ecological Statistics 5, 173-190.

## See Also

[control.mcmc.Bayes](#), [control.prior](#), [summary.Bayes.PrevMap](#), [matern](#), [matern.kernel](#), [create.ID.coords](#).

## Examples

```
set.seed(1234)
data(data_sim)
# Select a subset of data_sim with 50 observations
n.subset <- 50
data_subset <- data_sim[sample(1:nrow(data_sim),n.subset),]
# Set the MCMC control parameters
control.mcmc <- control.mcmc.Bayes(n.sim=10,burnin=0,thin=1,
                                   h.theta1=0.05,h.theta2=0.05,
                                   L.S.lim=c(1,50),epsilon.S.lim=c(0.01,0.02),
                                   start.beta=0,start.sigma2=1,start.phi=0.15,
                                   start.S=rep(0,n.subset))

cp <- control.prior(beta.mean=0,beta.covar=1,
                   log.normal.phi=c(log(0.15),0.05),
                   log.normal.sigma2=c(log(1),0.1))

fit.Bayes <- binomial.logistic.Bayes(formula=y~1,coords=~x1+x2,units.m=~units.m,
                                     data=data_subset,control.prior=cp,
                                     control.mcmc=control.mcmc,kappa=2)

summary(fit.Bayes)

par(mfrow=c(2,4))
autocor.plot(fit.Bayes,param="S",component.S="all")
autocor.plot(fit.Bayes,param="beta",component.beta=1)
autocor.plot(fit.Bayes,param="sigma2")
autocor.plot(fit.Bayes,param="phi")
trace.plot(fit.Bayes,param="S",component.S=30)
trace.plot(fit.Bayes,param="beta",component.beta=1)
trace.plot(fit.Bayes,param="sigma2")
trace.plot(fit.Bayes,param="phi")
```

---

binomial.logistic.MCML

*Monte Carlo Maximum Likelihood estimation for the binomial logistic model*

---

**Description**

This function performs Monte Carlo maximum likelihood (MCML) estimation for the geostatistical binomial logistic model.

**Usage**

```
binomial.logistic.MCML(formula, units.m, coords, data, ID.coords = NULL, par0,
  control.mcmc, kappa, fixed.rel.nugget = NULL, start.cov.pars,
  method = "BFGS", low.rank = FALSE, knots = NULL, messages = TRUE,
  plot.correlogram = TRUE)
```

**Arguments**

formula	an object of class <code>formula</code> (or one that can be coerced to that class): a symbolic description of the model to be fitted.
units.m	an object of class <code>formula</code> indicating the binomial denominators.
coords	an object of class <code>formula</code> indicating the geographic coordinates.
data	a data frame containing the variables in the model.
ID.coords	vector of ID values for the unique set of spatial coordinates obtained from <code>create.ID.coords</code> . These must be provided if, for example, spatial random effects are defined at household level but some of the covariates are at individual level. <b>Warning:</b> the household coordinates must all be distinct otherwise see <code>jitterDupCoords</code> . Default is <code>NULL</code> .
par0	parameters of the importance sampling distribution: these should be given in the following order <code>c(beta, sigma2, phi, tau2)</code> , where <code>beta</code> are the regression coefficients, <code>sigma2</code> is the variance of the Gaussian process, <code>phi</code> is the scale parameter of the spatial correlation and <code>tau2</code> is the variance of the nugget effect (if included in the model).
control.mcmc	output from <code>control.mcmc.MCML</code> .
kappa	fixed value for the shape parameter of the Matern covariance function.
fixed.rel.nugget	fixed value for the relative variance of the nugget effect; <code>fixed.rel.nugget=NULL</code> if this should be included in the estimation. Default is <code>fixed.rel.nugget=NULL</code> .
start.cov.pars	a vector of length two with elements corresponding to the starting values of <code>phi</code> and the relative variance of the nugget effect <code>nu2</code> , respectively, that are used in the optimization algorithm. If <code>nu2</code> is fixed through <code>fixed.rel.nugget</code> , then <code>start.cov.pars</code> represents the starting value for <code>phi</code> only.
method	method of optimization. If <code>method="BFGS"</code> then the <code>maxBFGS</code> function is used; otherwise <code>method="nllminb"</code> to use the <code>nllminb</code> function. Default is <code>method="BFGS"</code> .
low.rank	logical; if <code>low.rank=TRUE</code> a low-rank approximation of the Gaussian spatial process is used when fitting the model. Default is <code>low.rank=FALSE</code> .
knots	if <code>low.rank=TRUE</code> , <code>knots</code> is a matrix of spatial knots that are used in the low-rank approximation. Default is <code>knots=NULL</code> .
messages	logical; if <code>messages=TRUE</code> then status messages are printed on the screen (or output device) while the function is running. Default is <code>messages=TRUE</code> .

plot.correlogram

logical; if plot.correlogram=TRUE the autocorrelation plot of the samples of the random effect is displayed after completion of conditional simulation. Default is plot.correlogram=TRUE.

## Details

This function performs parameter estimation for a geostatistical binomial logistic model. Conditionally on a zero-mean stationary Gaussian process  $S(x)$  and mutually independent zero-mean Gaussian variables  $Z$  with variance tau2, the observations  $y$  are generated from a binomial distribution with probability  $p$  and binomial denominators units.m. A canonical logistic link is used, thus the linear predictor assumes the form

$$\log(p/(1-p)) = d'\beta + S(x) + Z,$$

where  $d$  is a vector of covariates with associated regression coefficients  $\beta$ . The Gaussian process  $S(x)$  has isotropic Matern covariance function (see [matern](#)) with variance sigma2, scale parameter phi and shape parameter kappa. In the binomial.logistic.MCML function, the shape parameter is treated as fixed. The relative variance of the nugget effect, nu2=tau2/sigma2, can also be fixed through the argument fixed.rel.nugget; if fixed.rel.nugget=NULL, then the relative variance of the nugget effect is also included in the estimation.

**Monte Carlo Maximum likelihood.** The Monte Carlo maximum likelihood method uses conditional simulation from the distribution of the random effect  $T(x) = d(x)'\beta + S(x) + Z$  given the data  $y$ , in order to approximate the high-dimensional intractable integral given by the likelihood function. The resulting approximation of the likelihood is then maximized by a numerical optimization algorithm which uses analytic expression for computation of the gradient vector and Hessian matrix. The functions used for numerical optimization are [maxBFGS](#) (method="BFGS"), from the [maxLik](#) package, and [nlminb](#) (method="nlminb").

**Using a two-level model to include household-level and individual-level information.** When analysing data from household surveys, some of the available information might be at household-level (e.g. material of house, temperature) and some at individual-level (e.g. age, gender). In this case, the Gaussian spatial process  $S(x)$  and the nugget effect  $Z$  are defined at household-level in order to account for extra-binomial variation between and within households, respectively.

**Low-rank approximation.** In the case of very large spatial data-sets, a low-rank approximation of the Gaussian spatial process  $S(x)$  might be computationally beneficial. Let  $(x_1, \dots, x_m)$  and  $(t_1, \dots, t_m)$  denote the set of sampling locations and a grid of spatial knots covering the area of interest, respectively. Then  $S(x)$  is approximated as  $\sum_{i=1}^m K(\|x - t_i\|; \phi, \kappa) U_i$ , where  $U_i$  are zero-mean mutually independent Gaussian variables with variance sigma2 and  $K(\cdot; \phi, \kappa)$  is the isotropic Matern kernel (see [matern.kernel](#)). Since the resulting approximation is no longer a stationary process (but only approximately), the parameter sigma2 is then multiplied by a factor constant.sigma2 so as to obtain a value that is closer to the actual variance of  $S(x)$ .

## Value

An object of class "PrevMap". The function [summary.PrevMap](#) is used to print a summary of the fitted model. The object is a list with the following components:

estimate: estimates of the model parameters; use the function [coef.PrevMap](#) to obtain estimates of covariance parameters on the original scale.

covariance: covariance matrix of the MCML estimates.  
 log.lik: maximum value of the log-likelihood.  
 y: binomial observations.  
 units.m: binomial denominators.  
 D: matrix of covariates.  
 coords: matrix of the observed sampling locations.  
 method: method of optimization used.  
 ID.coords: set of ID values defined through the argument ID.coords.  
 kappa: fixed value of the shape parameter of the Matern function.  
 knots: matrix of the spatial knots used in the low-rank approximation.  
 const.sigma2: adjustment factor for sigma2 in the low-rank approximation.  
 h: vector of the values of the tuning parameter at each iteration of the Langevin-Hastings MCMC algorithm; see [Laplace.sampling](#), or [Laplace.sampling.lr](#) if a low-rank approximation is used.  
 samples: matrix of the random effects samples from the importance sampling distribution used to approximate the likelihood function.  
 fixed.rel.nugget: fixed value for the relative variance of the nugget effect.  
 call: the matched call.

### Author(s)

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>  
 Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

### References

Christensen, O. F. (2004). *Monte carlo maximum likelihood in model-based geostatistics*. Journal of Computational and Graphical Statistics 13, 702-718.  
 Higdon, D. (1998). *A process-convolution approach to modeling temperatures in the North Atlantic Ocean*. Environmental and Ecological Statistics 5, 173-190.

### See Also

[Laplace.sampling](#), [Laplace.sampling.lr](#), [summary.PrevMap](#), [coef.PrevMap](#), [matern](#), [matern.kernel](#), [control.mcmc.MCML](#), [create.ID.coords](#).

