

Ecography

ECOG-00279

Saas, Y. and Gosselin, F. 2014. Comparison of regression methods for spatially-autocorrelated count data on regularly- and irregularly-spaced locations. – Ecography 37: xxx–xxx.

Supplementary material

Appendix 1: Proof of the biased nature of the variance estimator in a simple spatial case

Let (X_1, \dots, X_n) be a suite of independent and identically distributed (i.i.d.) random variables of mean μ and variance σ^2 . Let S^2 denote the corrected estimator of the variance given by the following expression:

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

where \bar{X} is the mean of the random variables X_1, \dots, X_n .

It is easy to prove that S^2 is an unbiased estimator of σ^2 , i.e. that $\mathbb{E}(S^2) = \sigma^2$.

Let us now consider a simple case of spatial dependencies, in the form of an unidimensional autoregressive spatial process of order 1. Let (X_1, \dots, X_n) be a vector of spatially-autocorrelated random variables. The spatial dependencies are assumed to be in the following form:

$$\text{cov}(X_i, X_j) = \sigma^2 \phi^{|i-j|}$$

where σ^2 is the variance of the suite and ϕ a spatial scale parameter between 0 and 1.

Let us prove that the variance σ^2 is under-estimated by the conventional estimator S^2 , i.e. that:

$$\mathbb{E}(S^2) < \sigma^2$$

We have:

$$\begin{aligned} (n-1)S^2 &= \sum_{i=1}^n (X_i - \bar{X})^2 \\ &= \sum_{i=1}^n (X_i - \mu + \mu - \bar{X})^2 \\ &= \sum_{i=1}^n (X_i - \mu)^2 + 2(\mu - \bar{X}) \cdot \sum_{i=1}^n (X_i - \mu) + n(\mu - \bar{X})^2 \\ &= \sum_{i=1}^n (X_i - \mu)^2 - n(\mu - \bar{X})^2 \end{aligned}$$

By taking the mean, we obtain:

$$\begin{aligned} (n-1)\mathbb{E}(S^2) &= \sum_{i=1}^n \mathbb{E}((X_i - \mu)^2) - n\mathbb{E}((\mu - \bar{X})^2) \\ &= \sum_{i=1}^n \sigma^2 - nVar(\bar{X}) \\ &= n\sigma^2 - nVar(\bar{X}) \end{aligned}$$

In the context of non-correlated observations, we could have written:

$$Var(\bar{X}) = \frac{1}{n^2} \sum_{i=1}^n Var(X_i) = \frac{\sigma^2}{n}$$

This expression is no longer true in the case of spatially-autocorrelated data, because spatial dependencies have to be taken into account. This can be achieved as follows:

$$\begin{aligned}
Var(\bar{X}) &= \frac{1}{n^2} Var\left(\sum_{i=1}^n X_i\right) \\
&= \frac{1}{n^2} cov\left(\sum_{i=1}^n X_i, \sum_{j=1}^n X_j\right) \\
&= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \sigma^2 \phi^{|i-j|}
\end{aligned}$$

Hence, we obtain:

$$(n-1)\mathbb{E}(S^2) = \sigma^2 \left(n - \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n \phi^{|i-j|} \right)$$

A simple series calculation shows that:

$$\sum_{i=1}^n \sum_{j=1}^n \phi^{|i-j|} = n + \frac{2\phi}{1-\phi} (n-1) - 2 \left(\frac{\phi}{1-\phi} \right)^2 (1 - \phi^{n-1})$$

Thus:

$$(n-1)\mathbb{E}(S^2) = \sigma^2 \left[n - 1 - \frac{2\phi}{1-\phi} \frac{n-1}{n} + 2 \left(\frac{\phi}{1-\phi} \right)^2 \frac{1-\phi^{n-1}}{n} \right]$$

It follows that:

$$\begin{aligned}
\mathbb{E}(S^2) - \sigma^2 &= 2 \left(\frac{\phi}{1-\phi} \right)^2 \frac{1-\phi^{n-1}}{n(n-1)} - \frac{2\phi}{n(1-\phi)} \\
&= \frac{2\phi}{n(1-\phi)} \left[\frac{\phi}{1-\phi} \frac{1-\phi^{n-1}}{n-1} - 1 \right] \\
&= \frac{2\phi}{n(1-\phi)} \left[\frac{1}{n-1} \sum_{i=1}^{n-1} \phi^i - 1 \right]
\end{aligned}$$

Since ϕ is between 0 and 1, it follows that:

$$\frac{1}{n-1} \sum_{i=1}^{n-1} \phi^i < \frac{n-1}{n-1} = 1$$

Which provides the desired result:

$$\mathbb{E}(S^2) - \sigma^2 < 0$$

Appendix 2: Computing code to implement the spatially-correlated methods in the R software

GLMMPQL

```
library(MASS)
GLMMPQL=glmmPQL(fixed=Y1~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),
data=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim")))
struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
struct.cor=Initialize(struct.cor,as(my.data,"data.frame") [,c("x","y")])
GLMMPQL=update(GLMMPQL,correlation=struct.cor)
```

GAMM

```
library(mgcv)
struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
struct.cor=Initialize(struct.cor,as(my.data,"data.frame") [,c("x","y")])
GAMM=gamm(Y1~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",
control=list(lmeControl(returnObject=TRUE,opt="optim")))
```

INLA

```
library(INLA)
INLA1=inla(Y1~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",
hyper=list(prec=list(param=c(1,0.01),initial=1))),family="poisson",data=my.data,
verbose=F,control.predictor=list(compute=T),control.inla=list(h=1e-4))
```

MCMCLH

```
library(geoRglm)
my.S.scale=0.0003
my.phi.scale=1
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y1
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,
cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,
burn.in=30000,n.iter=70000,phi.start=0)
MCMCLH=pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcinput,
model=my.model)
```

Appendix 3: R code to compute the gridded scenarios

```

setwd("P:/")

library(MASS)
library(pixmap)
library(sp)
library(spdep)
library(INLA)
library(mgcv)
library(RandomFields)
library(gstat)
library(geoR)
library(geoRglm)
library(magic)

I=20      # size of the grid
B=1000    # number of replicates
n=I*I     # number of observations

obs=1:n

x=rep(1:I,times=I)      # vector of the x-coords (row)
y=rep(1:I,each=I)        # vector of the y-coords (col)

## Neighbourhood matrix for INLA ##

grid=cell2nb(nrow=I,ncol=I,type="rook",torus=F)
nb2INLA("grid.dat",grid)

## GRID simulations ##

M=9      # number of methods
P=5      # number of regression parameters (intercept + 4 covariates)
HP=2     # number of hyperparameters (spatial parameters)
S=10     # number of scenarios

beta=c(1,rep(0.5,P-1))  # vector of regression parameters

# Initialisation of the coefficients #

GLM.coef=vector("list",S)
GLMquasi.coef=vector("list",S)
GLMMPQLiid.coef=vector("list",S)
GAMMid.coef=vector("list",S)
INLAiid.coef=vector("list",S)
GLMMPQL.coef=vector("list",S)
GAMM.coef=vector("list",S)
INLA.coef=vector("list",S)
MCMCLH.coef=vector("list",S)

for (s in 1:S)
{
  GLM.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GLMquasi.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GLMMPQLiid.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GAMMid.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  INLAiid.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GLMMPQL.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GAMM.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  INLA.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  MCMCLH.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
}

```

```

# Initialisation of the quantile coefficients for the Bayesian methods #

MCMCLH.quant=vector("list",S)
INLAiid.quant=vector("list",S)
INLA.quant=vector("list",S)

for (s in 1:s)
{
  MCMCLH.quant[[s]]=vector("list",B)
  INLAiid.quant[[s]]=vector("list",B)
  INLA.quant[[s]]=vector("list",B)
}

for (s in 1:s)
{
  for (b in 1:B)
  {
    MCMCLH.quant[[s]][[b]]=matrix(0,ncol=P,nrow=6)
    INLAiid.quant[[s]][[b]]=matrix(0,ncol=P,nrow=6)
    INLA.quant[[s]][[b]]=matrix(0,ncol=P,nrow=6)
  }
}

# Generation of the spatial matrices with exponential autocorrelation #

coords=as.data.frame(cbind(x,y))

mat.exp1=corExp(value=c(0.7,0.1),form=~x+y,nugget=T)
mat.exp1=Initialize(mat.exp1,coords)
mat.exp1=corMatrix(mat.exp1)

mat.exp2=corExp(value=c(3,0.1),form=~x+y,nugget=T)
mat.exp2=Initialize(mat.exp2,coords)
mat.exp2=corMatrix(mat.exp2)

mat.exp3=corExp(value=c(5,0.1),form=~x+y,nugget=T)
mat.exp3=Initialize(mat.exp3,coords)
mat.exp3=corMatrix(mat.exp3)

for (b in 1:B)
{
  set.seed(b)

## Generation of the covariates ##

sigma=0.7

X0=mvnrnorm(n=1,rep(0,n),sigma^2*diag(1,n),tol=1e-6,empirical=FALSE)
X1=mvnrnorm(n=1,rep(0,n),sigma^2*mat.exp2,tol=1e-6,empirical=FALSE)
X2=mvnrnorm(n=1,rep(0,n),sigma^2*mat.exp1,tol=1e-6,empirical=FALSE)
X3=mvnrnorm(n=1,rep(0,n),sigma^2*mat.exp2,tol=1e-6,empirical=FALSE)+rnorm(n, sd=1)
X4=mvnrnorm(n=1,rep(0,n),sigma^2*mat.exp1,tol=1e-6,empirical=FALSE)+rnorm(n, sd=1)

beta4bis=rep(beta[4]+0.5*(seq(0,1,len=I)-0.5),each=I)
beta5bis=rep(beta[5]+0.2*(seq(0,1,len=I)-0.5),times=I)

## Generation of the response variables ##

# Scenario 1 #
Y1=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta[4]*X3+beta[5]*X4
for (i in 1:n) { Y1[i]=rpois(1,exp(eta[i])) }

# Scenario 2 #

```

```

Y2=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta[4]*X3+beta[5]*X4
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp1,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y2[i]=rpois(1,exp(eta[i])+eta.error[i])) }

# Scenario 3 #
Y3=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta[4]*X3+beta[5]*X4
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp2,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y3[i]=rpois(1,exp(eta[i])+eta.error[i])) }

# Scenario 4 #
Y4=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta[4]*X3+beta[5]*X4
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp3,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y4[i]=rpois(1,exp(eta[i])+eta.error[i])) }

# Scenario 5 #
Y5=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta[4]*X3+beta[5]*X4
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp1,tol=1e-6,empirical=FALSE)+rnorm(n, sd=0.3)
for (i in 1:n) { Y5[i]=rpois(1,exp(eta[i])+eta.error[i])) }

# Scenario 6 #
Y6=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta[4]*X3+beta[5]*X4
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp2,tol=1e-6,empirical=FALSE)+rnorm(n, sd=0.3)
for (i in 1:n) { Y6[i]=rpois(1,exp(eta[i])+eta.error[i])) }

# Scenario 7 #
Y7=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta[4]*X3+beta[5]*X4
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp3,tol=1e-6,empirical=FALSE)+rnorm(n, sd=0.3)
for (i in 1:n) { Y7[i]=rpois(1,exp(eta[i])+eta.error[i])) }

# Scenario 8 #
Y8=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta4bis*X3+beta5bis*X4
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp3,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y8[i]=rpois(1,exp(eta[i])+eta.error[i])) }

# Scenario 9 #
Y9=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta[4]*X3+beta[5]*X4
eta.error1=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp3,tol=1e-6,empirical=FALSE)
eta.error2=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp1,tol=1e-6,empirical=FALSE)
eta.error=eta.error1*seq(0,1,len=I)+eta.error2*seq(1,0,len=I)
for (i in 1:n) { Y9[i]=rpois(1,exp(eta[i])+eta.error[i])) }

# Scenario 10 #
Y10=rep(0,n)
eta=beta[1]+beta[2]*X1+beta[3]*X2+beta4bis*X3+beta5bis*X4
eta.error1=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp3,tol=1e-6,empirical=FALSE)
eta.error2=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp1,tol=1e-6,empirical=FALSE)
eta.error=eta.error1*seq(0,1,len=I)+eta.error2*seq(1,0,len=I)
for (i in 1:n) { Y10[i]=rpois(1,exp(eta[i])+eta.error[i])) }

my.data=as.data.frame(cbind(x,y,X1,X2,X3,X4,Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9,Y10))

## Inference ##
# Y~X1+X2+X3+X4 #

# GLM #

GLM=try(glm(Y1~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[1]][b,]=rep(NA,2*p+HP) }

```

```

else { GLM.coef[[1]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y2~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[2]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[2]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y3~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[3]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[3]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y4~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[4]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[4]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y5~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[5]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[5]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y6~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[6]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[6]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y7~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[7]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[7]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y8~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[8]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[8]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y9~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[9]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[9]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y10~X1+X2+X3+X4,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[10]][b,]=rep(NA,2*p+hp) }
else { GLM.coef[[10]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

# GLMquasi #

GLMquasi=try(glm(Y1~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[1]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[1]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y2~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[2]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[2]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y3~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[3]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[3]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y4~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[4]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[4]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y5~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[5]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[5]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y6~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))

```

```

if (is(GLMquasi,"try-error")) { GLMquasi.coef[[6]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[6]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y7~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[7]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[7]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y8~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[8]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[8]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y9~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[9]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[9]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

GLMquasi=try(glm(Y10~X1+X2+X3+X4,family=quasipoisson(link="log"),data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[10]][b,]=rep(NA,2*p+hp) }
else { GLMquasi.coef[[10]][b,]=c(summary(GLMquasi)$coef[,1],summary(GLMquasi)$coef[,2],NA,
NA) }

# GLMMPQLiid #

GLMMPQLiid=try(glmmPQL(fixed=Y1~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[1]][b,]=rep(NA,2*p+hp) }
else { GLMMPQLiid.coef[[1]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y2~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[2]][b,]=rep(NA,2*p+hp) }
else { GLMMPQLiid.coef[[2]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y3~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[3]][b,]=rep(NA,2*p+hp) }
else { GLMMPQLiid.coef[[3]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y4~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[4]][b,]=rep(NA,2*p+hp) }
else { GLMMPQLiid.coef[[4]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y5~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[5]][b,]=rep(NA,2*p+hp) }
else { GLMMPQLiid.coef[[5]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y6~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[6]][b,]=rep(NA,2*p+hp) }
else { GLMMPQLiid.coef[[6]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y7~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[7]][b,]=rep(NA,2*p+hp) }

```

```

else { GLMMPQLiid.coef[[7]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
    GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y8~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
    =my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[8]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[8]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
    GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y9~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),data
    =my.data,control=list(lmeControl(returnObject=TRUE,opt="optim")))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[9]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[9]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
    GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y10~X1+X2+X3+X4,random=~1|obs,family=poisson(link="log"),
    data=my.data,control=list(lmeControl(returnObject=TRUE,opt="optim")))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[10]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[10]][b,]=c(summary(GLMMPQLiid)$coef$fixed,sqrt(diag(summary(
    GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

# GAMMiid #

GAMMiid=try(gamm(Y1~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
    link="log"),data=my.data)
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[1]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[1]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
    NA,NA) }

GAMMiid=try(gamm(Y2~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
    link="log"),data=my.data)
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[2]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[2]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
    NA,NA) }

GAMMiid=try(gamm(Y3~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
    link="log"),data=my.data)
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[3]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[3]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
    NA,NA) }

GAMMiid=try(gamm(Y4~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
    link="log"),data=my.data)
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[4]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[4]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
    NA,NA) }

GAMMiid=try(gamm(Y5~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
    link="log"),data=my.data)
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[5]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[5]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
    NA,NA) }

GAMMiid=try(gamm(Y6~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
    link="log"),data=my.data)
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[6]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[6]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
    NA,NA) }

GAMMiid=try(gamm(Y7~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
    link="log"),data=my.data)
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[7]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[7]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
    NA,NA) }

```

```

GAMMiid=try(gamm(Y8~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
  link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[8]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[8]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
  NA,NA) }

GAMMiid=try(gamm(Y9~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
  link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[9]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[9]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
  NA,NA) }

GAMMiid=try(gamm(Y10~X1+X2+X3+X4+te(x,y,bs=c("tp","tp")))+s(obs,bs="re"),family=poisson(
  link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[10]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[10]][b,]=c(coef(GAMMiid$gam)[1:P],sqrt(diag(vcov(GAMMiid$gam)))[1:P],
  ],NA,NA) }

# INLAiid #

INLAiid=try(inla(Y1~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[1]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[1]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[1]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
  NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y2~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[2]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[2]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[2]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
  NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y3~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[3]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[3]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[3]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
  NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y4~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[4]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[4]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[4]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
  NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y5~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1)))

```

```

,family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[5]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[5]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[5]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
    NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y6~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[6]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[6]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[6]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
    NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y7~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[7]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[7]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[7]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
    NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y8~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[8]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[8]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[8]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
    NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y9~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[9]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[9]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[9]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
    NA,INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y10~X1+X2+X3+X4+f(obs,model="iid",hyper=list(theta=list(param=c(1,0.01),
  prior="loggamma",initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=0.0001))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) { INLAiid.coef[[10]][b,]=
  rep(NA,2*P+HP) }
else { INLAiid.coef[[10]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],sqrt
  (1/INLAiid$summary.hyperpar[1,1]),NA)
  for (j in 1:P) { INLAiid.quant[[10]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,
    NA,INLAiid$summary.fixed[j,5],NA) } }

# GLMMPQL #

```

```

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y1~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[1]][b,]=rep(NA,2*p+hp) }
else {
  struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
  struct.cor=Initialize(struct.cor,as(my.data,"data.frame") [,c("x","y")])
  GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
  if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[1]][b,]=rep(NA,2*p+hp) }
  else { GLMMPQL.coef[[1]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
    $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y2~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[2]][b,]=rep(NA,2*p+hp) }
else {
  struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
  struct.cor=Initialize(struct.cor,as(my.data,"data.frame") [,c("x","y")])
  GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
  if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[2]][b,]=rep(NA,2*p+hp) }
  else { GLMMPQL.coef[[2]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
    $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y3~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[3]][b,]=rep(NA,2*p+hp) }
else {
  struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
  struct.cor=Initialize(struct.cor,as(my.data,"data.frame") [,c("x","y")])
  GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
  if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[3]][b,]=rep(NA,2*p+hp) }
  else { GLMMPQL.coef[[3]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
    $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y4~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[4]][b,]=rep(NA,2*p+hp) }
else {
  struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
  struct.cor=Initialize(struct.cor,as(my.data,"data.frame") [,c("x","y")])
  GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
  if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[4]][b,]=rep(NA,2*p+hp) }
  else { GLMMPQL.coef[[4]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
    $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y5~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[5]][b,]=rep(NA,2*p+hp) }
else {
  struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
  struct.cor=Initialize(struct.cor,as(my.data,"data.frame") [,c("x","y")])
  GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))

```

```

if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[5]][b,]=rep(NA,2*p+hp) }
else { GLMMPQL.coef[[5]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
    $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y6~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[6]][b,]=rep(NA,2*p+hp) }
else {
    struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
    struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
    GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
    if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[6]][b,]=rep(NA,2*p+hp) }
    else { GLMMPQL.coef[[6]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
        $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y7~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[7]][b,]=rep(NA,2*p+hp) }
else {
    struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
    struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
    GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
    if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[7]][b,]=rep(NA,2*p+hp) }
    else { GLMMPQL.coef[[7]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
        $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y8~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[8]][b,]=rep(NA,2*p+hp) }
else {
    struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
    struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
    GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
    if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[8]][b,]=rep(NA,2*p+hp) }
    else { GLMMPQL.coef[[8]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
        $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y9~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[9]][b,]=rep(NA,2*p+hp) }
else {
    struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
    struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
    GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
    if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[9]][b,]=rep(NA,2*p+hp) }
    else { GLMMPQL.coef[[9]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
        $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y10~X1+X2+X3+X4,random=~1|group,family=poisson(link="log"),data=
    my.data,control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[10]][b,]=rep(NA,2*p+hp) }

```

```

else {
  struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
  struct.cor=Initialize(struct.cor,as(my.data,"data.frame") [,c("x","y")])
  GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
  if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[10]][b,]=rep(NA,2*P+HP) }
  else { GLMMPQL.coef[[10]][b,]=c(summary(GLMMPQL)$coef$fixed,sqrt(diag(summary(GLMMPQL)
    $varFix)),as.numeric(VarCorr(GLMMPQL)[1,2]),NA) }
}

# GAMM #

GAMM=try(gamm(Y1~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[1]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[1]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y2~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[2]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[2]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y3~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[3]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[3]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y4~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[4]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[4]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y5~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[5]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[5]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y6~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[6]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[6]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y7~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[7]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[7]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y8~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[8]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[8]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y9~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
    lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GAMM,"try-error")) { GAMM.coef[[9]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[9]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y10~X1+X2+X3+X4+te(x,y,bs=c("tp","tp"))+s(obs,bs="re"),

```

```

correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
  lmeControl(returnObject=TRUE,opt="optim")))
if (is(GAMM,"try-error")) { GAMM.coef[[10]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[10]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

# INLA #

INLA=try(inla(Y1~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
  prec=list(param=c(1,0.01),initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[1]][b,]=rep(NA,2*P
  +HP) }
else { INLA.coef[[1]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
  INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
  for (j in 1:P) { INLA.quant[[1]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
    INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y2~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
  prec=list(param=c(1,0.01),initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[2]][b,]=rep(NA,2*P
  +HP) }
else { INLA.coef[[2]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
  INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
  for (j in 1:P) { INLA.quant[[2]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
    INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y3~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
  prec=list(param=c(1,0.01),initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[3]][b,]=rep(NA,2*P
  +HP) }
else { INLA.coef[[3]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
  INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
  for (j in 1:P) { INLA.quant[[3]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
    INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y4~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
  prec=list(param=c(1,0.01),initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[4]][b,]=rep(NA,2*P
  +HP) }
else { INLA.coef[[4]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
  INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
  for (j in 1:P) { INLA.quant[[4]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
    INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y5~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
  prec=list(param=c(1,0.01),initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[5]][b,]=rep(NA,2*P
  +HP) }
else { INLA.coef[[5]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
  INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
  for (j in 1:P) { INLA.quant[[5]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
    INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y6~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
  prec=list(param=c(1,0.01),initial=1))),family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
  list(h=1e-4)))

```

```

if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[6]][b,]=rep(NA,2*P
+HP) }
else { INLA.coef[[6]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
for (j in 1:P) { INLA.quant[[6]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y7~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[7]][b,]=rep(NA,2*P
+HP) }
else { INLA.coef[[7]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
for (j in 1:P) { INLA.quant[[7]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y8~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[8]][b,]=rep(NA,2*P
+HP) }
else { INLA.coef[[8]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
for (j in 1:P) { INLA.quant[[8]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y9~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[9]][b,]=rep(NA,2*P
+HP) }
else { INLA.coef[[9]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
for (j in 1:P) { INLA.quant[[9]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
INLA$summary.fixed[j,5],NA) } }

INLA=try(inla(Y10~X1+X2+X3+X4+f(obs,model="besagproper",graph.file="grid.dat",hyper=list(
prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),control.inla=
list(h=1e-4)))
if (is(INLA,"try-error") | length(INLA$summary.fixed)==0) { INLA.coef[[10]][b,]=rep(NA,2*
P+HP) }
else { INLA.coef[[10]][b,]=c(INLA$summary.fixed[,1],INLA$summary.fixed[,2],
INLA$summary.hyperpar[1,1],INLA$summary.hyperpar[2,1])
for (j in 1:P) { INLA.quant[[10]][[b]][,j]=c(NA,INLA$summary.fixed[j,3],NA,NA,
INLA$summary.fixed[j,5],NA) } }

# MCMCLH #

my.S.scale=0.0003
my.phi.scale=1
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y1
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
=30000,n.iter=70000,phi.start=0)
MCMCLH=try(poiss.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,

```

```

model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[1]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[1]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
  MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
  MCMCLH$posterior$phi$mean)
  for (j in 1:P) { MCMCLH.quant[[1]][[b]][,j]=quantile(
    MCMCLH$posterior$beta$sample[j,],c
    (0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0005
my.phi.scale=0.1
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y2
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
  sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
  =30000,n.iter=70000,phi.start=0)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[2]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[2]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
  MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
  MCMCLH$posterior$phi$mean)
  for (j in 1:P) { MCMCLH.quant[[2]][[b]][,j]=quantile(
    MCMCLH$posterior$beta$sample[j,],c
    (0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0005
my.phi.scale=0.1
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y3
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
  sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
  =30000,n.iter=70000,phi.start=0)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[3]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[3]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
  MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
  MCMCLH$posterior$phi$mean)
  for (j in 1:P) { MCMCLH.quant[[3]][[b]][,j]=quantile(
    MCMCLH$posterior$beta$sample[j,],c
    (0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0005
my.phi.scale=0.1
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y4
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
  sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
  =30000,n.iter=70000,phi.start=0)
my.output=list(sim.predict=T)

```

```

MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model,locations=cbind(my.data$x,my.data$y)))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[4]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[4]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[4]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],c
(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0005
my.phi.scale=0.02
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y5
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
=30000,n.iter=70000,phi.start=0)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[5]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[5]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[5]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],c
(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0002
my.phi.scale=0.02
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y6
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
=30000,n.iter=70000,phi.start=0)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[6]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[6]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[6]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],c
(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0002
my.phi.scale=0.02
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y7
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
=30000,n.iter=70000,phi.start=0)

```

```

my.output=list(sim.predict=T)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model,output=my.output,locations=cbind(my.data$x,my.data$y)))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[7]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[7]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
  MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
  MCMCLH$posterior$phi$mean)
  for (j in 1:P) { MCMCLH.quant[[7]][[b]][,j]=quantile(
    MCMCLH$posterior$beta$sample[j,],c
    (0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0005
my.phi.scale=0.2
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y8
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
  sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
  =30000,n.iter=70000,phi.start=0)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[8]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[8]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
  MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
  MCMCLH$posterior$phi$mean)
  for (j in 1:P) { MCMCLH.quant[[8]][[b]][,j]=quantile(
    MCMCLH$posterior$beta$sample[j,],c
    (0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0005
my.phi.scale=0.5
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y9
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
  sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
  =30000,n.iter=70000,phi.start=0)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[9]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[9]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
  MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
  MCMCLH$posterior$phi$mean)
  for (j in 1:P) { MCMCLH.quant[[9]][[b]][,j]=quantile(
    MCMCLH$posterior$beta$sample[j,],c
    (0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0005
my.phi.scale=0.2
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y10
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~X1+X2+X3+X4,trend.l=~X1+X2+X3+X4,cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
  sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,10,l=500))

```

```

my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,burn.in
  =30000,n.iter=70000,phi.start=0)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[10]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[10]][b,]=c(MCMCLH$posterior$beta$mean,sqrt(diag(
  MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),
  MCMCLH$posterior$phi$mean)
  for (j in 1:P) { MCMCLH.quant[[10]][[b]][,j]=quantile(
    MCMCLH$posterior$beta$sample[j,],c
    (0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }
}

```

Appendix 4: R code to compute the ungridded scenarios

```
setwd("P:/")

library(MASS)
library(pixmap)
library(sp)
library(spdep)
library(INLA)
library(mgcv)
library(RandomFields)
library(gstat)
library(geoR)
library(geoRglm)

# "unfactor" function #

unfactor=function(factors)
{
  return(levels(factors)[factors])
}

# Geodata of the plots in the forest sites #

runif(1)
data.dendroeco=read.csv("data_dendroeco.csv",sep=";")

x=data.dendroeco$X_L2E
y=data.dendroeco$Y_L2E
ind=!is.na(x)
x=x[ind]/1000
y=y[ind]/1000

placette=unfactor(data.dendroeco$numpla.GNB[ind])
n.placette=length(placette)      # number of plots

massif=unfactor(data.dendroeco$Massif[ind])
n.massif=length(unique(massif)) # number of forest sites

gestion=unfactor(data.dendroeco$gestion[ind])
gestion[gestion=="MAN"]=1
gestion[gestion=="UNM"]=2

n.placette_massif=rep(0,n.massif)
for (i in 1:n.massif) { n.placette_massif[i]=length(which(massif==unique(massif)[i])) }
sum(n.placette_massif)

placette=as.factor(1:n.placette)
massif=as.factor(rep(1:n.massif,n.placette_massif))
gestion=as.factor(as.numeric(gestion))

my.data.placette=as.data.frame(cbind(placette,massif,gestion,x,y))

## Neighbourhood matrices for INLA ##

my.data=my.data.placette
attach(my.data)

## Distance matrix for each site ##

list.mat.dist=vector("list",n.massif)
for (m in 1:n.massif) { list.mat.dist[[m]]=matrix(0,nrow=n.placette_massif[m],
ncol=n.placette_massif[m]) }
```

```

for (m in 1:n.massif)
{
  for (i in 1:n.placette_massif[m])
  {
    for (j in 1:n.placette_massif[m])
    {
      list.mat.dist[[m]][i,j]=sqrt((x[massif==m][i]-
      x[massif==m][j])^2+(y[massif==m][i]-y[massif==m][j])^2)
    }
  }
}

for (m in 1:n.massif)
{
  colnames(list.mat.dist[[m]])=rep("a",n.placette_massif[m])
  rownames(list.mat.dist[[m]])=rep("a",n.placette_massif[m])

  for (i in 1:n.placette_massif[m])
  {
    colnames(list.mat.dist[[m]])[i]=placette[massif==m][i]
    rownames(list.mat.dist[[m]])[i]=placette[massif==m][i]
  }
}

min=rep(0,n.massif)
max=rep(0,n.massif)
mean=rep(0,n.massif)
for (s in 1:n.massif)
{
  min[s]=min(list.mat.dist[[s]][list.mat.dist[[s]]!=0])
  max[s]=max(list.mat.dist[[s]][list.mat.dist[[s]]!=0])
  mean[s]=mean(list.mat.dist[[s]][list.mat.dist[[s]]!=0])
}

mean(min)
mean(max)
mean(mean)

# Limit distance to define the neighbourhood #

dist.lim=c(0,n.massif)
dist.lim=rep(1.5,n.massif) # 1.5 or 5

# List of the neighbours for each plot #

list.voisin=vector("list",n.massif)
n.voisin=vector("list",n.massif)
for (m in 1:n.massif) { list.voisin[[m]]=vector("list") ;
n.voisin[[m]]=vector("list") }

for (m in 1:n.massif)
{
  for (i in 1:n.placette_massif[m])
  {
    list.voisin[[m]][[i]]=as.numeric(colnames(list.mat.dist[[m]]))
    [list.mat.dist[[m]][i,]!=0 & list.mat.dist[[m]][i,<dist.lim[m]]]
    n.voisin[[m]][[i]]=length(list.voisin[[m]][[i]])
  }
}

for (m in 1:n.massif)
{
  names(list.voisin[[m]])=unfactor(placette[massif==m])
  names(n.voisin[[m]])=unfactor(placette[massif==m])
}