### Examples

```
set.seed(1234)
data(data_sim)
# Select a subset of data_sim with 50 observations
n.subset <- 10
data_subset <- data_sim[sample(1:nrow(data_sim),n.subset),]

# Set the MCMC control parameters
control.mcmc <- control.mcmc.MCML(n.sim=1000,burnin=0,thin=1,
```

```
h=1.65/(n.subset^2/3))

# Set the parameters of the importance sampling distribution
par0 <- c(0,1,0.15)

# Estimate the model parameters using MCML
fit.MCML <- binomial.logistic.MCML(y~1, units.m=~units.m, coords=~x1+x2,
data=data_subset, control.mcmc=control.mcmc,
fixed.rel.nugget=0,par0=par0,start.cov.pars=0.15,method="nlminb",kappa=2)
summary(fit.MCML,log.cov.pars=FALSE)
coef(fit.MCML)
```

---

coef.PrevMap

*Extract model coefficients*

---

### Description

coef extracts parameters estimates from models fitted with the functions [linear.model.MLE](#) and [binomial.logistic.MCML](#).

### Usage

```
## S3 method for class 'PrevMap'
coef(object, ...)
```

### Arguments

object            an object of class "PrevMap".  
...                other arguments.

### Value

coefficients extracted from the model object object.

### Author(s)

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

---

continuous.sample      *Spatially continuous sampling*

---

### Description

Draws a sample of spatial locations within a spatially continuous polygonal sampling region.

### Usage

```
continuous.sample(poly, n, delta, k = 0, rho = NULL)
```

### Arguments

poly	boundary of a polygon.
n	number of events.
delta	minimum permissible distance between any two events in preliminary sample.
k	number of locations in preliminary sample to be replaced by near neighbours of other preliminary sample locations in final sample (must be between 0 and n/2)
rho	maximum distance between close pairs of locations in final sample.

### Details

To draw a sample of size  $n$  from a spatially continuous region  $A$ , with the property that the distance between any two sampled locations is at least  $\delta$ , the following algorithm is used.

- Step 1. Set  $i = 1$  and generate a point  $x_1$  uniformly distributed on  $A$ .
- Step 2. Increase  $i$  by 1, generate a point  $x_i$  uniformly distributed on  $A$  and calculate the minimum,  $d_{\min}$ , of the distances from  $x_i$  to all  $x_j : j < i$ .
- Step 3. If  $d_{\min} \geq \delta$ , increase  $i$  by 1 and return to step 2 if  $i \leq n$ , otherwise stop;
- Step 4. If  $d_{\min} < \delta$ , return to step 2 without increasing  $i$ .

**Sampling close pairs of points.** For some purposes, it is desirable that a spatial sampling scheme include pairs of closely spaced points. In this case, the above algorithm requires the following additional steps to be taken. Let  $k$  be the required number of close pairs. Choose a value  $\rho$  such that a close pair of points will be a pair of points separated by a distance of at most  $\rho$ .

- Step 5. Set  $j = 1$  and draw a random sample of size 2 from the integers  $1, 2, \dots, n$ , say  $(i_1; i_2)$ ;
- Step 6. Replace  $x_{i_1}$  by  $x_{i_2} + u$ , where  $u$  is uniformly distributed on the disc with centre  $x_{i_2}$  and radius  $\rho$ , increase  $i$  by 1 and return to step 5 if  $i \leq k$ , otherwise stop.

### Value

A matrix of dimension  $n$  by 2 containing event locations.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

**Examples**

```
library(geoR)
data(parana)
poly<-parana$borders
poly<-matrix(c(poly[,1],poly[,2]),dim(poly)[1],2,byrow=FALSE)
set.seed(5871121)

# Generate spatially regular sample
xy.sample<-continuous.sample(poly,100,30)
plot(poly,type="l",xlab="X",ylab="Y")
points(xy.sample,pch=19,cex=0.5)
```

---

contour.pred.PrevMap *Contour plot of a predicted surface*

---

**Description**

plot.pred.PrevMap displays contours of predictions obtained from [spatial.pred.linear.MLE](#), [spatial.pred.linear.Bayes](#), [spatial.pred.binomial.MCML](#) and [spatial.pred.binomial.Bayes](#).

**Usage**

```
## S3 method for class 'pred.PrevMap'
contour(x, type = NULL, summary = "predictions", ...)
```

**Arguments**

x	an object of class "pred.PrevMap".
type	a character indicating the type of prediction to display: 'prevalence', 'odds', 'logit' or 'probit'.
summary	character indicating which summary to display: 'predictions', 'quantiles', 'standard.errors' or 'exceedance.prob'; default is 'predictions'. If summary="exceedance.prob", the argument type is ignored.
...	further arguments passed to <a href="#">contour</a> .

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>



---

control.mcmc.Bayes      *Control settings for the MCMC algorithm used for Bayesian inference*

---

### Description

This function defines the different tuning parameter that are used in the MCMC algorithm for Bayesian inference.

### Usage

```
control.mcmc.Bayes(n.sim, burnin, thin, h.theta1, h.theta2, h.theta3 = NULL,
  L.S.lim = NULL, epsilon.S.lim = NULL, start.beta, start.sigma2, start.phi,
  start.S, start.nugget = NULL, c1.h.theta1 = 0.01, c2.h.theta1 = 1e-04,
  c1.h.theta2 = 0.01, c2.h.theta2 = 1e-04, c1.h.theta3 = 0.01,
  c2.h.theta3 = 1e-04, linear.model = FALSE, binary = FALSE)
```

### Arguments

n.sim	total number of simulations.
burnin	initial number of samples to be discarded.
thin	value used to retain only every thin-th sampled value.
h.theta1	starting value of the tuning parameter of the proposal distribution for $\theta_1 = \log(\sigma^2)/2$ . See 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
h.theta2	starting value of the tuning parameter of the proposal distribution for $\theta_2 = \log(\sigma^2/\phi^{2\kappa})$ . See 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
h.theta3	starting value of the tuning parameter of the proposal distribution for $\theta_3 = \log(\tau^2)$ . See 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
L.S.lim	an atomic value or a vector of length 2 that is used to define the number of steps used at each iteration in the Hamiltonian Monte Carlo algorithm to update the spatial random effect; if a single value is provided than the number of steps is kept fixed, otherwise if a vector of length 2 is provided the number of steps is simulated at each iteration as <code>floor(runif(1,L.S.lim[1],L.S.lim[2]+1))</code> .
epsilon.S.lim	an atomic value or a vector of length 2 that is used to define the stepsize used at each iteration in the Hamiltonian Monte Carlo algorithm to update the spatial random effect; if a single value is provided than the stepsize is kept fixed, otherwise if a vector of length 2 is provided the stepsize is simulated at each iteration as <code>runif(1,epsilon.S.lim[1],epsilon.S.lim[2])</code> .
start.beta	starting value for the regression coefficients beta.
start.sigma2	starting value for sigma2.
start.phi	starting value for phi.
start.S	starting value for the spatial random effect.
start.nugget	starting value for the variance of the nugget effect; default is NULL if the nugget effect is not present.

c1.h.theta1	value of $c_1$ used to adaptively tune the variance of the Gaussian proposal for the transformed parameter $\log(\text{sigma2})/2$ ; see 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
c2.h.theta1	value of $c_2$ used to adaptively tune the variance of the Gaussian proposal for the transformed parameter $\log(\text{sigma2})/2$ ; see 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
c1.h.theta2	value of $c_1$ used to adaptively tune the variance of the Gaussian proposal for the transformed parameter $\log(\text{sigma2.curr}/(\text{phi.curr}^{(2*\text{kappa})}))$ ; see 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
c2.h.theta2	value of $c_2$ used to adaptively tune the variance of the Gaussian proposal for the transformed parameter $\log(\text{sigma2.curr}/(\text{phi.curr}^{(2*\text{kappa})}))$ ; see 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
c1.h.theta3	value of $c_1$ used to adaptively tune the variance of the Gaussian proposal for the transformed parameter $\log(\text{tau2})$ ; see 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
c2.h.theta3	value of $c_2$ used to adaptively tune the variance of the Gaussian proposal for the transformed parameter $\log(\text{tau2})$ ; see 'Details' in <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
linear.model	logical; if <code>linear.model=TRUE</code> , the control parameters are set for the geostatistical linear model. Default is <code>linear.model=FALSE</code> .
binary	logical; if <code>binary=TRUE</code> , the control parameters are set the binary geostatistical model. Default is <code>binary=FALSE</code> .

**Value**

an object of class "mcmc.Bayes.PrevMap".

**Author(s)**

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

---

control.mcmc.MCML	<i>Control settings for the MCMC algorithm used for classical inference on a binomial logistic model</i>
-------------------	--

---

**Description**

This function defines the options for the MCMC algorithm used in the Monte Carlo maximum likelihood method.

**Usage**

```
control.mcmc.MCML(n.sim, burnin, thin = 1, h, c1.h = 0.01, c2.h = 1e-04)
```

**Arguments**

n.sim	number of simulations.
burnin	length of the burn-in period.
thin	only every thin iterations, a sample is stored; default is thin=1.
h	tuning parameter of the proposal distribution; default is h=0.05.
c1.h	value of $c_1$ used in the adaptive scheme for h; default is c1.h=0.01. See also 'Details' in <a href="#">binomial.logistic.MCML</a>
c2.h	value of $c_2$ used in the adaptive scheme for h; default is c1.h=0.01. See also 'Details' in <a href="#">binomial.logistic.MCML</a>

**Value**

A list with processed arguments to be passed to the main function.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

**Examples**

```
control.mcmc <- control.mcmc.MCML(n.sim=1000,burnin=100,thin=1,h=0.05)
str(control.mcmc)
```

---

control.prior	<i>Priors specification</i>
---------------	-----------------------------

---

**Description**

This function is used to define priors for the model parameters of a Bayesian geostatistical model.

**Usage**

```
control.prior(beta.mean, beta.covar, log.prior.sigma2 = NULL,
  log.prior.phi = NULL, log.prior.nugget = NULL, uniform.sigma2 = NULL,
  log.normal.sigma2 = NULL, uniform.phi = NULL, log.normal.phi = NULL,
  uniform.nugget = NULL, log.normal.nugget = NULL)
```

**Arguments**

beta.mean	mean vector of the Gaussian prior for the regression coefficients.
beta.covar	covariance matrix of the Gaussian prior for the regression coefficients.