```

```

# mat.nb #

max.voisin=rep(0,n.massif)
for (m in 1:n.massif)
{
  max.voisin[m]=max(as.data.frame(n.voisin[[m]]))
}
max.voisin=max(max.voisin)

mat.nb=matrix(0,ncol=max.voisin+2,nrow=n.placette)
mat.nb[,1]=1:n.placette

for (i in 1:n.placette)
{
  m=massif[placette==i]
  if (length(list.voisin[[m]][[as.character(i)]])!=0) {
    mat.nb[i,2:(1+n.voisin[[m]][[as.character(i)]])]=
      list.voisin[[m]][[as.character(i)]] }
  mat.nb[i,max.voisin+2]=n.voisin[[m]][[as.character(i)]]
}

mat.nb=as.data.frame(mat.nb)

# mat.adj #

mat.adj=matrix(0,ncol=n.placette,nrow=n.placette)

for (i in 1:n.placette)
{
  ind.1=mat.nb[i,-(max.voisin+2)][mat.nb[i,-(max.voisin+2)]!=0]
  mat.adj[i,ind.1]=1
}

mat.adj=as.data.frame(mat.adj)

# Generation of the neighbourhood datafile for INLA #

INLA1.nb=matrix(0,ncol=max.voisin+2,nrow=n.placette)
INLA1.nb[,1]=1:n.placette

for (i in 1:n.placette)
{
  m=massif[placette==i]
  INLA1.nb[i,2]=n.voisin[[m]][[as.character(i)]]
  if (length(list.voisin[[m]][[as.character(i)]])!=0) {
    INLA1.nb[i,3:(2+n.voisin[[m]][[as.character(i)]])]=
      list.voisin[[m]][[as.character(i)]] }
}
write.table(INLA1.nb,file="GNBsimul.dat",append=F,sep=" ",
eol="\r\n",na="",row.names=F,col.names=F)

## UNGRID simulations ##

my.data=my.data.placette
n=n.placette      # number of observations (plots)

B=1000  # number of replicates
M=10    # number of methods
HP=2    # number of hyperparameters
P=5    # number of regression parameters (intercept + 4 covariates)
S=6    # number of scenarios

# Initialisation of the coefficients #

GLM.coef=vector("list",S)

```

```

GLMquasi.coef=vector("list",S)
GLMMPQLiid.coef=vector("list",S)
GAMMiid.coef=vector("list",S)
INLAiid.coef=vector("list",S)
GLMMPQL.coef=vector("list",S)
GAMM.coef=vector("list",S)
INLA1.coef=vector("list",S)
INLA2.coef=vector("list",S)
MCMCLH.coef=vector("list",S)

for (s in 1:S)
{
  GLM.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GLMquasi.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GLMMPQLiid.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GAMMiid.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  INLAiid.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GLMMPQL.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  GAMM.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  INLA1.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  INLA2.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
  MCMCLH.coef[[s]]=matrix(0,ncol=2*P+HP,nrow=B)
}

# Initialisation of the quantile coefficients for the Bayesian methods #

MCMCLH.quant=vector("list",S)
INLAiid.quant=vector("list",S)
INLA1.quant=vector("list",S)
INLA2.quant=vector("list",S)

for (s in 1:s)
{
  MCMCLH.quant[[s]]=vector("list",B)
  INLAiid.quant[[s]]=vector("list",B)
  INLA1.quant[[s]]=vector("list",B)
  INLA2.quant[[s]]=vector("list",B)
}

for (s in 1:s)
{
  for (b in 1:B)
  {
    MCMCLH.quant[[s]][[b]]=matrix(0,ncol=P,nrow=6)
    INLAiid.quant[[s]][[b]]=matrix(0,ncol=P,nrow=6)
    INLA1.quant[[s]][[b]]=matrix(0,ncol=P,nrow=6)
    INLA2.quant[[s]][[b]]=matrix(0,ncol=P,nrow=6)
  }
}

# Generation of the spatial matrices with exponential inter-plot autocorrelation #

coords=as.data.frame(cbind(x,y))

mat.exp1=corExp(value=c(0.7,0.1),form=~x+y,nugget=T)
mat.exp1=Initialize(mat.exp1,coords)
mat.exp1=corMatrix(mat.exp1)

mat.exp2=corExp(value=c(1.5,0.1),form=~x+y,nugget=T)
mat.exp2=Initialize(mat.exp2,coords)
mat.exp2=corMatrix(mat.exp2)

mat.exp3=corExp(value=c(3,0.1),form=~x+y,nugget=T)
mat.exp3=Initialize(mat.exp3,coords)
mat.exp3=corMatrix(mat.exp3)

```

```

for (i in 1:n)
{
  for (j in 1:n)
  {
    if(massif[i]!=massif[j])
    {
      mat.exp1[i,j]=0
      mat.exp2[i,j]=0
      mat.exp3[i,j]=0
    }
  }
}

for (b in 1:B)
{
  set.seed(b)

## Generation of the covariates ##

sigma=0.7
X1=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp2,tol=1e-6,empirical=FALSE)
X2=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp1,tol=1e-6,empirical=FALSE)
X3=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp2,tol=1e-6,empirical=FALSE)+rnorm(n,sd=1)

## Design matrix ##

X.cte=rep(1,n)
X.gestion=rep(0,n) ; X.gestion[gestion==2]=1
X=cbind(X.cte,X.gestion,X1,X2,X3)

## Generation of the response variables ##

beta=c(1.5,0.5,0.3,0.3,0.3)      # vector of the regression parameters
eta=X%*%beta

# Scenario 1 #
Y1=rep(0,n)
for (i in 1:n) { Y1[i]=rpois(1,exp(eta[i])) }

# Scenario 2 #
Y2=rep(0,n)
sigma=0.7
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp1,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y2[i]=rpois(1,exp(eta[i]+eta.error[i])) }

# Scenario 3 #
Y3=rep(0,n)
sigma=0.7
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp2,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y3[i]=rpois(1,exp(eta[i]+eta.error[i])) }

# Scenario 4 #
Y4=rep(0,n)
sigma=0.7
eta.error=mvrnorm(n=1,rep(0,n),sigma^2*mat.exp3,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y4[i]=rpois(1,exp(eta[i]+eta.error[i])) }

# Scenario 5 #

Y5=rep(0,n)
coords=as.data.frame(cbind(my.data$x,my.data$y))
phi=runif(n.massif,0.2,3)      # one spatial scale for each forest site
list.mat.corr=vector("list",n.massif)
for (m in 1:n.massif) { list.mat.corr[[m]]=matrix(0,nrow=n.placette_massif[m],
ncol=n.placette_massif[m]) }

```

```

for (m in 1:n.massif)
{
  coords=as.data.frame(cbind(my.data$x[my.data$massif==m],
  my.data$y[my.data$massif==m]))
  list.mat.corr[[m]]=sigma^2*corMatrix(Initialize(corExp(value=c(phi[m],0.1),
  form=~coords$V1+coords$V2,nugget=T),coords))
}
for (m in 1:n.massif)
{
  colnames(list.mat.corr[[m]])=rep("a",n.placette_massif[m])
  rownames(list.mat.corr[[m]])=rep("a",n.placette_massif[m])

  for (i in 1:n.placette_massif[m])
  {
    colnames(list.mat.corr[[m]])[i]=unfactor(placette$massif==m)[i]
    rownames(list.mat.corr[[m]])[i]=unfactor(placette$massif==m)[i]
  }
}
mat.corr=matrix(0,ncol=n.placette,nrow=n.placette)
for (m in 1:(n.massif-1))
{
  if (m==1) { mat.corr=list.mat.corr[[1]] }
  mat.corr=adiag(mat.corr,list.mat.corr[[m+1]])
}
eta.error=mvrnorm(n=1,rep(0,n),mat.corr,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y5[i]=rpois(1,exp(eta[i]+eta.error[i])) }

# Scenario 6 #

Y6=rep(0,n)
eta.error=mvrnorm(n=1,rep(0,n),(sigma/sqrt(3))^2*mat.exp1,tol=1e-6,empirical=FALSE)+
  mvrnorm(n=1,rep(0,n),(sigma/sqrt(3))^2*mat.exp2,tol=1e-6,empirical=FALSE)+mvrnorm(n
  =1,rep(0,n),(sigma/sqrt(3))^2*mat.exp3,tol=1e-6,empirical=FALSE)
for (i in 1:n) { Y6[i]=rpois(1,exp(eta[i]+eta.error[i])) }

my.data=as.data.frame(cbind(my.data$placette,Y1,Y2,Y3,Y4,Y5,Y6))

## Inference ##
# Y~gestion+X1+X2+X3 #

# GLM #

GLM=try(glm(Y1~gestion+X1+X2+X3,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[1]][b,]=rep(NA,2*P+HP) }
else { GLM.coef[[1]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y2~gestion+X1+X2+X3,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[2]][b,]=rep(NA,2*P+HP) }
else { GLM.coef[[2]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y3~gestion+X1+X2+X3,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[3]][b,]=rep(NA,2*P+HP) }
else { GLM.coef[[3]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y4~gestion+X1+X2+X3,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[4]][b,]=rep(NA,2*P+HP) }
else { GLM.coef[[4]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y5~gestion+X1+X2+X3,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[5]][b,]=rep(NA,2*P+HP) }
else { GLM.coef[[5]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

GLM=try(glm(Y6~gestion+X1+X2+X3,family=poisson(link="log"),data=my.data))
if (is(GLM,"try-error")) { GLM.coef[[6]][b,]=rep(NA,2*P+HP) }
else { GLM.coef[[6]][b,]=c(summary(GLM)$coef[,1],summary(GLM)$coef[,2],NA,NA) }

```

```

# GLMquasi #

GLMquasi=try(glm(Y1~gestion+X1+X2+X3,family=quasipoisson(link="log"),
data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[1]][b,]=rep(NA,2*P+HP) }
else { GLMquasi.coef[[1]][b,]=c(summary(GLMquasi)$coef[,1],
summary(GLMquasi)$coef[,2],NA,NA) }

GLMquasi=try(glm(Y2~gestion+X1+X2+X3,family=quasipoisson(link="log"),
data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[2]][b,]=rep(NA,2*P+HP) }
else { GLMquasi.coef[[2]][b,]=c(summary(GLMquasi)$coef[,1],
summary(GLMquasi)$coef[,2],NA,NA) }

GLMquasi=try(glm(Y3~gestion+X1+X2+X3,family=quasipoisson(link="log"),
,data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[3]][b,]=rep(NA,2*P+HP) }
else { GLMquasi.coef[[3]][b,]=c(summary(GLMquasi)$coef[,1],
summary(GLMquasi)$coef[,2],NA,NA) }

GLMquasi=try(glm(Y4~gestion+X1+X2+X3,family=quasipoisson(link="log"),
data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[4]][b,]=rep(NA,2*P+HP) }
else { GLMquasi.coef[[4]][b,]=c(summary(GLMquasi)$coef[,1],
summary(GLMquasi)$coef[,2],NA,NA) }

GLMquasi=try(glm(Y5~gestion+X1+X2+X3,family=quasipoisson(link="log"),
data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[5]][b,]=rep(NA,2*P+HP) }
else { GLMquasi.coef[[5]][b,]=c(summary(GLMquasi)$coef[,1],
summary(GLMquasi)$coef[,2],NA,NA) }

GLMquasi=try(glm(Y6~gestion+X1+X2+X3,family=quasipoisson(link="log"),
data=my.data))
if (is(GLMquasi,"try-error")) { GLMquasi.coef[[6]][b,]=rep(NA,2*P+HP) }
else { GLMquasi.coef[[6]][b,]=c(summary(GLMquasi)$coef[,1],
summary(GLMquasi)$coef[,2],NA,NA) }

# GLMMPQLiid #

GLMMPQLiid=try(glmmPQL(fixed=Y1~gestion+X1+X2+X3,random=~1|placette,
family=poisson(link="log"),data=my.data,
control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[1]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[1]][b,]=c(summary(GLMMPQLiid)$coef$fixed,
sqrt(diag(summary(GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y2~gestion+X1+X2+X3,random=~1|placette,
family=poisson(link="log"),data=my.data,
control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[2]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[2]][b,]=c(summary(GLMMPQLiid)$coef$fixed,
sqrt(diag(summary(GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y3~gestion+X1+X2+X3,random=~1|placette,
family=poisson(link="log"),data=my.data,
control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[3]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[3]][b,]=c(summary(GLMMPQLiid)$coef$fixed,
sqrt(diag(summary(GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y4~gestion+X1+X2+X3,random=~1|placette,
family=poisson(link="log"),data=my.data,
control=list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[4]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[4]][b,]=c(summary(GLMMPQLiid)$coef$fixed,

```

```

sqrt(diag(summary(GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y5~gestion+X1+X2+X3,random=~1|placette,
family=poisson(link="log"),data=my.data,
control=list(lmeControl(returnObject=TRUE,opt="optim"))))

if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[5]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[5]][b,]=c(summary(GLMMPQLiid)$coef$fixed,
sqrt(diag(summary(GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

GLMMPQLiid=try(glmmPQL(fixed=Y6~gestion+X1+X2+X3,random=~1|placette,
family=poisson(link="log"),data=my.data,
control=list(lmeControl(returnObject=TRUE,opt="optim"))))

if (is(GLMMPQLiid,"try-error")) { GLMMPQLiid.coef[[6]][b,]=rep(NA,2*P+HP) }
else { GLMMPQLiid.coef[[6]][b,]=c(summary(GLMMPQLiid)$coef$fixed,
sqrt(diag(summary(GLMMPQLiid)$varFix)),as.numeric(VarCorr(GLMMPQLiid)[1,2]),NA) }

# GAMMiid #

GAMMiid=try(gamm(Y1~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp"))+s(placette,bs="re"),
family=poisson(link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[1]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[1]][b,]=c(coef(GAMMiid$gam)[1:P],
sqrt(diag(vcov(GAMMiid$gam)))[1:P],NA,NA) }

GAMMiid=try(gamm(Y2~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp"))+s(placette,bs="re"),
family=poisson(link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[2]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[2]][b,]=c(coef(GAMMiid$gam)[1:P],
sqrt(diag(vcov(GAMMiid$gam)))[1:P],NA,NA) }

GAMMiid=try(gamm(Y3~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp"))+s(placette,bs="re"),
family=poisson(link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[3]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[3]][b,]=c(coef(GAMMiid$gam)[1:P],
sqrt(diag(vcov(GAMMiid$gam)))[1:P],NA,NA) }

GAMMiid=try(gamm(Y4~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp"))+s(placette,bs="re"),
family=poisson(link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[4]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[4]][b,]=c(coef(GAMMiid$gam)[1:P],
sqrt(diag(vcov(GAMMiid$gam)))[1:P],NA,NA) }

GAMMiid=try(gamm(Y5~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp"))+s(placette,bs="re"),
family=poisson(link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[5]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[5]][b,]=c(coef(GAMMiid$gam)[1:P],
sqrt(diag(vcov(GAMMiid$gam)))[1:P],NA,NA) }

GAMMiid=try(gamm(Y6~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp"))+s(placette,bs="re"),
family=poisson(link="log"),data=my.data))
if (is(GAMMiid,"try-error")) { GAMMiid.coef[[6]][b,]=rep(NA,2*P+HP) }
else { GAMMiid.coef[[6]][b,]=c(coef(GAMMiid$gam)[1:P],
sqrt(diag(vcov(GAMMiid$gam)))[1:P],NA,NA) }

# INLAiid #

INLAiid=try(inla(Y1~gestion+X1+X2+X3+f(placette,model="iid",hyper=list(theta=
list(param=c(1,0.01),prior="loggamma",initial=1))),family="poisson",data=my.data,
verbose=F,control.predictor=list(compute=T),control.inla=list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) {
INLAiid.coef[[1]][b,]=rep(NA,2*P+HP) }
else { INLAiid.coef[[1]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],
sqrt(1/INLAiid$summary.hyperpar[1,1]),NA)
for (j in 1:P) { INLAiid.quant[[1]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,NA,
INLAiid$summary.fixed[j,5],NA) } }

```

```

INLAiid=try(inla(Y2~gestion+X1+X2+X3+f(placette,model="iid",hyper=list(theta=
list(param=c(1,0.01),prior="loggamma",initial=1))),family="poisson",data=my.data,
verbose=F,control.predictor=list(compute=T),control.inla=list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) {
INLAiid.coef[[2]][b,]=rep(NA,2*p+hp) }
else { INLAiid.coef[[2]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],
sqrt(1/INLAiid$summary.hyperpar[1,1]),NA)
for (j in 1:P) { INLAiid.quant[[2]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,NA,
INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y3~gestion+X1+X2+X3+f(placette,model="iid",hyper=list(theta=
list(param=c(1,0.01),prior="loggamma",initial=1))),family="poisson",data=my.data,
verbose=F,control.predictor=list(compute=T),control.inla=list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) {
INLAiid.coef[[3]][b,]=rep(NA,2*p+hp) }
else { INLAiid.coef[[3]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],
sqrt(1/INLAiid$summary.hyperpar[1,1]),NA)
for (j in 1:P) { INLAiid.quant[[3]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,NA,
INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y4~gestion+X1+X2+X3+f(placette,model="iid",hyper=list(theta=
list(param=c(1,0.01),prior="loggamma",initial=1))),family="poisson",data=my.data,
verbose=F,control.predictor=list(compute=T),control.inla=list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) {
INLAiid.coef[[4]][b,]=rep(NA,2*p+hp) }
else { INLAiid.coef[[4]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],
sqrt(1/INLAiid$summary.hyperpar[1,1]),NA)
for (j in 1:P) { INLAiid.quant[[4]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,NA,
INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y5~gestion+X1+X2+X3+f(placette,model="iid",hyper=list(theta=
list(param=c(1,0.01),prior="loggamma",initial=1))),family="poisson",data=my.data,
verbose=F,control.predictor=list(compute=T),control.inla=list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) {
INLAiid.coef[[5]][b,]=rep(NA,2*p+hp) }
else { INLAiid.coef[[5]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],
sqrt(1/INLAiid$summary.hyperpar[1,1]),NA)
for (j in 1:P) { INLAiid.quant[[5]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,NA,
INLAiid$summary.fixed[j,5],NA) } }

INLAiid=try(inla(Y6~gestion+X1+X2+X3+f(placette,model="iid",hyper=list(theta=
list(param=c(1,0.01),prior="loggamma",initial=1))),family="poisson",data=my.data,
verbose=F,control.predictor=list(compute=T),control.inla=list(h=0.0001)))
if (is(INLAiid,"try-error") | length(INLAiid$summary.fixed)==0) {
INLAiid.coef[[6]][b,]=rep(NA,2*p+hp) }
else { INLAiid.coef[[6]][b,]=c(INLAiid$summary.fixed[,1],INLAiid$summary.fixed[,2],
sqrt(1/INLAiid$summary.hyperpar[1,1]),NA)
for (j in 1:P) { INLAiid.quant[[6]][[b]][,j]=c(NA,INLAiid$summary.fixed[j,3],NA,NA,
INLAiid$summary.fixed[j,5],NA) } }

# GLMMPQL #

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y1~gestion+X1+X2+X3,random=~1|group,family=poisson(link="log"),
data=my.data,control=
list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[1]][b,]=rep(NA,2*p+hp) }
else {
struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[1]][b,]=rep(NA,2*p+hp) }
else { GLMMPQL.coef[[1]][b,]=c(summary(GLMMPQL)$coef$fixed,
sqrt(diag(summary(GLMMPQL)$varFix)),
```

```

as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y2~gestion+X1+X2+X3,random=~1|group,family=poisson(link="log"),
    data=my.data,control=
list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[2]][b,]=rep(NA,2*p+hp) }
else {
struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[2]][b,]=rep(NA,2*p+hp) }
else { GLMMPQL.coef[[2]][b,]=c(summary(GLMMPQL)$coef$fixed,
sqrt(diag(summary(GLMMPQL)$varFix)),
as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y3~gestion+X1+X2+X3,random=~1|group,family=poisson(link="log"),
    data=my.data,control=
list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[3]][b,]=rep(NA,2*p+hp) }
else {
struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[3]][b,]=rep(NA,2*p+hp) }
else { GLMMPQL.coef[[3]][b,]=c(summary(GLMMPQL)$coef$fixed,
sqrt(diag(summary(GLMMPQL)$varFix)),
as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y4~gestion+X1+X2+X3,random=~1|group,family=poisson(link="log"),
    data=my.data,control=
list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[4]][b,]=rep(NA,2*p+hp) }
else {
struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[4]][b,]=rep(NA,2*p+hp) }
else { GLMMPQL.coef[[4]][b,]=c(summary(GLMMPQL)$coef$fixed,
sqrt(diag(summary(GLMMPQL)$varFix)),
as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y5~gestion+X1+X2+X3,random=~1|group,family=poisson(link="log"),
    data=my.data,control=
list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[5]][b,]=rep(NA,2*p+hp) }
else {
struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[5]][b,]=rep(NA,2*p+hp) }
else { GLMMPQL.coef[[5]][b,]=c(summary(GLMMPQL)$coef$fixed,
sqrt(diag(summary(GLMMPQL)$varFix)),
as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

```

```

attach(my.data)
group=factor(rep("a",n))
my.data=cbind(my.data,group)
GLMMPQL=try(glmmPQL(fixed=Y6~gestion+X1+X2+X3,random=~1|group,family=poisson(link="log"),
  data=my.data,control=
list(lmeControl(returnObject=TRUE,opt="optim"))))
if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[6]][b,]=rep(NA,2*P+HP) }
else {
  struct.cor=corSpatial(form=~x+y,nugget=T,type="exponential")
  struct.cor=Initialize(struct.cor,as(my.data,"data.frame")[,c("x","y")])
  GLMMPQL=try(update(GLMMPQL,correlation=struct.cor))
  if (is(GLMMPQL,"try-error")) { GLMMPQL.coef[[6]][b,]=rep(NA,2*P+HP) }
  else { GLMMPQL.coef[[6]][b,]=c(summary(GLMMPQL)$coef$fixed,
  sqrt(diag(summary(GLMMPQL)$varFix)),
  as.numeric(VarCorr(GLMMPQL)[1,2]),NA) } }

# GAMM #

GAMM=try(gamm(Y1~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp")))+s(my.data$placette,bs="re"),
  correlation=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=
  list(lmeControl(returnObject=TRUE,opt="optim")))
if (is(GAMM,"try-error")) { GAMM.coef[[1]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[1]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y2~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp")))+s(placette,bs="re"),correlation
=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
  lmeControl(returnObject=TRUE,opt="optim")))
if (is(GAMM,"try-error")) { GAMM.coef[[2]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[2]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y3~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp")))+s(placette,bs="re"),correlation
=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
  lmeControl(returnObject=TRUE,opt="optim")))
if (is(GAMM,"try-error")) { GAMM.coef[[3]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[3]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y4~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp")))+s(placette,bs="re"),correlation
=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
  lmeControl(returnObject=TRUE,opt="optim")))
if (is(GAMM,"try-error")) { GAMM.coef[[4]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[4]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y5~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp")))+s(placette,bs="re"),correlation
=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
  lmeControl(returnObject=TRUE,opt="optim")))
if (is(GAMM,"try-error")) { GAMM.coef[[5]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[5]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

GAMM=try(gamm(Y6~gestion+X1+X2+X3+te(x,y,bs=c("tp","tp")))+s(placette,bs="re"),correlation
=struct.cor,family=poisson(link="log"),data=my.data,method="ML",control=list(
  lmeControl(returnObject=TRUE,opt="optim")))
if (is(GAMM,"try-error")) { GAMM.coef[[6]][b,]=rep(NA,2*P+HP) }
else { GAMM.coef[[6]][b,]=c(coef(GAMM$gam)[1:P],sqrt(diag(vcov(GAMM$gam))[1:P]),NA,NA) }

# INLA1 #

INLA1=try(inla(Y1~gestion+X1+X2+X3+f(placette,model="besagproper",
graph.file="GNBsimul.5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),
control.inla=list(h=1e-4)))
if (is(INLA1,"try-error") | length(INLA1$summary.fixed)==0) { INLA1.coef[[1]][b,]=rep(NA
,2*P+HP) }
else { INLA1.coef[[1]][b,]=c(INLA1$summary.fixed[,1],INLA1$summary.fixed[,2],
INLA1$summary.hyperpar[1,1],
INLA1$summary.hyperpar[2,1]) }

```

```

for (j in 1:P) { INLA1.quant[[1]][[b]][,j]=c(NA,INLA1$summary.fixed[j,3],NA,NA,
INLA1$summary.fixed[j,5],NA) }

INLA1=try(inla(Y2~gestion+X1+X2+X3+f(placette,model="besagproper",
graph.file="GNBsimul.5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),
control.inla=list(h=1e-4)))
if (is(INLA1,"try-error") | length(INLA1$summary.fixed)==0) { INLA1.coef[[2]][b,]=rep(NA
,2*P+HP) }
else { INLA1.coef[[2]][b,]=c(INLA1$summary.fixed[,1],INLA1$summary.fixed[,2],
INLA1$summary.hyperpar[1,1],
INLA1$summary.hyperpar[2,1])
for (j in 1:P) { INLA1.quant[[2]][[b]][,j]=c(NA,INLA1$summary.fixed[j,3],NA,NA,
INLA1$summary.fixed[j,5],NA) } }

INLA1=try(inla(Y3~gestion+X1+X2+X3+f(placette,model="besagproper",
graph.file="GNBsimul.5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),
control.inla=list(h=1e-4)))
if (is(INLA1,"try-error") | length(INLA1$summary.fixed)==0) { INLA1.coef[[3]][b,]=rep(NA
,2*P+HP) }
else { INLA1.coef[[3]][b,]=c(INLA1$summary.fixed[,1],INLA1$summary.fixed[,2],
INLA1$summary.hyperpar[1,1],
INLA1$summary.hyperpar[2,1])
for (j in 1:P) { INLA1.quant[[3]][[b]][,j]=c(NA,INLA1$summary.fixed[j,3],NA,NA,
INLA1$summary.fixed[j,5],NA) } }

INLA1=try(inla(Y4~gestion+X1+X2+X3+f(placette,model="besagproper",
graph.file="GNBsimul.5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),
control.inla=list(h=1e-4)))
if (is(INLA1,"try-error") | length(INLA1$summary.fixed)==0) { INLA1.coef[[4]][b,]=rep(NA
,2*P+HP) }
else { INLA1.coef[[4]][b,]=c(INLA1$summary.fixed[,1],INLA1$summary.fixed[,2],
INLA1$summary.hyperpar[1,1],
INLA1$summary.hyperpar[2,1])
for (j in 1:P) { INLA1.quant[[4]][[b]][,j]=c(NA,INLA1$summary.fixed[j,3],NA,NA,
INLA1$summary.fixed[j,5],NA) } }

INLA1=try(inla(Y5~gestion+X1+X2+X3+f(placette,model="besagproper",
graph.file="GNBsimul.5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),
control.inla=list(h=1e-4)))
if (is(INLA1,"try-error") | length(INLA1$summary.fixed)==0) { INLA1.coef[[5]][b,]=rep(NA
,2*P+HP) }
else { INLA1.coef[[5]][b,]=c(INLA1$summary.fixed[,1],INLA1$summary.fixed[,2],
INLA1$summary.hyperpar[1,1],
INLA1$summary.hyperpar[2,1])
for (j in 1:P) { INLA1.quant[[5]][[b]][,j]=c(NA,INLA1$summary.fixed[j,3],NA,NA,
INLA1$summary.fixed[j,5],NA) } }

INLA1=try(inla(Y6~gestion+X1+X2+X3+f(placette,model="besagproper",
graph.file="GNBsimul.5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),
family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),
control.inla=list(h=1e-4)))
if (is(INLA1,"try-error") | length(INLA1$summary.fixed)==0) { INLA1.coef[[6]][b,]=rep(NA
,2*P+HP) }
else { INLA1.coef[[6]][b,]=c(INLA1$summary.fixed[,1],INLA1$summary.fixed[,2],
INLA1$summary.hyperpar[1,1],
INLA1$summary.hyperpar[2,1])
for (j in 1:P) { INLA1.quant[[6]][[b]][,j]=c(NA,INLA1$summary.fixed[j,3],NA,NA,
INLA1$summary.fixed[j,5],NA) } }

# INLA2 #

INLA2=try(inla(Y1~gestion+X1+X2+X3+f(placette,model="besagproper",

```

```

graph.file="GNBsimu5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),  

family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),  

control.inla=list(h=1e-4))  

if (is(INLA2,"try-error") | length(INLA2$summary.fixed)==0) { INLA2.coef[[1]][b,]=rep(NA  

,2*P+HP) }  

else { INLA2.coef[[1]][b,]=c(INLA2$summary.fixed[,1],INLA2$summary.fixed[,2],  

INLA2$summary.hyperpar[1,1],  

INLA2$summary.hyperpar[2,1])  

for (j in 1:P) { INLA2.quant[[1]][[b]][,j]=c(NA,INLA2$summary.fixed[j,3],NA,NA,  

INLA2$summary.fixed[j,5],NA) } }  

INLA2=try(inla(Y2~gestion+X1+X2+X3+f(placette,model="besagproper",  

graph.file="GNBsimu5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),  

family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),  

control.inla=list(h=1e-4))  

if (is(INLA2,"try-error") | length(INLA2$summary.fixed)==0) { INLA2.coef[[2]][b,]=rep(NA  

,2*P+HP) }  

else { INLA2.coef[[2]][b,]=c(INLA2$summary.fixed[,1],INLA2$summary.fixed[,2],  

INLA2$summary.hyperpar[1,1],  

INLA2$summary.hyperpar[2,1])  

for (j in 1:P) { INLA2.quant[[2]][[b]][,j]=c(NA,INLA2$summary.fixed[j,3],NA,NA,  

INLA2$summary.fixed[j,5],NA) } }  

INLA2=try(inla(Y3~gestion+X1+X2+X3+f(placette,model="besagproper",  

graph.file="GNBsimu5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),  

family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),  

control.inla=list(h=1e-4))  

if (is(INLA2,"try-error") | length(INLA2$summary.fixed)==0) { INLA2.coef[[3]][b,]=rep(NA  

,2*P+HP) }  

else { INLA2.coef[[3]][b,]=c(INLA2$summary.fixed[,1],INLA2$summary.fixed[,2],  

INLA2$summary.hyperpar[1,1],  

INLA2$summary.hyperpar[2,1])  

for (j in 1:P) { INLA2.quant[[3]][[b]][,j]=c(NA,INLA2$summary.fixed[j,3],NA,NA,  

INLA2$summary.fixed[j,5],NA) } }  

INLA2=try(inla(Y4~gestion+X1+X2+X3+f(placette,model="besagproper",  

graph.file="GNBsimu5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),  

family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),  

control.inla=list(h=1e-4))  

if (is(INLA2,"try-error") | length(INLA2$summary.fixed)==0) { INLA2.coef[[4]][b,]=rep(NA  

,2*P+HP) }  

else { INLA2.coef[[4]][b,]=c(INLA2$summary.fixed[,1],INLA2$summary.fixed[,2],  

INLA2$summary.hyperpar[1,1],  

INLA2$summary.hyperpar[2,1])  

for (j in 1:P) { INLA2.quant[[4]][[b]][,j]=c(NA,INLA2$summary.fixed[j,3],NA,NA,  

INLA2$summary.fixed[j,5],NA) } }  

INLA2=try(inla(Y5~gestion+X1+X2+X3+f(placette,model="besagproper",  

graph.file="GNBsimu5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),  

family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),  

control.inla=list(h=1e-4))  

if (is(INLA2,"try-error") | length(INLA2$summary.fixed)==0) { INLA2.coef[[5]][b,]=rep(NA  

,2*P+HP) }  

else { INLA2.coef[[5]][b,]=c(INLA2$summary.fixed[,1],INLA2$summary.fixed[,2],  

INLA2$summary.hyperpar[1,1],  

INLA2$summary.hyperpar[2,1])  

for (j in 1:P) { INLA2.quant[[5]][[b]][,j]=c(NA,INLA2$summary.fixed[j,3],NA,NA,  

INLA2$summary.fixed[j,5],NA) } }  

INLA2=try(inla(Y6~gestion+X1+X2+X3+f(placette,model="besagproper",  

graph.file="GNBsimu5.dat",hyper=list(prec=list(param=c(1,0.01),initial=1))),  

family="poisson",data=my.data,verbose=F,control.predictor=list(compute=T),  

control.inla=list(h=1e-4))  

if (is(INLA2,"try-error") | length(INLA2$summary.fixed)==0) { INLA2.coef[[6]][b,]=rep(NA  

,2*P+HP) }