- `log.prior.sigma2` a function corresponding to the log-density of the prior distribution for the variance `sigma2` of the Gaussian process. **Warning:** if a low-rank approximation is used, then `sigma2` corresponds to the variance of the iid zero-mean Gaussian variables. Default is NULL.
- `log.prior.phi` a function corresponding to the log-density of the prior distribution for the scale parameter of the Matern correlation function; default is NULL.
- `log.prior.nugget` optional: a function corresponding to the log-density of the prior distribution for the variance of the nugget effect; default is NULL with no nugget incorporated in the model; default is NULL.
- `uniform.sigma2` a vector of length two, corresponding to the lower and upper limit of the uniform prior on `sigma2`. Default is NULL.
- `log.normal.sigma2` a vector of length two, corresponding to the mean and standard deviation of the distribution on the log scale for the log-normal prior on `sigma2`. Default is NULL.
- `uniform.phi` a vector of length two, corresponding to the lower and upper limit of the uniform prior on `phi`. Default is NULL.
- `log.normal.phi` a vector of length two, corresponding to the mean and standard deviation of the distribution on the log scale for the log-normal prior on `phi`. Default is NULL.
- `uniform.nugget` a vector of length two, corresponding to the lower and upper limit of the uniform prior on `tau2`. Default is NULL.
- `log.normal.nugget` a vector of length two, corresponding to the mean and standard deviation of the distribution on the log scale for the log-normal prior on `tau2`. Default is NULL.

**Value**

a list corresponding the prior distributions for each model parameter.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

**See Also**

See "Priors definition" in the Details section of the [binomial.logistic.Bayes](#) function.

---

control.profile	<i>Auxiliary function for controlling profile log-likelihood in the linear Gaussian model</i>
-----------------	---

---

**Description**

Auxiliary function used by [loglik.linear.model](#). This function defines whether the profile-loglikelihood should be computed or evaluation of the likelihood is required by keeping the other parameters fixed.

**Usage**

```
control.profile(phi = NULL, rel.nugget = NULL, fixed.beta = NULL,
               fixed.sigma2 = NULL, fixed.phi = NULL, fixed.rel.nugget = NULL)
```

**Arguments**

phi	a vector of the different values that should be used in the likelihood evaluation for the scale parameter phi, or NULL if a single value is provided either as first argument in <code>start.par</code> (for profile likelihood maximization) or as fixed value in <code>fixed.phi</code> ; default is NULL.
rel.nugget	a vector of the different values that should be used in the likelihood evaluation for the relative variance of the nugget effect nu2, or NULL if a single value is provided either in <code>start.par</code> (for profile likelihood maximization) or as fixed value in <code>fixed.nu2</code> ; default is NULL.
fixed.beta	a vector for the fixed values of the regression coefficients beta, or NULL if profile log-likelihood is to be performed; default is NULL.
fixed.sigma2	value for the fixed variance of the Gaussian process sigma2, or NULL if profile log-likelihood is to be performed; default is NULL.
fixed.phi	value for the fixed scale parameter phi in the Matern function, or NULL if profile log-likelihood is to be performed; default is NULL.
fixed.rel.nugget	value for the fixed relative variance of the nugget effect; <code>fixed.rel.nugget=NULL</code> if profile log-likelihood is to be performed; default is NULL.

**Value**

A list with components named as the arguments.

**Author(s)**

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>  
 Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

**See Also**

[loglik.linear.model](#)

---

create.ID.coords      *ID spatial coordinates*

---

### Description

Creates ID values for the unique set of coordinates.

### Usage

```
create.ID.coords(data, coords)
```

### Arguments

data                    a data frame containing the spatial coordinates.  
 coords                 an object of class `formula` indicating the geographic coordinates.

### Value

a vector of integers indicating the corresponding rows in data for each distinct coordinate obtained with the `unique` function.

### Author(s)

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>  
 Peter J. Diggle <p.diggle@lancaster.ac.uk>

### Examples

```
x1 <- runif(5)
x2 <- runif(5)
data <- data.frame(x1=rep(x1,each=3),x2=rep(x2,each=3))
ID.coords <- create.ID.coords(data,coords=~x1+x2)
data[,c("x1", "x2")] = unique(data[,c("x1", "x2")])[ID.coords,]
```

---

data\_sim                    *Simulated binomial data-set over the unit square*

---

### Description

This binomial data-set was simulated by generating a zero-mean Gaussian process over a 30 by 30 grid covering the unit square. The parameters used in the simulation are  $\sigma^2=1$ ,  $\phi=0.15$  and  $\kappa=2$ . The nugget effect was not included, hence  $\tau^2=0$ . The variables are as follows:

- y binomial observations.
- units.m binomial denominators.
- x1 horizontal coordinates.
- x2 vertical coordinates.
- S simulated values of the Gaussian process.

**Usage**

```
data(data_sim)
```

**Format**

A data frame with 900 rows and 5 variables

---

dens.plot	<i>Density plot for posterior samples</i>
-----------	---

---

**Description**

Plots the autocorrelogram for the posterior samples of the model parameters and spatial random effects.

**Usage**

```
dens.plot(object, param, component.beta = NULL, component.S = NULL,
          hist = TRUE, ...)
```

**Arguments**

object	an object of class 'Bayes.PrevMap'.
param	a character indicating for which component of the model the density plot is required: param="beta" for the regression coefficients; param="sigma2" for the variance of the spatial random effect; param="phi" for the scale parameter of the Matern correlation function; param="tau2" for the variance of the nugget effect; param="S" for the spatial random effect.
component.beta	if param="beta", component.beta is a numeric value indicating the component of the regression coefficients; default is NULL.
component.S	if param="S", component.S can be a numeric value indicating the component of the spatial random effect. Default is NULL.
hist	logical; if TRUE a histogram is added to density plot.
...	additional parameters to pass to <a href="#">density</a> .

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

discrete.sample      *Spatially discrete sampling*

---

### Description

Draws a sub-sample from a set of units spatially located irregularly over some defined geographical region by imposing a minimum distance between any two sampled units.

### Usage

```
discrete.sample(xy.all, n, delta, k = 0)
```

### Arguments

xy.all	set of locations from which the sample will be drawn.
n	size of required sample.
delta	minimum distance between any two locations in preliminary sample.
k	number of locations in preliminary sample to be replaced by nearest neighbours of other preliminary sample locations in final sample (must be between 0 and n/2).

### Details

To draw a sample of size  $n$  from a population of spatial locations  $X_i : i = 1, \dots, N$ , with the property that the distance between any two sampled locations is at least  $\delta$ , the function implements the following algorithm.

- Step 1. Draw an initial sample of size  $n$  completely at random and call this  $x_i : i = 1, \dots, n$ .
- Step 2. Set  $i = 1$  and calculate the minimum,  $d_{\min}$ , of the distances from  $x_i$  to all other  $x_j$  in the initial sample.
- Step 3. If  $d_{\min} \geq \delta$ , increase  $i$  by 1 and return to step 2 if  $i \leq n$ , otherwise stop.
- Step 4. If  $d_{\min} < \delta$ , draw an integer  $j$  at random from  $1, 2, \dots, N$ , set  $x_i = X_j$  and return to step 3.

Samples generated in this way will exhibit a more regular spatial arrangement than would a random sample of the same size. The degree of regularity achievable will be influenced by the spatial arrangement of the population  $X_i : i = 1, \dots, N$ , the specified value of  $\delta$  and the sample size  $n$ . For any given population, if  $n$  and/or  $\delta$  are too large, a sample of the required size with the distance between any two sampled locations at least  $\delta$  will not be achievable; the suggested solution is then to run the algorithm with a smaller value of  $\delta$ .

**Sampling close pairs of points.** For some purposes, it is desirable that a spatial sampling scheme include pairs of closely spaced points. In this case, the above algorithm requires the following additional steps to be taken. Let  $k$  be the required number of close pairs.

- Step 5. Set  $j = 1$  and draw a random sample of size 2 from the integers  $1, 2, \dots, n$ , say  $(i_1, i_2)$ .



- Step 6. Find the integer  $r$  such that the distances from  $x_{i_1}$  to  $X_r$  is the minimum of all  $N - 1$  distances from  $x_{i_1}$  to the  $X_j$ .
- Step 7. Replace  $x_{i_2}$  by  $X_r$ , increase  $i$  by 1 and return to step 5 if  $i \leq k$ , otherwise stop.

**Value**

A matrix of dimension  $n$  by 2 containing the final sampled locations.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

**Examples**

```
x<-0.015+0.03*(1:33)
xall<-rep(x,33)
yall<-c(t(matrix(xall,33,33)))
xy<-cbind(xall,yall)+matrix(-0.0075+0.015*runif(33*33*2),33*33,2)
par(pty="s",mfrow=c(1,2))
plot(xy[,1],xy[,2],pch=19,cex=0.25,xlab="Easting",ylab="Northing",
      cex.lab=1,cex.axis=1,cex.main=1)

set.seed(15892)
# Generate spatially random sample
xy.sample<-xy[sample(1:dim(xy)[1],50,replace=FALSE),]
points(xy.sample[,1],xy.sample[,2],pch=19,col="red")
points(xy[,1],xy[,2],pch=19,cex=0.25)
plot(xy[,1],xy[,2],pch=19,cex=0.25,xlab="Easting",ylab="Northing",
      cex.lab=1,cex.axis=1,cex.main=1)

set.seed(15892)
# Generate spatially regular sample
xy.sample<-discrete.sample(xy,50,0.08)
points(xy.sample[,1],xy.sample[,2],pch=19,col="red")
points(xy[,1],xy[,2],pch=19,cex=0.25)
```

---

Laplace.sampling

*Langevin-Hastings MCMC for conditional simulation*

---

**Description**

This function simulates from the conditional distribution of a Gaussian random effect, given binomial observations  $y$ .

**Usage**

```
Laplace.sampling(mu, Sigma, y, units.m, control.mcmc, ID.coords = NULL,
  messages = TRUE, plot.correlogram = TRUE)
```

**Arguments**

<code>mu</code>	mean vector of the marginal distribution of the random effect.
<code>Sigma</code>	covariance matrix of the marginal distribution of the random effect.
<code>y</code>	vector of binomial observations.
<code>units.m</code>	vector of binomial denominators.
<code>control.mcmc</code>	output from <code>control.mcmc.MCML</code> .
<code>ID.coords</code>	vector of ID values for the unique set of spatial coordinates obtained from <code>create.ID.coords</code> . These must be provided if, for example, spatial random effects are defined at household level but some of the covariates are at individual level. <b>Warning:</b> the household coordinates must all be distinct otherwise see <code>jitterDupCoords</code> . Default is NULL.
<code>messages</code>	logical; if <code>messages=TRUE</code> then status messages are printed on the screen (or output device) while the function is running. Default is <code>messages=TRUE</code> .
<code>plot.correlogram</code>	logical; if <code>plot.correlogram=TRUE</code> the autocorrelation plot of the conditional simulations is displayed.

**Details**

Conditionally on the random effect  $S$ , the data  $y$  follow a binomial distribution with probability  $p$  and binomial denominators `units.m`. The logistic link function is used for the linear predictor, which assumes the form

$$\log(p/(1-p)) = S.$$

The random effect  $S$  has a multivariate Gaussian distribution with mean `mu` and covariance matrix `Sigma`.

**Laplace sampling.** This function generates samples from the distribution of  $S$  given the data  $y$ . Specifically a Langevin-Hastings algorithm is used to update  $\tilde{S} = \tilde{\Sigma}^{-1/2}(S - \tilde{s})$  where  $\tilde{\Sigma}$  and  $\tilde{s}$  are the inverse of the negative Hessian and the mode of the distribution of  $S$  given  $y$ , respectively. At each iteration a new value  $\tilde{s}_{prop}$  for  $\tilde{S}$  is proposed from a multivariate Gaussian distribution with mean

$$\tilde{s}_{curr} + (h/2)\nabla \log f(\tilde{S}|y),$$

where  $\tilde{s}_{curr}$  is the current value for  $\tilde{S}$ ,  $h$  is a tuning parameter and  $\nabla \log f(\tilde{S}|y)$  is the the gradient of the log-density of the distribution of  $\tilde{S}$  given  $y$ . The tuning parameter  $h$  is updated according to the following adaptive scheme: the value of  $h$  at the  $i$ -th iteration, say  $h_i$ , is given by

$$h_i = h_{i-1} + c_1 i^{-c_2} (\alpha_i - 0.547),$$

where  $c_1 > 0$  and  $0 < c_2 < 1$  are pre-defined constants, and  $\alpha_i$  is the acceptance rate at the  $i$ -th iteration (0.547 is the optimal acceptance rate for a multivariate standard Gaussian distribution). The starting value for  $h$ , and the values for  $c_1$  and  $c_2$  can be set through the function `control.mcmc.MCML`.

**Random effects at household-level.** When the data consist of two nested levels, such as households and individuals within households, the argument `ID.coords` must be used to define the household

IDs for each individual. Let  $i$  and  $j$  denote the  $i$ -th household and the  $j$ -th person within that household; the logistic link function then assumes the form

$$\log(p_{ij}/(1 - p_{ij})) = \mu_{ij} + S_i$$

where the random effects  $S_i$  are now defined at household level and have mean zero.

### Value

A list with the following components

samples: a matrix, each row of which corresponds to a sample from the predictive distribution.

h: vector of the values of the tuning parameter at each iteration of the Langevin-Hastings MCMC algorithm.

### Author(s)

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

### See Also

[control.mcmc.MCML](#), [create.ID.coords](#).

### Examples

```
set.seed(1234)
data(data_sim)
n.subset <- 50
data_subset <- data_sim[sample(1:nrow(data_sim),n.subset),]
mu <- rep(0,50)
Sigma <- varcov.spatial(coords=data_subset[,c("x1","x2")],
                        cov.pars=c(1,0.15),kappa=2)$varcov
control.mcmc <- control.mcmc.MCML(n.sim=1000,burnin=0,thin=1,
                                h=1.65/(n.subset^2/3))
invisible(Laplace.sampling(mu=mu,Sigma=Sigma,
                          y=data_subset$y,units.m=data_subset$units.m,
                          control.mcmc=control.mcmc))
```

---

Laplace.sampling.lr     *Langevin-Hastings MCMC for conditional simulation (low-rank approximation)*

---

### Description

This function simulates from the conditional distribution of the random effects in a binomial mixed model.

**Usage**

```
Laplace.sampling.lr(mu, sigma2, K, y, units.m, control.mcmc, messages = TRUE,
  plot.correlogram = TRUE)
```

**Arguments**

mu	mean vector of the linear predictor.
sigma2	variance of the random effect.
K	random effect design matrix, or kernel matrix for the low-rank approximation.
y	vector of binomial observations.
units.m	vector of binomial denominators.
control.mcmc	output from <code>control.mcmc.MCML</code> .
messages	logical; if messages=TRUE then status messages are printed on the screen (or output device) while the function is running. Default is messages=TRUE.
plot.correlogram	logical; if plot.correlogram=TRUE the autocorrelation plot of the conditional simulations is displayed.

**Details**

Conditionally on  $Z$ , the data  $y$  follow a binomial distribution with probability  $p$  and binomial denominators `units.m`. Let  $K$  denote the random effects design matrix; a logistic link function is used, thus the linear predictor assumes the form

$$\log(p/(1-p)) = \mu + KZ$$

where  $\mu$  is the mean vector component defined through `mu`. The random effect  $Z$  has iid components distributed as zero-mean Gaussian variables with variance `sigma2`.

**Laplace sampling.** This function generates samples from the distribution of  $Z$  given the data  $y$ . Specifically, a Langevin-Hastings algorithm is used to update  $\tilde{Z} = \tilde{\Sigma}^{-1/2}(Z - \tilde{z})$  where  $\tilde{\Sigma}$  and  $\tilde{z}$  are the inverse of the negative Hessian and the mode of the distribution of  $Z$  given  $y$ , respectively. At each iteration a new value  $\tilde{z}_{prop}$  for  $\tilde{Z}$  is proposed from a multivariate Gaussian distribution with mean

$$\tilde{z}_{curr} + (h/2)\nabla \log f(\tilde{Z}|y),$$

where  $\tilde{z}_{curr}$  is the current value for  $\tilde{Z}$ ,  $h$  is a tuning parameter and  $\nabla \log f(\tilde{Z}|y)$  is the the gradient of the log-density of the distribution of  $\tilde{Z}$  given  $y$ . The tuning parameter  $h$  is updated according to the following adaptive scheme: the value of  $h$  at the  $i$ -th iteration, say  $h_i$ , is given by

$$h_i = h_{i-1} + c_1 i^{-c_2} (\alpha_i - 0.547),$$

where  $c_1 > 0$  and  $0 < c_2 < 1$  are pre-defined constants, and  $\alpha_i$  is the acceptance rate at the  $i$ -th iteration (0.547 is the optimal acceptance rate for a multivariate standard Gaussian distribution). The starting value for  $h$ , and the values for  $c_1$  and  $c_2$  can be set through the function `control.mcmc.MCML`.

**Value**

A list with the following components

`samples`: a matrix, each row of which corresponds to a sample from the predictive distribution.

`h`: vector of the values of the tuning parameter at each iteration of the Langevin-Hastings MCMC algorithm.

**Author(s)**

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

**See Also**

[control.mcmc.MCML](#).

---

linear.model.Bayes      *Bayesian estimation for the geostatistical linear Gaussian model*

---

**Description**

This function performs Bayesian estimation for the geostatistical linear Gaussian model.

**Usage**

```
linear.model.Bayes(formula, coords, data, kappa, control.mcmc, control.prior,
  low.rank = FALSE, knots = NULL, messages = TRUE)
```

**Arguments**

<code>formula</code>	an object of class " <a href="#">formula</a> " (or one that can be coerced to that class): a symbolic description of the model to be fitted.
<code>coords</code>	an object of class <a href="#">formula</a> indicating the geographic coordinates.
<code>data</code>	a data frame containing the variables in the model.
<code>kappa</code>	shape parameter of the Matern covariance function.
<code>control.mcmc</code>	output from <a href="#">control.mcmc.Bayes</a> .
<code>control.prior</code>	output from <a href="#">control.prior</a> .
<code>low.rank</code>	logical; if <code>low.rank=TRUE</code> a low-rank approximation is fitted.
<code>knots</code>	if <code>low.rank=TRUE</code> , <code>knots</code> is a matrix of spatial knots used in the low-rank approximation. Default is <code>knots=NULL</code> .
<code>messages</code>	logical; if <code>messages=TRUE</code> then status messages are printed on the screen (or output device) while the function is running. Default is <code>messages=TRUE</code> .

## Details

This function performs Bayesian estimation for the geostatistical linear Gaussian model, specified as

$$Y = d'\beta + S(x) + Z,$$

where  $Y$  is the measured outcome,  $d$  is a vector of covariates,  $\beta$  is a vector of regression coefficients,  $S(x)$  is a stationary Gaussian spatial process and  $Z$  are independent zero-mean Gaussian variables with variance `tau2`. More specifically,  $S(x)$  has an isotropic Matern covariance function with variance `sigma2`, scale parameter `phi` and shape parameter `kappa`. The shape parameter `kappa` is treated as fixed.

**Priors definition.** Priors can be defined through the function `control.prior`. The hierarchical structure of the priors is the following. Let  $\theta$  be the vector of the covariance parameters  $(\sigma^2, \phi, \tau^2)$ ; then each component of  $\theta$  can have independent priors freely defined by the user. However, uniform and log-normal priors are also available as default priors for each of the covariance parameters. To remove the nugget effect  $Z$ , no prior should be defined for `tau2`. Conditionally on `sigma2`, the vector of regression coefficients `beta` has a multivariate Gaussian prior with mean `beta.mean` and covariance matrix `sigma2*beta.covar`, while in the low-rank approximation the covariance matrix is simply `beta.covar`.