```

```

else { INLA2.coef[[6]][b,]=c(INLA2$summary.fixed[,1],INLA2$summary.fixed[,2],
    INLA2$summary.hyperpar[1,1],
    INLA2$summary.hyperpar[2,1])
    for (j in 1:P) { INLA2.quant[[6]][[b]][,j]=c(NA,INLA2$summary.fixed[j,3],NA,NA,
        INLA2$summary.fixed[j,5],NA) } }

# MCMCLH #

my.S.scale=0.0009
my.phi.scale=1
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y1
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~gestion+X1+X2+X3,trend.l=~gestion+X1+X2+X3,
cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,5,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,
burn.in=30000,n.iter=70000,phi.start=0)
MCMCLH=try(poiss.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[1]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[1]][b,]=c(MCMCLH$posterior$beta$mean,
sqrt(diag(MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[1]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],
c(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.001
my.phi.scale=0.03
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y2
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~gestion+X1+X2+X3,trend.l=~gestion+X1+X2+X3,
cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,5,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,
burn.in=30000,n.iter=70000,phi.start=0)
MCMCLH=try(poiss.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[2]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[2]][b,]=c(MCMCLH$posterior$beta$mean,
sqrt(diag(MCMCLH$posterior$beta$var)),sqrt(MCMCLH$posterior$sigmasq$mean),MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[2]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],
c(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0008
my.phi.scale=0.05
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y3
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~gestion+X1+X2+X3,trend.l=~gestion+X1+X2+X3,
cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,5,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,

```

```

burn.in=30000,n.iter=70000,phi.start=0)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model,output=my.output,locations=cbind(my.data$x,my.data$y)))
my.output=list(sim.predict=T)
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[3]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[3]][b,]=c(MCMCLH$posterior$beta$mean,
sqrt(diag(MCMCLH$posterior$beta$var)), 
sqrt(MCMCLH$posterior$sigmasq$mean),MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[3]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],
c(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.001
my.phi.scale=0.2
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y4
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~gestion+X1+X2+X3,trend.l=~gestion+X1+X2+X3,
cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,5,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,
burn.in=30000,n.iter=70000)
my.output=list(sim.predict=T)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model,output=my.output,locations=cbind(my.data$x,my.data$y)))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[4]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[4]][b,]=c(MCMCLH$posterior$beta$mean,
sqrt(diag(MCMCLH$posterior$beta$var)), 
sqrt(MCMCLH$posterior$sigmasq$mean),MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[4]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],
c(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.0007
my.phi.scale=0.05
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y5
my.geodata$cov.model="exponential"
my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~gestion+X1+X2+X3,trend.l=~gestion+X1+X2+X3,
cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,5,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,
burn.in=30000,n.iter=70000)
my.output=list(sim.predict=T)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model,output=my.output,locations=cbind(my.data$x,my.data$y)))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[5]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[5]][b,]=c(MCMCLH$posterior$beta$mean,
sqrt(diag(MCMCLH$posterior$beta$var)), 
sqrt(MCMCLH$posterior$sigmasq$mean),MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[5]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],
c(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

my.S.scale=0.001
my.phi.scale=0.1
my.geodata=vector("list")
my.geodata$coords=cbind(x,y)
my.geodata$data=my.data$Y6
my.geodata$cov.model="exponential"

```

```

my.geodata$units.m=rep(1,n)
my.model=list(trend.d=~gestion+X1+X2+X3,trend.l=~gestion+X1+X2+X3,
cov.model="exponential")
my.prior=prior.glm.control(beta.prior="normal",beta=rep(0,P),beta.var.std=diag(1,P),
sigmasq.prior="uniform",phi.prior="uniform",phi.discrete=seq(0,5,l=500))
my.mcmcinput=mcmc.control(S.scale=my.S.scale,thin=10,phi.scale=my.phi.scale,
burn.in=30000,n.iter=70000)
my.output=list(sim.predict=T)
MCMCLH=try(pois.krige.bayes(my.geodata,prior=my.prior,mcmc.input=my.mcmcinput,
model=my.model,output=my.output,locations=cbind(my.data$x,my.data$y)))
if (is(MCMCLH,"try-error")) { MCMCLH.coef[[6]][b,]=rep(NA,2*P+HP) }
else { MCMCLH.coef[[6]][b,]=c(MCMCLH$posterior$beta$mean,
sqrt(diag(MCMCLH$posterior$beta$var)),
sqrt(MCMCLH$posterior$sigmasq$mean),MCMCLH$posterior$phi$mean)
for (j in 1:P) { MCMCLH.quant[[6]][[b]][,j]=quantile(
MCMCLH$posterior$beta$sample[j,],
c(0.01/2,0.05/2,0.1/2,1-0.01/2,1-0.05/2,1-0.1/2)) } }

}

```

Appendix 5: Inter-plot distance matrices for the forest sites in the irregularly-spaced scenarios

[1,]	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.000000	4.149015	2.789313	3.796691	4.407446	7.863933	8.061611
[2,]	4.149015	0.000000	5.391776	6.932239	6.769669	6.384297	8.225342
[3,]	2.789313	5.391776	0.000000	1.568175	1.640435	6.260281	5.593054
[4,]	3.796691	6.932239	1.568175	0.000000	1.346923	7.298747	5.946443
[5,]	4.407446	6.769669	1.640435	1.346923	0.000000	6.096013	4.600101
[6,]	7.863933	6.384297	6.260281	7.298747	6.096013	0.000000	3.231910
[7,]	8.061611	8.225342	5.593054	5.946443	4.600101	3.231910	0.000000
[8,]	9.116741	10.858200	6.334033	5.629344	4.710926	7.181880	3.997437
[9,]	12.814486	13.042431	10.166711	10.073093	8.828590	7.155518	4.888482
[10,]	9.341541	11.725335	6.687339	5.582453	5.075777	8.841959	5.760229
[11,]	6.982316	7.862161	4.333666	4.474776	3.140997	4.157696	1.544936
[12,]	5.554881	7.914158	2.823266	2.042988	1.195015	6.463863	4.365564
[13,]	9.157034	9.120303	6.699043	7.009544	5.667120	3.530401	1.106763
[14,]	8.937573	8.716760	6.565912	6.989514	5.642591	3.021698	1.055774
[15,]	5.143978	6.312366	2.561309	3.083187	1.787638	4.336005	3.034133
[16,]	5.200251	6.755804	2.473249	2.668208	1.323741	4.922893	3.289511
[17,]	4.009907	5.461671	1.539267	2.511128	1.549871	4.825043	4.104004
[18,]	5.446248	6.886745	2.733354	2.920096	1.573297	4.771387	3.030017
[19,]	4.180712	5.407440	1.797638	2.782489	1.769706	4.547228	3.897230
[20,]	4.228769	5.824104	1.610345	2.339602	1.249740	4.959242	3.982799
[21,]	4.379673	5.756684	1.842156	2.624134	1.502819	4.674614	3.761371
[22,]	4.427568	5.611339	1.989590	2.859472	1.753454	4.441741	3.660550
[23,]	3.914409	5.288381	1.536930	2.608959	1.703430	4.767431	4.170308
[24,]	4.833003	6.381227	2.140289	2.527232	1.228160	4.880395	3.514513
	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	9.116741	12.814486	9.341541	6.982316	5.554881	9.1570343	8.9375729
[2,]	10.858200	13.042431	11.725335	7.862161	7.914158	9.1203029	8.7167602
[3,]	6.334033	10.166711	6.687339	4.333666	2.823266	6.6990431	6.5659117
[4,]	5.629344	10.073093	5.582453	4.474776	2.042988	7.0095436	6.9895136
[5,]	4.710926	8.828590	5.075777	3.140997	1.195015	5.6671201	5.6425910
[6,]	7.181880	7.155518	8.841959	4.157696	6.463863	3.5304008	3.0216984
[7,]	3.997437	4.888482	5.760229	1.544936	4.365564	1.1067633	1.0557736
[8,]	0.000000	5.208866	1.905629	3.236645	3.615275	4.2730516	4.6714307
[9,]	5.208866	0.000000	6.850565	5.841512	8.096276	3.9225371	4.3374981
[10,]	1.905629	6.850565	0.000000	4.718758	3.881384	6.1491233	6.5227362
[11,]	3.236645	5.841512	4.718758	0.000000	2.820806	2.5464408	2.5976736
[12,]	3.615275	8.096276	3.881384	2.820806	0.000000	5.3351252	5.4181787
[13,]	4.273052	3.922537	6.149123	2.546441	5.335125	0.0000000	0.5113052
[14,]	4.671431	4.337498	6.522736	2.597674	5.418179	0.5113052	0.0000000
[15,]	4.546401	7.671726	5.548620	1.841526	2.221276	4.1390337	4.0253180
[16,]	4.156093	7.697870	5.011530	1.888600	1.589866	4.3701991	4.3252941
[17,]	5.498498	8.804595	6.264604	2.977391	2.522599	5.2100528	5.0494353
[18,]	3.989247	7.434767	4.933123	1.620904	1.697634	4.1068315	4.0696911
[19,]	5.491189	8.641888	6.338219	2.839911	2.668555	5.0007409	4.8235420
[20,]	5.161746	8.602568	5.901511	2.761363	2.165095	5.0887032	4.9618163
[21,]	5.157339	8.435767	5.985590	2.604540	2.331613	4.8681325	4.7242533
[22,]	5.267593	8.392045	6.158079	2.587524	2.557116	4.7655902	4.5987708
[23,]	5.660303	8.903430	6.438414	3.089286	2.694811	5.2743079	5.0982355
[24,]	4.548134	8.028539	5.364644	2.193508	1.791385	4.6110278	4.5291465
	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]
[1,]	5.1439779	5.2002512	4.0099069	5.4462482	4.1807120	4.2287687	4.3796732
[2,]	6.3123660	6.7558038	5.4616705	6.8867445	5.4074398	5.8241038	5.7566839
[3,]	2.5613094	2.4732493	1.5392674	2.7333542	1.7976385	1.6103447	1.8421563
[4,]	3.0831875	2.6682078	2.5111282	2.9200962	2.7824890	2.3396019	2.6241343
[5,]	1.7876381	1.3237409	1.5498710	1.5732975	1.7697062	1.2497400	1.5028187
[6,]	4.3360048	4.9228928	4.8250430	4.7713873	4.5472277	4.9592421	4.6746144
[7,]	3.0341333	3.2895114	4.1040035	3.0300165	3.8972304	3.9827986	3.7613706
[8,]	4.5464010	4.1560926	5.4984980	3.9892472	5.4911893	5.1617456	5.1573392

[9,]	7.6717263	7.6978699	8.8045954	7.4347672	8.6418876	8.6025676	8.4357670
[10,]	5.5486197	5.0115296	6.2646041	4.9331225	6.3382194	5.9015111	5.9855898
[11,]	1.8415257	1.8885997	2.9773910	1.6209041	2.8399114	2.7613629	2.6045401
[12,]	2.2212755	1.5898657	2.5225989	1.6976342	2.6685549	2.1650951	2.3316134
[13,]	4.1390337	4.3701991	5.2100528	4.1068315	5.0007409	5.0887032	4.8681325
[14,]	4.0253180	4.3252941	5.0494353	4.0696911	4.8235420	4.9618163	4.7242533
[15,]	0.0000000	0.6325820	1.1359727	0.6186316	1.0135453	0.9521029	0.7643049
[16,]	0.6325820	0.0000000	1.3431612	0.2677779	1.3494747	1.0190289	1.0053944
[17,]	1.1359727	1.3431612	0.0000000	1.5359300	0.2778489	0.3643432	0.3747693
[18,]	0.6186316	0.2677779	1.5359300	0.0000000	1.5027179	1.2371035	1.1769286
[19,]	1.0135453	1.3494747	0.2778489	1.5027179	0.0000000	0.5244292	0.3527776
[20,]	0.9521029	1.0190289	0.3643432	1.2371035	0.5244292	0.0000000	0.2850018
[21,]	0.7643049	1.0053944	0.3747693	1.1769286	0.3527776	0.2850018	0.0000000
[22,]	0.7642578	1.1497869	0.4524820	1.2794237	0.2523886	0.5238416	0.2506472
[23,]	1.2509596	1.5042513	0.1745222	1.6872916	0.2746962	0.5388581	0.5116815
[24,]	0.5602865	0.3938642	0.9503710	0.6147992	0.9753794	0.6272264	0.6245518
	[,22]	[,23]	[,24]				
[1,]	4.4275678	3.9144089	4.8330026				
[2,]	5.6113390	5.2883806	6.3812265				
[3,]	1.9895899	1.5369301	2.1402890				
[4,]	2.8594722	2.6089586	2.5272317				
[5,]	1.7534537	1.7034301	1.2281600				
[6,]	4.4417406	4.7674314	4.8803948				
[7,]	3.6605498	4.1703080	3.5145131				
[8,]	5.2675934	5.6603026	4.5481338				
[9,]	8.3920449	8.9034300	8.0285387				
[10,]	6.1580790	6.4384142	5.3646435				
[11,]	2.5875241	3.0892855	2.1935077				
[12,]	2.5571158	2.6948108	1.7913849				
[13,]	4.7655902	5.2743079	4.6110278				
[14,]	4.5987708	5.0982355	4.5291465				
[15,]	0.7642578	1.2509596	0.5602865				
[16,]	1.1497869	1.5042513	0.3938642				
[17,]	0.4524820	0.1745222	0.9503710				
[18,]	1.2794237	1.6872916	0.6147992				
[19,]	0.2523886	0.2746962	0.9753794				
[20,]	0.5238416	0.5388581	0.6272264				
[21,]	0.2506472	0.5116815	0.6245518				
[22,]	0.0000000	0.5134920	0.8002481				
[23,]	0.5134920	0.0000000	1.1135695				
[24,]	0.8002481	1.1135695	0.0000000				

[[2]]							
	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.0000000	3.0020426	6.6434624	6.4375015	3.3021940	3.1450291	0.9321464
[2,]	3.0020426	0.0000000	7.4602660	7.2564241	3.3848730	2.6542351	2.9956537
[3,]	6.6434624	7.4602660	0.0000000	0.2095543	4.0759872	4.8188303	7.5715315
[4,]	6.4375015	7.2564241	0.2095543	0.0000000	3.8717176	4.6174515	7.3650923
[5,]	3.3021940	3.3848730	4.0759872	3.8717176	0.0000000	0.7817013	4.1038687
[6,]	3.1450291	2.6542351	4.8188303	4.6174515	0.7817013	0.0000000	3.8295907
[7,]	0.9321464	2.9956537	7.5715315	7.3650923	4.1038687	3.8295907	0.0000000
[8,]	0.7388322	2.2652861	6.6828045	6.4738067	3.0142495	2.6999002	1.1315485
[9,]	2.1692674	4.0333268	4.5030321	4.3001424	2.0249449	2.4210107	3.1010321
[10,]	1.1613544	1.8605343	6.9766546	6.7671390	3.1653412	2.7429444	1.2494339
[11,]	1.3976924	1.6946941	7.2187886	7.0092346	3.3562755	2.8856445	1.3175299
[12,]	1.5917110	1.5003959	7.2713210	7.0618532	3.3596704	2.8499375	1.5011612
[13,]	1.9644704	1.0751688	6.8268960	6.6186230	2.8117228	2.2200324	2.1347330
[14,]	2.6111959	0.7337493	7.7919259	7.5846272	3.7394899	3.0767099	2.4204721
[15,]	3.8599080	1.0392776	7.3090717	7.1141102	3.3360991	2.5544459	3.9796800
[16,]	4.1385236	1.3336574	7.3375552	7.1458981	3.4246531	2.6457366	4.2743590
	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	0.7388322	2.169267	1.1613544	1.3976924	1.5917110	1.9644704	2.6111959
[2,]	2.2652861	4.033327	1.8605343	1.6946941	1.5003959	1.0751688	0.7337493
[3,]	6.6828045	4.503032	6.9766546	7.2187886	7.2713210	6.8268960	7.7919259
[4,]	6.4738067	4.300142	6.7671390	7.0092346	7.0618532	6.6186230	7.5846272

```

[5,] 3.0142495 2.024945 3.1653412 3.3562755 3.3596704 2.8117228 3.7394899
[6,] 2.6999002 2.421011 2.7429444 2.8856445 2.8499375 2.2200324 3.0767099
[7,] 1.1315485 3.101032 1.2494339 1.3175299 1.5011612 2.1347330 2.4204721
[8,] 0.0000000 2.381860 0.4651505 0.7397060 0.9123404 1.2271777 1.9141486
[9,] 2.3818602 0.000000 2.7831214 3.0618565 3.1826663 3.0420002 4.0169846
[10,] 0.4651505 2.783121 0.0000000 0.2828427 0.4472136 0.8944272 1.4560220
[11,] 0.7397060 3.061856 0.2828427 0.0000000 0.2000000 0.8485281 1.2165525
[12,] 0.9123404 3.182666 0.4472136 0.2000000 0.0000000 0.7211103 1.0198039
[13,] 1.2271777 3.042000 0.8944272 0.8485281 0.7211103 0.0000000 1.0000000
[14,] 1.9141486 4.016985 1.4560220 1.2165525 1.0198039 1.0000000 0.0000000
[15,] 3.1263405 4.473767 2.7805082 2.6626352 2.4799246 1.9017955 1.7635266
[16,] 3.4076504 4.670125 3.0700831 2.9569934 2.7753576 2.1866435 2.0533412
[,15] [,16]
[1,] 3.8599080 4.1385236
[2,] 1.0392776 1.3336574
[3,] 7.3090717 7.3375552
[4,] 7.1141102 7.1458981
[5,] 3.3360991 3.4246531
[6,] 2.5544459 2.6457366
[7,] 3.9796800 4.2743590
[8,] 3.1263405 3.4076504
[9,] 4.4737668 4.6701247
[10,] 2.7805082 3.0700831
[11,] 2.6626352 2.9569934
[12,] 2.4799246 2.7753576
[13,] 1.9017955 2.1866435
[14,] 1.7635266 2.0533412
[15,] 0.0000000 0.2970387
[16,] 0.2970387 0.0000000