**Updating the covariance parameters using a Metropolis-Hastings algorithm.** In the MCMC algorithm implemented in `linear.model.Bayes`, the transformed parameters

$$(\theta_1, \theta_2, \theta_3) = (\log(\sigma^2)/2, \log(\sigma^2/\phi^{2\kappa}), \log(\tau^2))$$

are independently updated using a Metropolis Hastings algorithm. At the  $i$ -th iteration, a new value is proposed for each from a univariate Gaussian distribution with variance, say  $h_i^2$ , tuned according the following adaptive scheme

$$h_i = h_{i-1} + c_1 i^{-c_2} (\alpha_i - 0.45),$$

where  $\alpha_i$  is the acceptance rate at the  $i$ -th iteration (0.45 is the optimal acceptance rate for a univariate Gaussian distribution) whilst  $c_1 > 0$  and  $0 < c_2 < 1$  are pre-defined constants. The starting values  $h_1$  for each of the parameters  $\theta_1, \theta_2$  and  $\theta_3$  can be set using the function `control.mcmc.Bayes` through the arguments `h.theta1`, `h.theta2` and `h.theta3`. To define values for  $c_1$  and  $c_2$ , see the documentation of `control.mcmc.Bayes`.

**Low-rank approximation.** In the case of very large spatial data-sets, a low-rank approximation of the Gaussian spatial process  $S(x)$  might be computationally beneficial. Let  $(x_1, \dots, x_m)$  and  $(t_1, \dots, t_m)$  denote the set of sampling locations and a grid of spatial knots covering the area of interest, respectively. Then  $S(x)$  is approximated as  $\sum_{i=1}^m K(\|x - t_i\|; \phi, \kappa) U_i$ , where  $U_i$  are zero-mean mutually independent Gaussian variables with variance `sigma2` and  $K(\cdot; \phi, \kappa)$  is the isotropic Matern kernel (see `matern.kernel`). Since the resulting approximation is no longer a stationary process (but only approximately), `sigma2` may take very different values from the actual variance of the Gaussian process to approximate. The function `adjust.sigma2` can then be used to (approximately) explore the range for `sigma2`. For example if the variance of the Gaussian process is 0.5, then an approximate value for `sigma2` is `0.5/const.sigma2`, where `const.sigma2` is the value obtained with `adjust.sigma2`.

## Value

An object of class "Bayes.PrevMap". The function `summary.Bayes.PrevMap` is used to print a summary of the fitted model. The object is a list with the following components:

estimate: matrix of the posterior samples for each of the model parameters.  
 S: matrix of the posterior samples for each component of the random effect. This is only returned for the low-rank approximation.  
 y: response variable.  
 D: matrix of covariates.  
 coords: matrix of the observed sampling locations.  
 kappa: values of the shape parameter of the Matern function.  
 knots: matrix of spatial knots used in the low-rank approximation.  
 const.sigma2: vector of the values of the multiplicative factor used to adjust the sigma2 in the low-rank approximation.  
 h1: vector of values taken by the tuning parameter h.theta1 at each iteration.  
 h2: vector of values taken by the tuning parameter h.theta2 at each iteration.  
 h3: vector of values taken by the tuning parameter h.theta3 at each iteration.  
 call: the matched call.

### Author(s)

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

### References

Higdon, D. (1998). *A process-convolution approach to modeling temperatures in the North Atlantic Ocean*. Environmental and Ecological Statistics 5, 173-190.

### See Also

[control.prior](#), [control.mcmc.Bayes](#), [shape.matern](#), [summary.Bayes.PrevMap](#), [autocor.plot](#), [trace.plot](#), [dens.plot](#), [matern](#), [matern.kernel](#), [adjust.sigma2](#).

### Examples

```
set.seed(1234)
data(loaloe)
# Empirical logit transformation
loaloe$logit <- log((loaloe$NO_INF+0.5)/(loaloe$NO_EXAM-loaloe$NO_INF+0.5))

cp <- control.prior(beta.mean=-2.3,beta.covar=20,
                   log.normal.sigma2=c(0.9,5),
                   log.normal.phi=c(-0.17,2),
                   log.normal.nugget=c(-1,1))
control.mcmc <- control.mcmc.Bayes(n.sim=10,burnin=0,thin=1,
                                  h.theta1=0.5,h.theta2=0.5,h.theta3=0.5,
                                  c1.h.theta3=0.01,c2.h.theta3=0.0001,linear.model=TRUE,
                                  start.beta=-2.3,start.sigma2=2.45,
                                  start.phi=0.65,start.nugget=0.34)
fit.Bayes <- linear.model.Bayes(logit ~ 1,coords=~LONGITUDE+LATITUDE,
```

```

data=loaloe,kappa=0.5, control.mcmc=control.mcmc,
control.prior = cp)
summary(fit.Bayes)

```

---

linear.model.MLE	<i>Maximum Likelihood estimation for the geostatistical linear Gaussian model</i>
------------------	---

---

### Description

This function performs maximum likelihood estimation for the geostatistical linear Gaussian Model.

### Usage

```

linear.model.MLE(formula, coords, data, kappa, fixed.rel.nugget = NULL,
start.cov.pars, method = "BFGS", low.rank = FALSE, knots = NULL,
messages = TRUE)

```

### Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted.
coords	an object of class <code>formula</code> indicating the geographic coordinates.
data	a data frame containing the variables in the model.
kappa	shape parameter of the Matern covariance function.
fixed.rel.nugget	fixed value for the relative variance of the nugget effect; default is <code>fixed.rel.nugget=NULL</code> if this should be included in the estimation.
start.cov.pars	a vector of length two with elements corresponding to the starting values of $\phi$ and the relative variance of the nugget effect $\nu^2$ , respectively, that are used in the optimization algorithm. If $\nu^2$ is fixed through <code>fixed.rel.nugget</code> , then <code>start.cov.pars</code> represents the starting value for $\phi$ only.
method	method of optimization. If <code>method="BFGS"</code> then the <code>maxBFGS</code> function is used; otherwise <code>method="nlminb"</code> to use the <code>nlminb</code> function. Default is <code>method="BFGS"</code> .
low.rank	logical; if <code>low.rank=TRUE</code> a low-rank approximation of the Gaussian spatial process is used when fitting the model. Default is <code>low.rank=FALSE</code> .
knots	if <code>low.rank=TRUE</code> , <code>knots</code> is a matrix of spatial knots that are used in the low-rank approximation. Default is <code>knots=NULL</code> .
messages	logical; if <code>messages=TRUE</code> then status messages are printed on the screen (or output device) while the function is running. Default is <code>messages=TRUE</code> .



## Details

This function estimates the parameters of a geostatistical linear Gaussian model, specified as

$$Y = d'\beta + S(x) + Z,$$

where  $Y$  is the measured outcome,  $d$  is a vector of covariates,  $\beta$  is a vector of regression coefficients,  $S(x)$  is a stationary Gaussian spatial process and  $Z$  are independent zero-mean Gaussian variables with variance `tau2`. More specifically,  $S(x)$  has an isotropic Matern covariance function with variance `sigma2`, scale parameter `phi` and shape parameter `kappa`. In the estimation, the shape parameter `kappa` is treated as fixed. The relative variance of the nugget effect, `nu2=tau2/sigma2`, can be fixed through the argument `fixed.rel.nugget`; if `fixed.rel.nugget=NULL`, then the variance of the nugget effect is also included in the estimation.

**Low-rank approximation.** In the case of very large spatial data-sets, a low-rank approximation of the Gaussian spatial process  $S(x)$  can be computationally beneficial. Let  $(x_1, \dots, x_m)$  and  $(t_1, \dots, t_m)$  denote the set of sampling locations and a grid of spatial knots covering the area of interest, respectively. Then  $S(x)$  is approximated as  $\sum_{i=1}^m K(\|x - t_i\|; \phi, \kappa) U_i$ , where  $U_i$  are zero-mean mutually independent Gaussian variables with variance `sigma2` and  $K(\cdot; \phi, \kappa)$  is the isotropic Matern kernel (see [matern.kernel](#)). Since the resulting approximation is no longer a stationary process, the parameter `sigma2` is adjusted by a factor `constant.sigma2`. See [adjust.sigma2](#) for more details on the the computation of the adjustment factor `constant.sigma2` in the low-rank approximation.

## Value

An object of class "PrevMap". The function [summary.PrevMap](#) is used to print a summary of the fitted model. The object is a list with the following components:

`estimate`: estimates of the model parameters; use the function [coef.PrevMap](#) to obtain estimates of covariance parameters on the original scale.

`covariance`: covariance matrix of the ML estimates.

`log.lik`: maximum value of the log-likelihood.

`y`: response variable.

`D`: matrix of covariates.

`coords`: matrix of the observed sampling locations.

`method`: method of optimization used.

`kappa`: fixed value of the shape parameter of the Matern function.

`knots`: matrix of the spatial knots used in the low-rank approximation.

`const.sigma2`: adjustment factor for `sigma2` in the low-rank approximation.

`fixed.rel.nugget`: fixed value for the relative variance of the nugget effect.

`call`: the matched call.

## Author(s)

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

## References

Higdon, D. (1998). *A process-convolution approach to modeling temperatures in the North Atlantic Ocean*. Environmental and Ecological Statistics 5, 173-190.

## See Also

[shape.matern](#), [summary.PrevMap](#), [coef.PrevMap](#), [matern](#), [matern.kernel](#), [maxBFGS](#), [nlminb](#).

## Examples

```
data(loaloa)
# Empirical logit transformation
loaloa$logit <- log((loaloa$NO_INF+0.5)/(loaloa$NO_EXAM-loaloa$NO_INF+0.5))
fit.MLE <- linear.model.MLE(logit ~ 1, coords=~LONGITUDE+LATITUDE,
                           data=loaloa, start.cov.pars=c(0.2,0.15),
                           kappa=0.5)
summary(fit.MLE)

# Low-rank approximation
data(data_sim)
n.subset <- 200
data_subset <- data_sim[sample(1:nrow(data_sim),n.subset),]

# Logit transformation
data_subset$logit <- log(data_subset$y+0.5)/
                    (data_subset$units.m-data_subset$y+0.5)
knots <- as.matrix(expand.grid(seq(-0.2,1.2,length=8),seq(-0.2,1.2,length=8)))

fit <- linear.model.MLE(formula=logit~1, coords=~x1+x2, data=data_subset,
                       kappa=2, start.cov.pars=c(0.15,0.1), low.rank=TRUE,
                       knots=knots)
summary(fit, log.cov.pars=FALSE)
```

---

loaloa

*Loa loa prevalence data from 197 village surveys*

---

## Description

This data-set relates to a study of the prevalence of *Loa loa* (eyeworm) in a series of surveys undertaken in 197 villages in west Africa (Cameroon and southern Nigeria). The variables are as follows:

- ROW row id: 1 to 197.
- VILLCODE village id.
- LONGITUDE Longitude in degrees.
- LATITUDE Latitude in degrees.
- NO\_EXAM Number of people tested.

- NO\_INF Number of positive test results.
- ELEVATION Height above sea-level in metres.
- MEAN9901 Mean of all NDVI values recorded at village location, 1999-2001
- MAX9901 Maximum of all NDVI values recorded at village location, 1999-2001
- MIN9901 Minimum of all NDVI values recorded at village location, 1999-2001
- MIN9901 Minimum of all NDVI values recorded at village location, 1999-2001
- STDEV9901 standard deviation of all NDVI values recorded at village location, 1999-2001

### Usage

```
data(loaloa)
```

### Format

A data frame with 197 rows and 11 variables

### References

Diggle, P.J., Thomson, M.C., Christensen, O.F., Rowlingson, B., Obsomer, V., Gardon, J., Wanji, S., Takougang, I., Enyong, P., Kamgno, J., Remme, H., Boussinesq, M. and Molyneux, D.H. (2007). Spatial modelling and prediction of Loa loa risk: decision making under uncertainty. *Annals of Tropical Medicine and Parasitology*, 101, 499-509.

---

loglik.ci

*Profile likelihood confidence intervals*

---

### Description

Computes confidence intervals based on the interpolated profile likelihood computed for a single covariance parameter.

### Usage

```
loglik.ci(object, coverage = 0.95, plot.spline.profile = TRUE)
```

### Arguments

object	object of class "profile.PrevMap" obtained from <a href="#">loglik.linear.model</a> .
coverage	a value between 0 and 1 indicating the coverage of the confidence interval based on the interpolated profile likelihood. Default is coverage=0.95.
plot.spline.profile	logical; if TRUE an interpolating spline of the profile-likelihood of for a univariate parameter is plotted. Default is FALSE.

**Value**

A list with elements `lower` and `upper` for the upper and lower limits of the confidence interval, respectively.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

`loglik.linear.model`    *Profile log-likelihood or fixed parameters likelihood evaluation for the covariance parameters in the geostatistical linear model*

---

**Description**

Computes profile log-likelihood, or evaluates likelihood keeping the other parameters fixed, for the scale parameter  $\phi$  of the Matern function and the relative variance of the nugget effect  $\nu^2$  in the linear Gaussian model.

**Usage**

```
loglik.linear.model(object, control.profile, plot.profile = TRUE,
  messages = TRUE)
```

**Arguments**

`object`            an object of class 'PrevMap', which is the fitted linear model obtained with the function `linear.model.MLE`.

`control.profile`    control parameters obtained with `control.profile`.

`plot.profile`      logical; if TRUE a plot of the computed profile likelihood is displayed.

`messages`          logical; if `messages=TRUE` then status messages are printed on the screen (or output device) while the function is running. Default is `messages=TRUE`.

**Value**

an object of class "profile.PrevMap" which is a list with the following values

- `eval.points.phi`: vector of the values used for  $\phi$  in the evaluation of the likelihood.
- `eval.points.rel.nugget`: vector of the values used for  $\nu^2$  in the evaluation of the likelihood.
- `profile.phi`: vector of the values of the likelihood function evaluated at `eval.points.phi`.
- `profile.rel.nugget`: vector of the values of the likelihood function evaluated at `eval.points.rel.nugget`.
- `profile.phi.rel.nugget`: matrix of the values of the likelihood function evaluated at `eval.points.phi` and `eval.points.rel.nugget`.
- `fixed.par`: logical value; TRUE is the evaluation if the likelihood is carried out by fixing the other parameters, and FALSE if the computation of the profile-likelihood was performed instead.

**Author(s)**

Emanuele Giorgi &lt;e.giorgi@lancaster.ac.uk&gt;

Peter J. Diggle &lt;p.diggle@lancaster.ac.uk&gt;

matern.kernel

*Matern kernel***Description**

This function computes values of the Matern kernel for given distances and parameters.

**Usage**

```
matern.kernel(u, rho, kappa)
```

**Arguments**

u	a vector, matrix or array with values of the distances between pairs of data locations.
rho	value of the (re-parametrized) scale parameter; this corresponds to the re-parametrization $\text{rho} = 2 \cdot \sqrt{\text{kappa}} \cdot \text{phi}$ .
kappa	value of the shape parameter.

**Details**

The Matern kernel is defined as:

$$K(u; \phi, \kappa) = \frac{\Gamma(\kappa + 1)^{1/2} \kappa^{(\kappa+1)/4} u^{(\kappa-1)/2}}{\pi^{1/2} \Gamma((\kappa + 1)/2) \Gamma(\kappa)^{1/2} (2\kappa^{1/2}\phi)^{(\kappa+1)/2}} \mathcal{K}_\kappa(u/\phi), u > 0,$$

where  $\phi$  and  $\kappa$  are the scale and shape parameters, respectively, and  $\mathcal{K}_\kappa(\cdot)$  is the modified Bessel function of the third kind of order  $\kappa$ . The family is valid for  $\phi > 0$  and  $\kappa > 0$ .

**Value**

A vector matrix or array, according to the argument u, with the values of the Matern kernel function for the given distances.

**Author(s)**

Emanuele Giorgi &lt;e.giorgi@lancaster.ac.uk&gt;

Peter J. Diggle &lt;p.diggle@lancaster.ac.uk&gt;

---

plot.pred.PrevMap      *Plot of a predicted surface*

---

### Description

plot.pred.PrevMap displays predictions obtained from [spatial.pred.linear.MLE](#), [spatial.pred.linear.Bayes](#), [spatial.pred.linear.Matern](#) and [spatial.pred.binomial.Bayes](#).

### Usage

```
## S3 method for class 'pred.PrevMap'
plot(x, type = NULL, summary = "predictions", ...)
```

### Arguments

x	an object of class "PrevMap".
type	a character indicating the type of prediction to display: 'prevalence', 'odds', 'logit' or 'probit'. Default is NULL.
summary	character indicating which summary to display: 'predictions', 'quantiles', 'standard.errors' or 'exceedance.prob'; default is 'predictions'. If summary="exceedance.prob", the argument type is ignored.
...	further arguments passed to <a href="#">plot</a> .

### Author(s)

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

---

plot.profile.PrevMap      *Plot of the profile log-likelihood for the covariance parameters of the Matern function*

---

### Description

This function displays a plot of the profile log-likelihood that is computed by the function [loglik.linear.model](#).

### Usage

```
## S3 method for class 'profile.PrevMap'
plot(x, log.scale = FALSE,
     plot.spline.profile = FALSE, ...)
```

**Arguments**

x	object of class "profile.PrevMap" obtained as output from <a href="#">loglik.linear.model</a> .
log.scale	logical; if log.scale=TRUE, the profile likelihood is plotted on the log-scale of the parameter values.
plot.spline.profile	logical; if TRUE an interpolating spline of the profile-likelihood of for a univariate parameter is plotted. Default is FALSE.
...	further arguments passed to <a href="#">plot</a> if the profile log-likelihood is for only one parameter, or to <a href="#">contour</a> for the bi-variate profile-likelihood.

**Value**

A plot is returned. No value is returned.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

plot.shape.matern	<i>Plot of the profile likelihood for the shape parameter of the Matern covariance function</i>
-------------------	---

---

**Description**

This function plots the profile likelihood for the shape parameter of the Matern covariance function using the output from [shape.matern](#) function.

**Usage**

```
## S3 method for class 'shape.matern'
plot(x, plot.spline = TRUE, ...)
```

**Arguments**

x	an object of class 'shape.matern' obtained as result of a call to <a href="#">shape.matern</a>
plot.spline	logical; if TRUE an interpolating spline of the profile likelihood is added to the plot.
...	further arguments passed to <a href="#">plot</a> .

**Value**

The function does not return any value.

**Author(s)**

Emanuele Giorgi &lt;e.giorgi@lancaster.ac.uk&gt;

Peter J. Diggle &lt;p.diggle@lancaster.ac.uk&gt;

**See Also**[shape.matern](#)


---

shape.matern	<i>Profile likelihood for the shape parameter of the Matern covariance function</i>
--------------	---

---

**Description**

This function plots the profile likelihood for the shape parameter of the Matern covariance function used in the linear Gaussian model. It also computes confidence intervals of coverage coverage by interpolating the profile likelihood with a spline and using the asymptotic distribution of a chi-squared with one degree of freedom.

**Usage**

```
shape.matern(formula, coords, data, set.kappa, fixed.rel.nugget = NULL,
             start.par, coverage = NULL, plot.profile = TRUE, messages = TRUE)
```

**Arguments**

formula	an object of class <a href="#">formula</a> (or one that can be coerced to that class): a symbolic description of the model to be fitted.
coords	an object of class <a href="#">formula</a> indicating the geographic coordinates.
data	a data frame containing the variables in the model.
set.kappa	a vector indicating the set values for evaluation of the profile likelihood.
fixed.rel.nugget	a value for the relative variance nu2 of the nugget effect, that is then treated as fixed. Default is NULL.
start.par	starting values for the scale parameter phi and the relative variance of the nugget effect nu2; if fixed.rel.nugget is provided, then a starting value for phi only should be provided.
coverage	a value between 0 and 1 indicating the coverage of the confidence interval based on the interpolated profile likelihood for the shape parameter. Default is coverage=NULL and no confidence interval is then computed.
plot.profile	logical; if TRUE the computed profile-likelihood is plotted together with the interpolating spline.
messages	logical; if messages=TRUE then status messages are printed on the screen (or output device) while the function is running. Default is messages=TRUE.