```

[[3]]

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.0000000	0.2016358	1.203756	8.7515179	8.8936972	5.0534160	5.0969452
[2,]	0.2016358	0.0000000	1.051439	8.6294763	8.7713689	4.9616709	5.0092216
[3,]	1.2037558	1.0514395	0.000000	8.8320898	8.9710216	5.3865370	5.4536734
[4,]	8.7515179	8.6294763	8.832090	0.0000000	0.1429021	3.8648510	3.8708283
[5,]	8.8936972	8.7713689	8.971022	0.1429021	0.0000000	4.0067555	4.0121492
[6,]	5.0534160	4.9616709	5.386537	3.8648510	4.0067555	0.0000000	0.1209339
[7,]	5.0969452	5.0092216	5.453673	3.8708283	4.0121492	0.1209339	0.0000000
[8,]	5.0357605	4.9551514	5.435894	4.0149433	4.1554125	0.3057924	0.2034158
[9,]	4.7281321	4.6467163	5.129953	4.2727304	4.4141414	0.4219502	0.4023444
[10,]	4.6898977	4.6129874	5.118483	4.3598232	4.5008280	0.5288677	0.4924409
	[,8]	[,9]	[,10]				
[1,]	5.0357605	4.7281321	4.6898977				
[2,]	4.9551514	4.6467163	4.6129874				
[3,]	5.4358940	5.1299533	5.1184826				
[4,]	4.0149433	4.2727304	4.3598232				
[5,]	4.1554125	4.4141414	4.5008280				
[6,]	0.3057924	0.4219502	0.5288677				
[7,]	0.2034158	0.4023444	0.4924409				
[8,]	0.0000000	0.3088122	0.3572338				
[9,]	0.3088122	0.0000000	0.1190168				
[10,]	0.3572338	0.1190168	0.0000000				

[[4]]

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.0000000	1.437783	0.7920158	1.697085	1.2060336	3.423345	3.8083310
[2,]	1.4377830	0.000000	1.2010150	0.409011	1.2721985	3.394846	2.8150114
[3,]	0.7920158	1.201015	0.000000	1.267844	0.4140193	2.693168	3.0691566
[4,]	1.6970851	0.409011	1.2678442	0.000000	1.2020004	3.115533	2.4060251
[5,]	1.2060336	1.272198	0.4140193	1.202000	0.0000000	2.326929	2.6953697
[6,]	3.4233447	3.394846	2.6931684	3.115533	2.3269293	0.000000	2.4329460
[7,]	3.8083310	2.815011	3.0691566	2.406025	2.6953697	2.432946	0.0000000
[8,]	3.6918327	2.843637	2.9234240	2.439428	2.5279913	2.008638	0.4310104

[9,]	4.1656954	3.195013	3.4157378	2.786006	3.0327850	2.508017	0.3803801
[10,]	4.8275934	3.686192	4.1216537	3.288784	3.7670260	3.402509	1.1122846
[11,]	4.6571104	3.598637	3.9247344	3.192918	3.5527297	3.027539	0.8574066
[12,]	5.1887729	4.072158	4.4699021	3.672524	4.1055325	3.557145	1.4169817
[13,]	4.9746273	3.845809	5.0297922	3.945674	5.1088568	6.917167	5.1674032
[14,]	5.0806948	3.810643	5.0115763	3.828040	5.0263773	6.627820	4.6893741
[15,]	5.3918830	4.420791	5.5609061	4.581792	5.6925851	7.635320	6.0045544
[16,]	6.9952876	5.973073	7.1356167	6.098884	7.2437679	9.081236	7.2508042
[17,]	4.8508280	3.497749	4.2744517	3.163802	4.0036869	4.321378	1.8939982
[18,]	4.8161035	3.408703	4.3121878	3.120202	4.0864987	4.683388	2.3014713
[19,]	5.1078559	3.721757	4.5665769	3.411612	4.3157096	4.719294	2.2946863
[20,]	4.1722365	3.163865	4.3082665	3.331395	4.4356560	6.405851	4.9094205
[21,]	4.2977400	3.010346	4.2110944	3.028845	4.2259815	5.877278	4.0535942
[22,]	5.2410686	4.812586	5.6964348	5.121103	5.9636060	8.206510	7.1301518
[23,]	5.4809456	4.044455	5.1149104	3.853254	4.9645946	5.871507	3.5409730
[24,]	5.2753028	4.608988	5.6185981	4.864885	5.8340800	7.980399	6.6699944
	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	3.6918327	4.1656954	4.8275934	4.6571104	5.1887729	4.9746273	5.0806948
[2,]	2.8436373	3.1950127	3.6861918	3.5986365	4.0721582	3.8458088	3.8106430
[3,]	2.9234240	3.4157378	4.1216537	3.9247344	4.4699021	5.0297922	5.0115763
[4,]	2.4394280	2.7860065	3.2887840	3.1929179	3.6725242	3.9456742	3.8280400
[5,]	2.5279913	3.0327850	3.7670260	3.5527297	4.1055325	5.1088568	5.0263773
[6,]	2.0086383	2.5080169	3.4025086	3.0275389	3.5571451	6.9171672	6.6278202
[7,]	0.4310104	0.3803801	1.1122846	0.8574066	1.4169817	5.1674032	4.6893741
[8,]	0.0000000	0.5639902	1.4518158	1.1251542	1.6965297	5.4878376	5.0434961
[9,]	0.5639902	0.0000000	0.9015831	0.5611640	1.1327886	5.4461698	4.9346110
[10,]	1.4518158	0.9015831	0.0000000	0.3946707	0.3950114	5.2470454	4.6402597
[11,]	1.1251542	0.5611640	0.3946707	0.0000000	0.5723679	5.4623850	4.8887304
[12,]	1.6965297	1.1327886	0.3950114	0.5723679	0.0000000	5.5782278	4.9491446
[13,]	5.4878376	5.4461698	5.2470454	5.4623850	5.5782278	0.0000000	0.7896487
[14,]	5.0434961	4.9346110	4.6402597	4.8887304	4.9491446	0.7896487	0.0000000
[15,]	6.3071725	6.2966293	6.1308727	6.3360345	6.4669283	0.8965562	1.6415462
[16,]	7.5950450	7.5018095	7.1892772	7.4502615	7.4824181	2.1657694	2.5678325
[17,]	2.3244176	1.9242396	1.2841815	1.6254442	1.5277277	4.1119398	3.4469070
[18,]	2.7292380	2.3958800	1.8405600	2.1613070	2.0980727	3.5676628	2.8860833
[19,]	2.7253853	2.3246165	1.6467963	2.0052703	1.8462557	3.9332568	3.2306092
[20,]	5.1781850	5.2274350	5.1786517	5.3353767	5.5410148	0.8995521	1.4464788
[21,]	4.3804004	4.3300518	4.1480146	4.3524053	4.4885901	1.1161729	0.8012496
[22,]	7.3355810	7.4771798	7.5390224	7.6566962	7.9139272	2.8422591	3.6263508
[23,]	3.9629449	3.6554837	3.0773048	3.4150584	3.2931828	2.8118097	2.0344427
[24,]	6.9162245	6.9982262	6.9698483	7.1232257	7.3316983	2.0140817	2.8037133
	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]
[1,]	5.3918830	6.995288	4.8508280	4.8161035	5.1078559	4.1722365	4.2977400
[2,]	4.4207911	5.973073	3.4977490	3.4087031	3.7217571	3.1638649	3.0103457
[3,]	5.5609061	7.135617	4.2744517	4.3121878	4.5665769	4.3082665	4.2110944
[4,]	4.5817922	6.098884	3.1638018	3.1202021	3.4116122	3.3313951	3.0288455
[5,]	5.6925851	7.243768	4.0036869	4.0864987	4.3157096	4.4356560	4.2259815
[6,]	7.6353196	9.081236	4.3213777	4.6833877	4.7192941	6.4058514	5.8772783
[7,]	6.0045544	7.250804	1.8939982	2.3014713	2.2946863	4.9094205	4.0535942
[8,]	6.3071725	7.595045	2.3244176	2.7292380	2.7253853	5.1781850	4.3804004
[9,]	6.2966293	7.501810	1.9242396	2.3958800	2.3246165	5.2274350	4.3300518
[10,]	6.1308727	7.189277	1.2841815	1.8405600	1.6467963	5.1786517	4.1480146
[11,]	6.3360345	7.450262	1.6254442	2.1613070	2.0052703	5.3353767	4.3524053
[12,]	6.4669283	7.482418	1.5277277	2.0980727	1.8462557	5.5410148	4.4885901
[13,]	0.8965562	2.165769	4.1119398	3.5676628	3.9332568	0.8995521	1.1161729
[14,]	1.6415462	2.567833	3.4469070	2.8860833	3.2306092	1.4464788	0.8012496
[15,]	0.0000000	1.608657	5.0074578	4.4641950	4.8293265	1.2569364	1.9843450
[16,]	1.6086572	0.000000	5.9613438	5.3915190	5.7001403	2.8325616	3.2270806
[17,]	5.0074578	5.961344	0.0000000	0.5708283	0.4021791	4.1804609	3.0570288
[18,]	4.4641950	5.391519	0.5708283	0.0000000	0.4021007	3.6900759	2.5376919
[19,]	4.8293265	5.700140	0.4021791	0.4021007	0.0000000	4.0844981	2.9232819
[20,]	1.2569364	2.832562	4.1804609	3.6900759	4.0844981	0.0000000	1.1948104
[21,]	1.9843450	3.227081	3.0570288	2.5376919	2.9232819	1.1948104	0.0000000
[22,]	2.1600190	3.119533	6.6031002	6.1230559	6.5186956	2.4349860	3.6184650
[23,]	3.6753292	4.334692	1.7937037	1.2621822	1.4469765	3.2200056	2.0390292
[24,]	1.2710287	2.328193	5.9531081	5.4447017	5.8315271	1.7911988	2.9082457

	[,22]	[,23]	[,24]
[1,]	5.2410686	5.480946	5.2753028
[2,]	4.8125856	4.044455	4.6089877
[3,]	5.6964348	5.114910	5.6185981
[4,]	5.1211032	3.853254	4.8648850
[5,]	5.9636060	4.964595	5.8340800
[6,]	8.2065102	5.871507	7.9803993
[7,]	7.1301518	3.540973	6.6699944
[8,]	7.3355810	3.962945	6.9162245
[9,]	7.4771798	3.655484	6.9982262
[10,]	7.5390224	3.077305	6.9698483
[11,]	7.6566962	3.415058	7.1232257
[12,]	7.9139272	3.293183	7.3316983
[13,]	2.8422591	2.811810	2.0140817
[14,]	3.6263508	2.034443	2.8037133
[15,]	2.1600190	3.675329	1.2710287
[16,]	3.1195333	4.334692	2.3281933
[17,]	6.6031002	1.793704	5.9531081
[18,]	6.1230559	1.262182	5.4447017
[19,]	6.5186956	1.446977	5.8315271
[20,]	2.4349860	3.220006	1.7911988
[21,]	3.6184650	2.039029	2.9082457
[22,]	0.0000000	5.597125	0.8949749
[23,]	5.5971247	0.000000	4.8140362
[24,]	0.8949749	4.814036	0.0000000

[[5]]

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.0000000	0.3751706	4.2094318	4.9979908	4.7140994	0.8854016	3.9582351
[2,]	0.3751706	0.0000000	4.3323381	5.1091366	5.0716581	1.1841423	4.3305838
[3,]	4.2094318	4.3323381	0.0000000	0.7981103	4.1425708	4.6823605	4.4483868
[4,]	4.9979908	5.1091366	0.7981103	0.0000000	4.6042486	5.4804708	5.0474212
[5,]	4.7140994	5.0716581	4.1425708	4.6042486	0.0000000	4.3741254	1.1181878
[6,]	0.8854016	1.1841423	4.6823605	5.4804708	4.3741254	0.0000000	3.4755064
[7,]	3.9582351	4.3305838	4.4483868	5.0474212	1.1181878	3.4755064	0.0000000
[8,]	3.8802733	4.2510234	4.2991917	4.8998225	1.0743277	3.4250385	0.1493051
[9,]	3.9282006	4.2974901	4.2370447	4.8278956	0.9742982	3.4918431	0.2272378
[10,]	4.5529957	4.9071863	3.9393352	4.4126901	0.2124265	4.2411652	1.1090000
[11,]	4.6982935	5.0501462	3.9424016	4.3893337	0.2291397	4.3999557	1.2690394
[12,]	4.4006872	4.7533173	3.8115283	4.3021753	0.3701094	4.1031071	1.0711867
	[,8]	[,9]	[,10]	[,11]	[,12]		
[1,]	3.8802733	3.9282006	4.5529957	4.6982935	4.4006872		
[2,]	4.2510234	4.2974901	4.9071863	5.0501462	4.7533173		
[3,]	4.2991917	4.2370447	3.9393352	3.9424016	3.8115283		
[4,]	4.8998225	4.8278956	4.4126901	4.3893337	4.3021753		
[5,]	1.0743277	0.9742982	0.2124265	0.2291397	0.3701094		
[6,]	3.4250385	3.4918431	4.2411652	4.3999557	4.1031071		
[7,]	0.1493051	0.2272378	1.1090000	1.2690394	1.0711867		
[8,]	0.0000000	0.1074663	1.0383087	1.2035618	0.9817051		
[9,]	0.1074663	0.0000000	0.9323143	1.0981922	0.8742402		
[10,]	1.0383087	0.9323143	0.0000000	0.1690680	0.1635910		
[11,]	1.2035618	1.0981922	0.1690680	0.0000000	0.2986001		
[12,]	0.9817051	0.8742402	0.1635910	0.2986001	0.0000000		

[[6]]

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.000000	5.861517	7.895360	1.759923	1.0070402	1.3997857	2.5766273
[2,]	5.861517	0.000000	2.348569	7.359936	4.8751030	4.7198637	3.6690005
[3,]	7.895360	2.348569	0.000000	9.508637	6.9609158	6.6104963	5.9427745
[4,]	1.759923	7.359936	9.508637	0.000000	2.5598141	3.1570686	3.7746165
[5,]	1.007040	4.875103	6.960916	2.559814	0.0000000	0.9329577	1.6159165
[6,]	1.399786	4.719864	6.610496	3.157069	0.9329577	0.0000000	2.1231769
[7,]	2.576627	3.669001	5.942775	3.774617	1.6159165	2.1231769	0.0000000
[8,]	2.282605	3.810507	6.040888	3.569686	1.3027759	1.7933391	0.3326094

```

[,8]
[1,] 2.2826049
[2,] 3.8105068
[3,] 6.0408880
[4,] 3.5696864
[5,] 1.3027759
[6,] 1.7933391
[7,] 0.3326094
[8,] 0.0000000

[[7]]
 [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.0000000 0.6678091 2.1121603 2.3590477 5.154732 1.5504141 0.7319870
[2,] 0.6678091 0.0000000 1.4480345 1.6972451 4.501158 1.1186657 0.1118928
[3,] 2.1121603 1.4480345 0.0000000 0.2540256 3.062649 1.4799949 1.4064665
[4,] 2.3590477 1.6972451 0.2540256 0.0000000 2.809342 1.6991086 1.6584297
[5,] 5.1547315 4.5011576 3.0626485 2.8093424 0.0000000 4.3378618 4.4669887
[6,] 1.5504141 1.1186657 1.4799949 1.6991086 4.337862 0.0000000 1.0069364
[7,] 0.7319870 0.1118928 1.4064665 1.6584297 4.466989 1.0069364 0.0000000
[8,] 0.7075486 1.3345239 2.7771687 3.0285246 5.835217 1.9350941 1.3709449
[9,] 0.3678872 0.6001666 2.0050865 2.2583881 5.067696 1.2227841 0.6065905
[10,] 1.4646327 0.9597385 1.2743979 1.5024463 4.192156 0.2294014 0.8485906
 [,8]      [,9]      [,10]
[1,] 0.7075486 0.3678872 1.4646327
[2,] 1.3345239 0.6001666 0.9597385
[3,] 2.7771687 2.0050865 1.2743979
[4,] 3.0285246 2.2583881 1.5024463
[5,] 5.8352174 5.0676957 4.1921558
[6,] 1.9350941 1.2227841 0.2294014
[7,] 1.3709449 0.6065905 0.8485906
[8,] 0.0000000 0.7902569 1.9238513
[9,] 0.7902569 0.0000000 1.1684802
[10,] 1.9238513 1.1684802 0.0000000

[[8]]
 [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.000000 10.0726490 1.320259 9.8133012 10.478076 10.0462451 9.7240796
[2,] 10.072649 0.0000000 10.097315 0.4831076 1.441112 0.4548428 0.5636391
[3,] 1.320259 10.0973149 0.0000000 9.8940590 10.675253 10.0116571 9.6928808
[4,] 9.813301 0.4831076 9.894059 0.0000000 1.159436 0.8862150 0.8422173
[5,] 10.478076 1.4411124 10.675253 1.1594361 0.0000000 1.8927359 1.9599944
[6,] 10.046245 0.4548428 10.011657 0.8862150 1.892736 0.0000000 0.3221878
[7,] 9.724080 0.5636391 9.692881 0.8422173 1.959994 0.3221878 0.0000000
[8,] 8.310321 2.1082146 8.186487 2.1314268 3.257161 1.8884928 1.5934296
[9,] 6.673295 4.6994881 7.234924 4.2587365 4.394402 4.9347926 4.6738816
[10,] 6.545967 3.8036291 6.842053 3.4408094 3.946728 3.9301567 3.6306254
[11,] 6.436984 3.9243406 6.745150 3.5614708 4.061084 4.0496021 3.7494597
[12,] 6.860299 3.2333303 6.867240 3.0346486 3.922270 3.1859550 2.8637839
[13,] 6.495826 4.4163788 6.144954 4.3928539 5.457914 4.1908023 3.9029816
[14,] 6.685526 4.7043843 6.229184 4.7292309 5.828929 4.4370749 4.1682439
[15,] 3.649958 7.4222910 3.012805 7.2961229 8.220779 7.2693660 6.9612112
[16,] 1.416651 8.6690948 2.011579 8.4026515 9.061484 8.6537061 8.3315628
[17,] 4.934222 5.1475498 5.103472 4.8791089 5.582525 5.1499162 4.8286505
[18,] 5.049660 5.0560345 5.262764 4.7719315 5.442484 5.0765434 4.7565807
[19,] 5.309113 4.7695066 5.368129 4.5354162 5.317170 4.7380140 4.4158272
[20,] 5.303632 4.7731358 5.440167 4.5107793 5.236171 4.7718480 4.4504759
[21,] 7.480969 3.1903160 7.252549 3.1960765 4.296244 2.9558528 2.6698818
[22,] 7.642929 4.4263055 7.140506 4.5492743 5.697952 4.0883584 3.8652737
[23,] 7.561639 4.1893217 7.108768 4.2935991 5.436769 3.8661311 3.6314516
[24,] 7.421666 3.6648728 7.090707 3.7154235 4.837303 3.3870744 3.1227606
[25,] 7.947323 3.9223443 7.509697 4.0629216 5.216991 3.5758508 3.3598122
[26,] 7.746271 3.6902478 7.368988 3.7924127 4.936514 3.3729567 3.1336139
[27,] 7.827742 3.5982565 7.459293 3.7065327 4.852965 3.2774453 3.0405823
[28,] 6.935554 3.3772276 6.802000 3.2788890 4.288547 3.2308791 2.9187396

```

[29,]	6.852376	3.3870667	6.751168	3.2631335	4.244396	3.2642887	2.9480587
	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	8.310321	6.673295	6.5459670	6.4369842	6.8602989	6.4958257	6.6855260
[2,]	2.108215	4.699488	3.8036291	3.9243406	3.2333303	4.4163788	4.7043843
[3,]	8.186487	7.234924	6.8420533	6.7451501	6.8672405	6.1449544	6.2291843
[4,]	2.131427	4.258737	3.4408094	3.5614708	3.0346486	4.3928539	4.7292309
[5,]	3.257161	4.394402	3.9467284	4.0610838	3.9222698	5.4579136	5.8289290
[6,]	1.888493	4.934793	3.9301567	4.0496021	3.1859550	4.1908023	4.4370749
[7,]	1.593430	4.673882	3.6306254	3.7494597	2.8637839	3.9029816	4.1682439
[8,]	0.000000	4.197098	2.8363168	2.9356406	1.5775221	2.3109429	2.6046084
[9,]	4.197098	0.000000	1.4745203	1.4241717	2.9765935	4.6424746	5.1956155
[10,]	2.836317	1.474520	0.0000000	0.1208015	1.5041027	3.2101171	3.7619157
[11,]	2.935641	1.424172	0.1208015	0.0000000	1.5697650	3.2337200	3.7865310
[12,]	1.577522	2.976594	1.5041027	1.5697650	0.0000000	1.8814997	2.4064150
[13,]	2.310943	4.642475	3.2101171	3.2337200	1.8814997	0.0000000	0.5531889
[14,]	2.604608	5.195615	3.7619157	3.7865310	2.4064150	0.5531889	0.0000000
[15,]	5.390706	5.695897	4.7807374	4.7187597	4.2986585	3.2006815	3.2286251
[16,]	6.947710	5.322463	5.1315305	5.0233377	5.4694515	5.2547960	5.5071093
[17,]	3.546873	2.670173	1.7781240	1.6976979	1.9924708	2.5814138	3.0762258
[18,]	3.525925	2.450343	1.5923169	1.5058991	1.9546340	2.7085540	3.2173885
[19,]	3.065319	2.888328	1.7414273	1.6927711	1.5527691	2.0661474	2.5776782
[20,]	3.174746	2.588312	1.5193081	1.4560110	1.6150697	2.3600019	2.8802335
[21,]	1.082121	4.340563	2.8713596	2.9385039	1.3687399	1.2350490	1.5342555
[22,]	2.460876	5.796642	4.3257325	4.3726931	2.8490969	1.3580788	0.9730961
[23,]	2.188082	5.506790	4.0343666	4.0834918	2.5523622	1.1590729	0.8816439
[24,]	1.584154	4.832683	3.3581699	3.4139142	1.8622417	0.9458990	1.0522001
[25,]	2.011252	5.609812	4.1359116	4.1952976	2.6337046	1.4934661	1.2805284
[26,]	1.692270	5.205296	3.7316518	3.7919030	2.2289302	1.2507826	1.1893801
[27,]	1.615774	5.197933	3.7258411	3.7888855	2.2217842	1.3321652	1.2886000
[28,]	1.386032	3.692285	2.2178512	2.2733730	0.7315279	1.2331829	1.7211351
[29,]	1.457945	3.513220	2.0390206	2.0933385	0.5635069	1.3482229	1.8567340
	[,15]	[,16]	[,17]	[,18]	[,19]	[,20]	[,21]
[1,]	3.649958	1.416651	4.9342222	5.0496603	5.3091130	5.3036324	7.4809685
[2,]	7.422291	8.669095	5.1475498	5.0560345	4.7695066	4.7731358	3.1903160
[3,]	3.012805	2.011579	5.1034722	5.2627637	5.3681285	5.4401671	7.2525489
[4,]	7.296123	8.402651	4.8791089	4.7719315	4.5354162	4.5107793	3.1960765
[5,]	8.220779	9.061484	5.5825245	5.4424842	5.3171695	5.2361713	4.2962436
[6,]	7.269366	8.653706	5.1499162	5.0765434	4.7380140	4.7718480	2.9558528
[7,]	6.961211	8.331563	4.8286505	4.7565807	4.4158272	4.4504759	2.6698818
[8,]	5.390706	6.947710	3.5468730	3.5259248	3.0653189	3.1747457	1.0821206
[9,]	5.695897	5.322463	2.6701732	2.4503430	2.8883277	2.5883124	4.3405631
[10,]	4.780737	5.131530	1.7781240	1.5923169	1.7414273	1.5193081	2.8713596
[11,]	4.718760	5.023338	1.6976979	1.5058991	1.6927711	1.4560110	2.9385039
[12,]	4.298659	5.469452	1.9924708	1.9546340	1.5527691	1.6150697	1.3687399
[13,]	3.200681	5.254796	2.5814138	2.7085540	2.0661474	2.3600019	1.2350490
[14,]	3.228625	5.507109	3.0762258	3.2173885	2.5776782	2.8802335	1.5342555
[15,]	0.000000	2.785829	3.0610660	3.2765337	3.0538188	3.2628817	4.3792620
[16,]	2.785829	0.000000	3.5243609	3.6350232	3.9167017	3.8964458	6.1640757
[17,]	3.061066	3.524361	0.0000000	0.2198590	0.5404665	0.3787374	2.9888093
[18,]	3.276534	3.635023	0.2198590	0.0000000	0.6424407	0.3768979	3.0314869
[19,]	3.053819	3.916702	0.5404665	0.6424407	0.0000000	0.3170647	2.4557087
[20,]	3.262882	3.896446	0.3787374	0.3768979	0.3170647	0.0000000	2.6552674
[21,]	4.379262	6.164076	2.9888093	3.0314869	2.4557087	2.6552674	0.0000000
[22,]	4.128190	6.478578	3.9316812	4.0484942	3.4073299	3.6896381	1.5942145
[23,]	4.104096	6.365291	3.7012956	3.8088724	3.1706455	3.4448943	1.2938949
[24,]	4.140856	6.155245	3.2111711	3.2903231	2.6709496	2.9161115	0.5914499
[25,]	4.507162	6.735822	3.9630014	4.0546164	3.4253952	3.6835214	1.2878078
[26,]	4.389729	6.501339	3.6136477	3.6949138	3.0737264	3.3209926	0.8852463
[27,]	4.483396	6.576820	3.6557950	3.7318570	3.1154109	3.3568028	0.8628615
[28,]	4.046326	5.585858	2.2976486	2.3265186	1.7748558	1.9522646	0.7185499
[29,]	4.038674	5.492118	2.1570721	2.1754747	1.6436666	1.8038897	0.8913344
	[,22]	[,23]	[,24]	[,25]	[,26]	[,27]	[,28]
[1,]	7.6429294	7.5616386	7.4216656	7.9473229	7.7462707	7.8277425	6.9355539
[2,]	4.4263055	4.1893217	3.6648728	3.9223443	3.6902478	3.5982565	3.3772276
[3,]	7.1405059	7.1087682	7.0907066	7.5096972	7.3689876	7.4592931	6.8020001
[4,]	4.5492743	4.2935991	3.7154235	4.0629216	3.7924127	3.7065327	3.2788890

```

[5,] 5.6979516 5.4367686 4.8373034 5.2169915 4.9365140 4.8529646 4.2885467
[6,] 4.0883584 3.8661311 3.3870744 3.5758508 3.3729567 3.2774453 3.2308791
[7,] 3.8652737 3.6314516 3.1227606 3.3598122 3.1336139 3.0405823 2.9187396
[8,] 2.4608765 2.1880825 1.5841540 2.0112523 1.6922701 1.6157738 1.3860321
[9,] 5.7966425 5.5067904 4.8326830 5.6098119 5.2052959 5.1979329 3.6922852
[10,] 4.3257325 4.0343666 3.3581699 4.1359116 3.7316518 3.7258411 2.2178512
[11,] 4.3726931 4.0834918 3.4139142 4.1952976 3.7919030 3.7888855 2.2733730
[12,] 2.8490969 2.5523622 1.8622417 2.6337046 2.2289302 2.2217842 0.7315279
[13,] 1.3580788 1.1590729 0.9458990 1.4934661 1.2507826 1.3321652 1.2331829
[14,] 0.9730961 0.8816439 1.0522001 1.2805284 1.1893801 1.2886000 1.7211351
[15,] 4.1281899 4.1040960 4.1408556 4.5071615 4.3897295 4.4833963 4.0463260
[16,] 6.4785780 6.3652913 6.1552452 6.7358221 6.5013391 6.5768203 5.5858583
[17,] 3.9316812 3.7012956 3.2111711 3.9630014 3.6136477 3.6557950 2.2976486
[18,] 4.0484942 3.8088724 3.2903231 4.0546164 3.6949138 3.7318570 2.3265186
[19,] 3.4073299 3.1706455 2.6709496 3.4253952 3.0737264 3.1154109 1.7748558
[20,] 3.6896381 3.4448943 2.9161115 3.6835214 3.3209926 3.3568028 1.9522646
[21,] 1.5942145 1.2938949 0.5914499 1.2878078 0.8852463 0.8628615 0.7185499
[22,] 0.0000000 0.3016820 1.0114094 0.5211842 0.7693874 0.8463534 2.1180940
[23,] 0.3016820 0.0000000 0.7097359 0.4053061 0.5013272 0.5910685 1.8222651
[24,] 1.0114094 0.7097359 0.0000000 0.7902050 0.4052777 0.4463194 1.1406599
[25,] 0.5211842 0.4053061 0.7902050 0.0000000 0.4051790 0.4312830 1.9235002
[26,] 0.7693874 0.5013272 0.4052777 0.4051790 0.0000000 0.1011781 1.5219491
[27,] 0.8463534 0.5910685 0.4463194 0.4312830 0.1011781 0.0000000 1.5261887
[28,] 2.1180940 1.8222651 1.1406599 1.9235002 1.5219491 1.5261887 0.0000000
[29,] 2.2922541 1.9978581 1.3206014 2.1044002 1.7030238 1.7070949 0.1810994
[ ,29]

[1,] 6.8523757
[2,] 3.3870667
[3,] 6.7511685
[4,] 3.2631335
[5,] 4.2443963
[6,] 3.2642887
[7,] 2.9480587
[8,] 1.4579451
[9,] 3.5132197
[10,] 2.0390206
[11,] 2.0933385
[12,] 0.5635069
[13,] 1.3482229
[14,] 1.8567340
[15,] 4.0386737
[16,] 5.4921180
[17,] 2.1570721
[18,] 2.1754747
[19,] 1.6436666
[20,] 1.8038897
[21,] 0.8913344
[22,] 2.2922541
[23,] 1.9978581
[24,] 1.3206014
[25,] 2.1044002
[26,] 1.7030238
[27,] 1.7070949
[28,] 0.1810994
[29,] 0.0000000

```

```

[[9]]
      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.000000 5.3459735 5.4902562 5.460305 4.777538 2.2426993 2.393818
[2,] 5.345974 0.0000000 0.4017785 2.230170 4.974548 3.1378383 3.742742
[3,] 5.490256 0.4017785 0.0000000 2.626740 4.749765 3.3248460 4.022541
[4,] 5.460305 2.2301697 2.6267404 0.000000 6.758526 3.3635413 3.185865
[5,] 4.777538 4.9745477 4.7497652 6.758526 0.000000 4.3838533 5.564917
[6,] 2.242699 3.1378383 3.3248460 3.363541 4.383853 0.0000000 1.202004
[7,] 2.393818 3.7427415 4.0225415 3.185865 5.564917 1.2020037 0.000000