**Value**

The function returns an object of class 'shape.matern' that is a list with the following components

- set.kappa set of values of the shape parameter used to evaluate the profile-likelihood.

- val.kappa values of the profile likelihood.

If a value for coverage is specified, the list also contains lower, upper and kappa.hat that correspond to the lower and upper limits of the confidence interval, and the maximum likelihood estimate for the shape parameter, respectively.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

spatial.pred.binomial.Bayes

*Bayesian spatial prediction for the binomial logistic and binary probit models*

---

**Description**

This function performs Bayesian spatial prediction for the binomial logistic and binary probit models.

**Usage**

```
spatial.pred.binomial.Bayes(object, grid.pred, predictors = NULL,
  type = "marginal", scale.predictions = "prevalence",
  quantiles = c(0.025, 0.975), standard.errors = FALSE, thresholds = NULL,
  scale.thresholds = NULL, messages = TRUE)
```

**Arguments**

object	an object of class "Bayes.PrevMap" obtained as result of a call to <a href="#">binomial.logistic.Bayes</a> or <a href="#">binary.probit.Bayes</a> .
grid.pred	a matrix of prediction locations.
predictors	a data frame of the values of the explanatory variables at each of the locations in grid.pred; each column correspond to a variable and each row to a location. <b>Warning:</b> the names of the columns in the data frame must match those in the data used to fit the model. Default is predictors=NULL for models with only an intercept.
type	a character indicating the type of spatial predictions: type="marginal" for marginal predictions or type="joint" for joint predictions. Default is type="marginal". In the case of a low-rank approximation only joint predictions are available.

scale.predictions	a character vector of maximum length 3, indicating the required scale on which spatial prediction is carried out: "logit", "prevalence", "odds" and "probit". Default is scale.predictions="prevalence".
quantiles	a vector of quantiles used to summarise the spatial predictions.
standard.errors	logical; if standard.errors=TRUE, then standard errors for each scale.predictions are returned. Default is standard.errors=FALSE.
thresholds	a vector of exceedance thresholds; default is NULL.
scale.thresholds	a character value ("logit", "prevalence", "odds" or "probit") indicating the scale on which exceedance thresholds are provided.
messages	logical; if messages=TRUE then status messages are printed on the screen (or output device) while the function is running. Default is messages=TRUE.

### Value

A "pred.PrevMap" object list with the following components: logit; prevalence; odds; probit; exceedance.prob, corresponding to a matrix of the exceedance probabilities where each column corresponds to a specified value in thresholds; samples, corresponding to a matrix of the posterior samples at each prediction locations for the linear predictor; grid.pred prediction locations. Each of the three components logit, prevalence, odds and probit is also a list with the following components:

predictions: a vector of the predictive mean for the associated quantity (logit, odds or prevalence).

standard.errors: a vector of prediction standard errors (if standard.errors=TRUE).

quantiles: a matrix of quantiles of the resulting predictions with each column corresponding to a quantile specified through the argument quantiles.

### Author(s)

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>  
 Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

spatial.pred.binomial.MCML

*Spatial predictions for the binomial logistic model using plug-in of MCML estimates*

---

### Description

This function performs spatial prediction for fixed parameters at the Monte Carlo maximum likelihood estimates of a geostatistical binomial logistic model.

**Usage**

```
spatial.pred.binomial.MCML(object, grid.pred, predictors = NULL, control.mcmc,
  type = "marginal", scale.predictions = c("logit", "prevalence", "odds"),
  quantiles = c(0.025, 0.975), standard.errors = FALSE, thresholds = NULL,
  scale.thresholds = NULL, plot.correlogram = FALSE, messages = TRUE)
```

**Arguments**

object	an object of class "PrevMap" obtained as result of a call to <code>binomial.logistic.MCML</code> .
grid.pred	a matrix of prediction locations.
predictors	a data frame of the values of the explanatory variables at each of the locations in <code>grid.pred</code> ; each column correspond to a variable and each row to a location. <b>Warning:</b> the names of the columns in the data frame must match those in the data used to fit the model. Default is <code>predictors=NULL</code> for models with only an intercept.
control.mcmc	output from <code>control.mcmc.MCML</code> .
type	a character indicating the type of spatial predictions: <code>type="marginal"</code> for marginal predictions or <code>type="joint"</code> for joint predictions. Default is <code>type="marginal"</code> . In the case of a low-rank approximation only joint predictions are available.
scale.predictions	a character vector of maximum length 3, indicating the required scale on which spatial prediction is carried out: "logit", "prevalence" and "odds". Default is <code>scale.predictions=c("logit", "prevalence", "odds")</code> .
quantiles	a vector of quantiles used to summarise the spatial predictions.
standard.errors	logical; if <code>standard.errors=TRUE</code> , then standard errors for each <code>scale.predictions</code> are returned. Default is <code>standard.errors=FALSE</code> .
thresholds	a vector of exceedance thresholds; default is <code>thresholds=NULL</code> .
scale.thresholds	a character value indicating the scale on which exceedance thresholds are provided; "logit", "prevalence" or "odds". Default is <code>scale.thresholds=NULL</code> .
plot.correlogram	logical; if <code>plot.correlogram=TRUE</code> the autocorrelation plot of the conditional simulations is displayed.
messages	logical; if <code>messages=TRUE</code> then status messages are printed on the screen (or output device) while the function is running. Default is <code>messages=TRUE</code> .

**Value**

A "pred.PrevMap" object list with the following components: `logit`; `prevalence`; `odds`; `exceedance.prob`, corresponding to a matrix of the exceedance probabilities where each column corresponds to a specified value in `thresholds`; `samples`, corresponding to a matrix of the posterior samples at each prediction locations for the linear predictor of the binomial logistic model (if `scale.predictions="logit"` this component is `NULL`); `grid.pred` prediction locations. Each of the three components `logit`, `prevalence` and `odds` is also a list with the following components:

predictions: a vector of the predictive mean for the associated quantity (logit, odds or prevalence).  
 standard.errors: a vector of prediction standard errors (if standard.errors=TRUE).  
 quantiles: a matrix of quantiles of the resulting predictions with each column corresponding to a quantile specified through the argument quantiles.

### Author(s)

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

spatial.pred.linear.Bayes

*Bayesian spatial predictions for the geostatistical Linear Gaussian model*

---

### Description

This function performs Bayesian prediction for a geostatistical linear Gaussian model.

### Usage

```
spatial.pred.linear.Bayes(object, grid.pred, predictors = NULL,
  type = "marginal", scale.predictions = c("logit", "prevalence", "odds"),
  quantiles = c(0.025, 0.975), standard.errors = FALSE, thresholds = NULL,
  scale.thresholds = NULL, messages = TRUE)
```

### Arguments

object	an object of class "Bayes.PrevMap" obtained as result of a call to <a href="#">linear.model.Bayes</a> .
grid.pred	a matrix of prediction locations.
predictors	a data frame of the values of the explanatory variables at each of the locations in grid.pred; each column correspond to a variable and each row to a location. <b>Warning:</b> the names of the columns in the data frame must match those in the data used to fit the model. Default is predictors=NULL for models with only an intercept.
type	a character indicating the type of spatial predictions: type="marginal" for marginal predictions or type="joint" for joint predictions. Default is type="marginal". In the case of a low-rank approximation only joint predictions are available.
scale.predictions	a character vector of maximum length 3, indicating the required scale on which spatial prediction is carried out: "logit", "prevalence" and "odds". Default is scale.predictions=c("logit", "prevalence", "odds").
quantiles	a vector of quantiles used to summarise the spatial predictions.

standard.errors	logical; if standard.errors=TRUE, then standard errors for each scale.predictions are returned. Default is standard.errors=FALSE.
thresholds	a vector of exceedance thresholds; default is thresholds=NULL.
scale.thresholds	a character value indicating the scale on which exceedance thresholds are provided: "logit", "prevalence" or "odds". Default is scale.thresholds=NULL.
messages	logical; if messages=TRUE then status messages are printed on the screen (or output device) while the function is running. Default is messages=TRUE.

**Value**

A "pred.PrevMap" object list with the following components: logit; prevalence; odds; exceedance.prob, corresponding to a matrix of the exceedance probabilities where each column corresponds to a specified value in thresholds; grid.pred prediction locations. Each of the three components logit, prevalence and odds is also a list with the following components:

predictions: a vector of the predictive mean for the associated quantity (logit, odds or prevalence).

standard.errors: a vector of prediction standard errors (if standard.errors=TRUE).

quantiles: a matrix of quantiles of the resulting predictions with each column corresponding to a quantile specified through the argument quantiles.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

spatial.pred.linear.MLE

*Spatial predictions for the geostatistical Linear Gaussian model using plug-in of ML estimates*

---

**Description**

This function performs spatial prediction for fixed parameters at the maximum likelihood estimates of a linear geostatistical model.

**Usage**

```
spatial.pred.linear.MLE(object, grid.pred, predictors = NULL,
  type = "marginal", scale.predictions = c("logit", "prevalence", "odds"),
  quantiles = c(0.025, 0.975), n.sim.prev = 1000, standard.errors = FALSE,
  thresholds = NULL, scale.thresholds = NULL, messages = TRUE)
```

**Arguments**

object	an object of class "PrevMap" obtained as result of a call to <code>linear.model.MLE</code> .
grid.pred	a matrix of prediction locations.
predictors	a data frame of the values of the explanatory variables at each of the locations in <code>grid.pred</code> ; each column correspond to a variable and each row to a location. <b>Warning:</b> the names of the columns in the data frame must match those in the data used to fit the model. Default is <code>predictors=NULL</code> for models with only an intercept.
type	a character indicating the type of spatial predictions: <code>type="marginal"</code> for marginal predictions or <code>type="joint"</code> for joint predictions. Default is <code>type="marginal"</code> . In the case of a low-rank approximation only marginal predictions are available.
scale.predictions	a character vector of maximum length 3, indicating the required scale on which spatial prediction is carried out: "logit", "prevalence" and "odds". Default is <code>scale.predictions=c("logit","prevalence","odds")</code> .
quantiles	a vector of quantiles used to summarise the spatial predictions.
n.sim.prev	number of simulation for predictions of prevalence. Default is <code>n.sim.prev=1000</code> .
standard.errors	logical; if <code>standard.errors=TRUE</code> , then standard errors for each <code>scale.predictions</code> are returned. Default is <code>standard.errors=FALSE</code> .
thresholds	a vector of exceedance thresholds; default is <code>thresholds=NULL</code> .
scale.thresholds	a character value indicating the scale on which exceedance thresholds are provided; "logit", "prevalence" or "odds". Default is <code>scale.thresholds=NULL</code> .
messages	logical; if <code>messages=TRUE</code> then status messages are printed on the screen (or output device) while the function is running. Default is <code>messages=TRUE</code> .

**Value**

A "pred.PrevMap" object list with the following components: `logit`; `prevalence`; `odds`; `exceedance.prob`, corresponding to a matrix of the exceedance probabilities where each column corresponds to a specified value in `thresholds`; `grid.pred` prediction locations. Each of the three components `logit`, `prevalence` and `odds` is also a list with the following components:

`predictions`: a vector of the predictive mean for the associated quantity (logit, odds or prevalence).

`standard.errors`: a vector of prediction standard errors (if `standard.errors=TRUE`).

`quantiles`: a matrix of quantiles of the resulting predictions with each column corresponding to a quantile specified through the argument `quantiles`.

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

summary.Bayes.PrevMap *Summarizing Bayesian model fits*

---

### Description

summary method for the class "Bayes.PrevMap" that computes the posterior mean, median, mode and high posterior density intervals using samples from Bayesian fits.

### Usage

```
## S3 method for class 'Bayes.PrevMap'  
summary(object, hpd.coverage = 0.95, ...)
```

### Arguments

object	an object of class "Bayes.PrevMap" obtained as result of a call to <a href="#">binomial.logistic.Bayes</a> or <a href="#">linear.model.Bayes</a> .
hpd.coverage	value of the coverage of the high posterior density intervals; default is 0.95.
...	further arguments passed to or from other methods.

### Value

A list with the following values

linear: logical value that is TRUE if a linear model was fitted and FALSE otherwise.

binary: logical value that is TRUE if a binary model was fitted and FALSE otherwise.

probit: logical value that is TRUE if a binary model with probit link function was fitted and FALSE if with logistic link function.

ck: logical value that is TRUE if a low-rank approximation was fitted and FALSE otherwise.

beta: matrix of the posterior summaries for the regression coefficients.

sigma2: vector of the posterior summaries for sigma2.

phi: vector of the posterior summaries for phi.

tau2: vector of the posterior summaries for tau2.

call: matched call.

kappa: fixed value of the shape parameter of the Matern covariance function.

### Author(s)

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>

summary.PrevMap

*Summarizing likelihood-based model fits***Description**

summary method for the class "PrevMap" that computes the standard errors and p-values of likelihood-based model fits.

**Usage**

```
## S3 method for class 'PrevMap'
summary(object, log.cov.pars = TRUE, ...)
```

**Arguments**

object	an object of class "PrevMap" obtained as result of a call to <a href="#">binomial.logistic.MCML</a> or <a href="#">linear.model.MLE</a> .
log.cov.pars	logical; if log.cov.pars=TRUE the estimates of the covariance parameters are given on the log-scale. Note that standard errors are also adjusted accordingly. Default is log.cov.pars=TRUE.
...	further arguments passed to or from other methods.

**Value**

A list with the following components

linear: logical value; linear=TRUE if a linear model was fitted and linear=FALSE otherwise.

ck: logical value; ck=TRUE if a low-rank approximation was used and ck=FALSE otherwise.

coefficients: matrix of the estimates, standard errors and p-values of the estimates of the regression coefficients.

cov.pars: matrix of the estimates and standard errors of the covariance parameters.

log.lik: value of likelihood function at the maximum likelihood estimates.

kappa: fixed value of the shape parameter of the Matern covariance function.

fixed.rel.nugget: fixed value for the relative variance of the nugget effect.

call: matched call.

**Author(s)**

Emanuele Giorgi <[e.giorgi@lancaster.ac.uk](mailto:e.giorgi@lancaster.ac.uk)>

Peter J. Diggle <[p.diggle@lancaster.ac.uk](mailto:p.diggle@lancaster.ac.uk)>



---

trace.plot	<i>Trace-plots for posterior samples</i>
------------	--

---

**Description**

Displays the trace-plots for the posterior samples of the model parameters and spatial random effects.

**Usage**

```
trace.plot(object, param, component.beta = NULL, component.S = NULL)
```

**Arguments**

object	an object of class 'Bayes.PrevMap'.
param	a character indicating for which component of the model the density plot is required: param="beta" for the regression coefficients; param="sigma2" for the variance of the spatial random effect; param="phi" for the scale parameter of the Matern correlation function; param="tau2" for the variance of the nugget effect; param="S" for the spatial random effect.
component.beta	if param="beta", component.beta is a numeric value indicating the component of the regression coefficients; default is NULL.
component.S	if param="S", component.S can be a numeric value indicating the component of the spatial random effect. Default is NULL.
...	additional parameters to pass to <a href="#">density</a> .

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>  
 Peter J. Diggle <p.diggle@lancaster.ac.uk>

---

trace.plot.MCML	<i>Trace-plots of the importance sampling distribution samples from the MCML method</i>
-----------------	---

---

**Description**

Trace-plots of the MCMC samples from the importance sampling distribution used in [binomial.logistic.MCML](#).

**Usage**

```
trace.plot.MCML(object, component = NULL, ...)
```

**Arguments**

object	an object of class "PrevMap" obtained as result of a call to <code>binomial.logistic.MCML</code> .
component	a positive integer indicating the number of the random effect component for which a trace-plot is required. If component=NULL, then a component is selected at random. Default is component=NULL.
...	further arguments passed to <code>plot</code> .

**Author(s)**

Emanuele Giorgi <e.giorgi@lancaster.ac.uk>

Peter J. Diggle <p.diggle@lancaster.ac.uk>

# Index

## \*Topic **datasets**

- data\_sim, [22](#)
  - loaloo, [34](#)
- adjust.sigma2, [2](#), [6](#), [9](#), [30](#), [31](#), [33](#)
- autocor.plot, [3](#), [31](#)
- binary.probit.Bayes, [4](#), [41](#)
- binomial.logistic.Bayes, [7](#), [17](#), [18](#), [20](#), [41](#), [47](#)
- binomial.logistic.MCML, [10](#), [14](#), [19](#), [43](#), [48–50](#)
- coef.PrevMap, [12](#), [13](#), [14](#), [33](#), [34](#)
- continuous.sample, [15](#)
- contour, [16](#), [39](#)
- contour.pred.PrevMap, [16](#)
- control.mcmc.Bayes, [4–8](#), [10](#), [17](#), [29–31](#)
- control.mcmc.MCML, [11](#), [13](#), [18](#), [26–29](#), [43](#)
- control.prior, [4–8](#), [10](#), [19](#), [29–31](#)
- control.profile, [21](#), [36](#)
- create.ID.coords, [4](#), [6](#), [7](#), [10](#), [11](#), [13](#), [22](#), [26](#), [27](#)
- data\_sim, [22](#)
- dens.plot, [23](#), [31](#)
- density, [23](#), [49](#)
- discrete.sample, [24](#)
- formula, [4](#), [7](#), [11](#), [22](#), [29](#), [32](#), [40](#)
- jitterDupCoords, [4](#), [7](#), [11](#), [26](#)
- Laplace.sampling, [13](#), [25](#)
- Laplace.sampling.lr, [13](#), [27](#)
- linear.model.Bayes, [17](#), [18](#), [29](#), [44](#), [47](#)
- linear.model.MLE, [14](#), [32](#), [36](#), [46](#), [48](#)
- loaloo, [34](#)
- loglik.ci, [35](#)
- loglik.linear.model, [21](#), [35](#), [36](#), [38](#), [39](#)
- matern, [5](#), [6](#), [8](#), [10](#), [12](#), [13](#), [31](#), [34](#)
- matern.kernel, [3](#), [6](#), [9](#), [10](#), [12](#), [13](#), [30](#), [31](#), [33](#), [34](#), [37](#)
- maxBFGS, [11](#), [12](#), [32](#), [34](#)
- nlminb, [11](#), [12](#), [32](#), [34](#)
- pdist, [3](#)
- plot, [38](#), [39](#), [50](#)
- plot.pred.PrevMap, [38](#)
- plot.profile.PrevMap, [38](#)
- plot.shape.matern, [39](#)
- shape.matern, [31](#), [34](#), [39](#), [40](#), [40](#)
- spatial.pred.binomial.Bayes, [16](#), [38](#), [41](#)
- spatial.pred.binomial.MCML, [16](#), [38](#), [42](#)
- spatial.pred.linear.Bayes, [16](#), [38](#), [44](#)
- spatial.pred.linear.MLE, [16](#), [38](#), [45](#)
- summary.Bayes.PrevMap, [6](#), [9](#), [10](#), [30](#), [31](#), [47](#)
- summary.PrevMap, [12](#), [13](#), [33](#), [34](#), [48](#)
- trace.plot, [31](#), [49](#)
- trace.plot.MCML, [49](#)
- unique, [22](#)