```

[8,]	0.867947	4.4881285	4.6248768	4.744752	4.278594	1.4266184	1.890097
[9,]	1.335150	4.0782310	4.1929772	4.513915	3.901926	1.1504803	1.924030
[10,]	1.275251	4.0863731	4.2188109	4.425411	4.090535	1.0700477	1.751203
[11,]	1.516206	3.8297765	3.9786493	4.120977	4.186512	0.7692002	1.511400
[12,]	1.682403	3.6637381	3.8118658	3.990165	4.135967	0.6277993	1.493173
[13,]	1.859130	3.5058674	3.6787721	3.724511	4.338517	0.3854205	1.244098
[14,]	1.679603	3.7378674	3.9262059	3.816246	4.562543	0.6016062	1.086669
	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]	[,14]
[1,]	0.8679470	1.3351498	1.2752513	1.5162061	1.6824030	1.8591302	1.6796029
[2,]	4.4881285	4.0782310	4.0863731	3.8297765	3.6637381	3.5058674	3.7378674
[3,]	4.6248768	4.1929772	4.2188109	3.9786493	3.8118658	3.6787721	3.9262059
[4,]	4.7447523	4.5139146	4.4254107	4.1209772	3.9901652	3.7245108	3.8162458
[5,]	4.2785944	3.9019263	4.0905354	4.1865118	4.1359674	4.3385174	4.5625432
[6,]	1.4266184	1.1504803	1.0700477	0.7692002	0.6277993	0.3854205	0.6016062
[7,]	1.8900966	1.9240296	1.7512033	1.5114000	1.4931728	1.2440981	1.0866688
[8,]	0.0000000	0.4900378	0.4073156	0.6669820	0.8290259	1.0427277	0.9328933
[9,]	0.4900378	0.0000000	0.1925123	0.4291806	0.5273566	0.8081398	0.8444057
[10,]	0.4073156	0.1925123	0.0000000	0.3045275	0.4465356	0.7010827	0.6853204
[11,]	0.6669820	0.4291806	0.3045275	0.0000000	0.1667843	0.3965577	0.4248023
[12,]	0.8290259	0.5273566	0.4465356	0.1667843	0.0000000	0.2862953	0.4276658
[13,]	1.0427277	0.8081398	0.7010827	0.3965577	0.2862953	0.0000000	0.2881753
[14,]	0.9328933	0.8444057	0.6853204	0.4248023	0.4276658	0.2881753	0.0000000

[[10]]

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.0000000	0.6668043	1.344595	2.4173978	9.8939113	2.5949499	0.5043858
[2,]	0.6668043	0.0000000	2.002284	3.0681186	10.5607055	3.2562716	0.2099738
[3,]	1.3445951	2.0022837	0.000000	1.0760093	8.5792592	1.2551227	1.8488410
[4,]	2.4173978	3.0681186	1.076009	0.0000000	7.5696967	0.3163116	2.9217127
[5,]	9.8939113	10.5607055	8.579259	7.5696967	0.0000000	7.3323956	10.3862698
[6,]	2.5949499	3.2562716	1.255123	0.3163116	7.3323956	0.0000000	3.0977695
[7,]	0.5043858	0.2099738	1.848841	2.9217127	10.3862698	3.0977695	0.0000000
[8,]	8.9830666	9.6497959	7.675772	6.6792291	0.9325798	6.4341977	9.4731201
	[,8]						
[1,]	8.9830666						
[2,]	9.6497959						
[3,]	7.6757716						
[4,]	6.6792291						
[5,]	0.9325798						
[6,]	6.4341977						
[7,]	9.4731201						
[8,]	0.0000000						

[[11]]

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.0000000	0.2067704	0.4478236	0.9702464	1.8332335	3.191845	5.2127751
[2,]	0.2067704	0.0000000	0.2644920	0.7687939	1.6436828	2.996665	5.0155010
[3,]	0.4478236	0.2644920	0.0000000	0.5326537	1.3856453	2.744324	4.8661346
[4,]	0.9702464	0.7687939	0.5326537	0.0000000	0.8978246	2.232220	4.3861092
[5,]	1.8332335	1.6436828	1.3856453	0.8978246	0.0000000	1.366943	3.9681271
[6,]	3.1918449	2.9966648	2.7443243	2.2322195	1.3669429	0.000000	3.2173359
[7,]	5.2127751	5.0155010	4.8661346	4.3861092	3.9681271	3.217336	0.0000000
[8,]	5.7613793	5.5650303	5.4183453	4.9401743	4.5179293	3.721962	0.5546639
[9,]	3.4314901	3.2247868	3.0121531	2.4823283	1.8370991	1.020417	2.2410935
[10,]	4.0860060	3.8811489	3.6974322	3.1855199	2.6568952	1.868218	1.3609691
	[,8]	[,9]	[,10]				
[1,]	5.7613793	3.4314901	4.0860060				
[2,]	5.5650303	3.2247868	3.8811489				
[3,]	5.4183453	3.0121531	3.6974322				
[4,]	4.9401743	2.4823283	3.1855199				
[5,]	4.5179293	1.8370991	2.6568952				
[6,]	3.7219621	1.0204166	1.8682176				
[7,]	0.5546639	2.2410935	1.3609691				
[8,]	0.0000000	2.7683844	1.8899124				

```
[9,] 2.7683844 0.0000000 0.8803051
[10,] 1.8899124 0.8803051 0.0000000
```

```
[[12]]
[,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.0000000 3.3105891 3.5777088 3.9968849 4.1745391 3.6702962 3.4968949
[2,] 3.3105891 0.0000000 0.2828427 0.8411237 1.0276074 0.4371201 0.3850636
[3,] 3.5777088 0.2828427 0.0000000 0.7380305 0.9066295 0.1851324 0.2673462
[4,] 3.9968849 0.8411237 0.7380305 0.0000000 0.1893198 0.8493015 1.0022540
[5,] 4.1745391 1.0276074 0.9066295 0.1893198 0.0000000 1.0000125 1.1656865
[6,] 3.6702962 0.4371201 0.1851324 0.8493015 1.0000125 0.0000000 0.2000000
[7,] 3.4968949 0.3850636 0.2673462 1.0022540 1.1656865 0.2000000 0.0000000
[8,] 0.1626346 3.2245387 3.4876998 3.9296597 4.1095938 3.5742457 3.3972977
[,8]
[1,] 0.1626346
[2,] 3.2245387
[3,] 3.4876998
[4,] 3.9296597
[5,] 4.1095938
[6,] 3.5742457
[7,] 3.3972977
[8,] 0.0000000
```

```
[[13]]
[,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 0.0000000 0.6068047 10.6444354 10.4971839 5.013432 1.0845727
[2,] 0.6068047 0.0000000 10.7471839 10.6064045 5.596692 1.5332482
[3,] 10.6444354 10.7471839 0.0000000 0.1843909 9.653339 11.4255157
[4,] 10.4971839 10.6064045 0.1843909 0.0000000 9.471084 11.2709567
[5,] 5.0134320 5.5966919 9.6533385 9.4710845 0.000000 4.7373927
[6,] 1.0845727 1.5332482 11.4255157 11.2709567 4.737393 0.0000000
[7,] 1.9159783 2.5140535 10.9143003 10.7497565 3.443725 1.3090519
[8,] 2.1015142 2.6916229 11.1031034 10.9375018 3.424146 1.3961776
[9,] 9.8793465 10.1637877 3.3442902 3.1878377 7.398154 10.3975344
[10,] 9.9653304 10.4334383 7.2862888 7.1234989 5.738325 10.0773943
[11,] 3.4099761 3.0833795 8.5850312 8.4754728 6.962228 4.4944674
[12,] 1.5750292 2.1367707 11.2299052 11.0690528 4.008672 0.7681517
[13,] 0.9268058 1.4935786 10.9767220 10.8209952 4.391409 0.4678985
[14,] 9.7748297 9.9165125 1.0840724 0.8997939 8.577917 10.5108368
[15,] 10.9467254 11.0440493 0.3153982 0.4928002 9.962876 11.7331567
[16,] 0.8051397 1.3441287 11.0439651 10.8898039 4.573967 0.3838854
[,7]      [,8]      [,9]      [,10]     [,11]     [,12]     [,13]
[1,] 1.9159783 2.1015142 9.879347 9.965330 3.409976 1.5750292 0.9268058
[2,] 2.5140535 2.6916229 10.163788 10.433438 3.083379 2.1367707 1.4935786
[3,] 10.9143003 11.1031034 3.344290 7.286289 8.585031 11.2299052 10.9767220
[4,] 10.7497565 10.9375018 3.187838 7.123499 8.475473 11.0690528 10.8209952
[5,] 3.4437249 3.4241463 7.398154 5.738325 6.962228 4.0086722 4.3914090
[6,] 1.3090519 1.3961776 10.397534 10.077394 4.494467 0.7681517 0.4678985
[7,] 0.0000000 0.2350170 9.555884 8.907896 5.002960 0.5654184 1.0538738
[8,] 0.2350170 0.0000000 9.695484 8.957613 5.232296 0.6280326 1.2078812
[9,] 9.5558843 9.6954844 0.000000 3.953106 8.890545 9.9951255 9.9296719
[10,] 8.9078957 8.9576133 3.953106 0.000000 10.325580 9.4535031 9.6480498
[11,] 5.0029599 5.2322963 8.890545 10.325580 0.000000 4.8666056 4.2613287
[12,] 0.5654184 0.6280326 9.995125 9.453503 4.866606 0.0000000 0.6484227
[13,] 1.0538738 1.2078812 9.929672 9.648050 4.261329 0.6484227 0.0000000
[14,] 9.9404936 10.1228546 2.504693 6.365092 7.951214 10.2778062 10.0555808
[15,] 11.2275919 11.4168257 3.551265 7.502747 8.850740 11.5414000 11.2851603
[16,] 1.2329043 1.3784310 10.045298 9.810665 4.186561 0.7942777 0.1842525
[,14]     [,15]     [,16]
[1,] 9.7748297 10.9467254 0.8051397
[2,] 9.9165125 11.0440493 1.3441287
[3,] 1.0840724 0.3153982 11.0439651
[4,] 0.8997939 0.4928002 10.8898039
[5,] 8.5779170 9.9628763 4.5739665
```

```

[6,] 10.5108368 11.7331567 0.3838854
[7,] 9.9404936 11.2275919 1.2329043
[8,] 10.1228546 11.4168257 1.3784310
[9,] 2.5046932 3.5512653 10.0452983
[10,] 6.3650925 7.5027468 9.8106646
[11,] 7.9512138 8.8507403 4.1865615
[12,] 10.2778062 11.5414000 0.7942777
[13,] 10.0555808 11.2851603 0.1842525
[14,] 0.0000000 1.3873734 10.1321402
[15,] 1.3873734 0.0000000 11.3513531
[16,] 10.1321402 11.3513531 0.0000000

```

```

[[14]]
[,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.0000000 0.8935418 0.9934531 1.1901681 0.7699279 0.8561688 0.5106878
[2,] 0.8935418 0.0000000 0.4133038 0.5695410 0.2645184 0.4166461 1.2706727
[3,] 0.9934531 0.4133038 0.0000000 0.2032462 0.2573908 0.1448378 1.2142590
[4,] 1.1901681 0.5695410 0.2032462 0.0000000 0.4589651 0.3340135 1.3758205
[5,] 0.7699279 0.2645184 0.2573908 0.4589651 0.0000000 0.1808535 1.0672137
[6,] 0.8561688 0.4166461 0.1448378 0.3340135 0.1808535 0.0000000 1.0706451
[7,] 0.5106878 1.2706727 1.2142590 1.3758205 1.0672137 1.0706451 0.0000000
[8,] 0.7233623 0.8282620 0.5890620 0.7072715 0.5651230 0.4676152 0.7033100
[,8]
[1,] 0.7233623
[2,] 0.8282620
[3,] 0.5890620
[4,] 0.7072715
[5,] 0.5651230
[6,] 0.4676152
[7,] 0.7033100
[8,] 0.0000000

```

Appendix 6: Results for the estimation of the intercept

The results for the intercept deviated significantly from the results for the other regression parameters. In particular, on gridded scenarios, the non-spatial methods as well as some spatial methods (GLMMPQL for example) appeared to provide biased mean estimates: they consistently overestimated the intercept. On irregularly-spaced locations, all the regression methods (except MCMCLH) strongly under-estimated the intercept. For gridded data, MCMCLH and INLA were the only methods that managed to convincingly estimate the intercept, and only MCMCLH remained convincing for irregularly-spaced data.

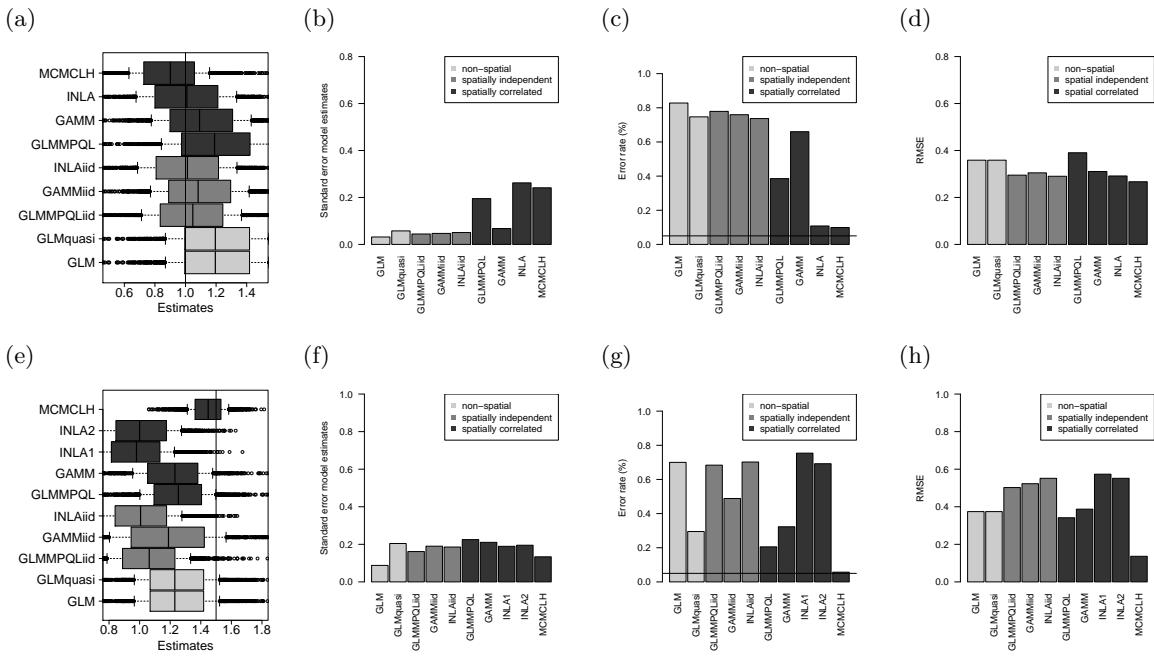


Fig. 6.1. Inferential results for the intercept of the different regression methods (see Table 1) fitted to 1000 simulated datasets from the gridded scenario GRID.4 (plots (a) to (d)) and the irregularly-spaced scenario UNGRID.4 (plots (e) to (h)). Plots (a)-(e) are box-plots of the 1000 mean estimates of the intercept for the different regression methods; the true value of the intercept is indicated by the vertical line. Plots (b)-(f), (c)-(g) and (d)-(h) respectively illustrate the standard error model estimates, the Root Mean Squared Error (RMSE) and the type I error rate for the intercept. In each plot, non-spatial methods are pale grey, spatially-independent methods are intermediate grey and spatially-correlated methods are dark grey.

Appendix 7: Results for the estimation of the intercept and the covariance parameters in over- and under-dispersed cases

The intercept and the covariance parameter estimates are biased in both over- and under-dispersed cases, and the bias increases with the degree of dispersion. The intercept and the variance are under-estimated in over-dispersed cases, and they are over-estimated in under-dispersed case. Conversely, the scale parameter is over-estimated in over-dispersed case and under-estimated in under-dispersed case.

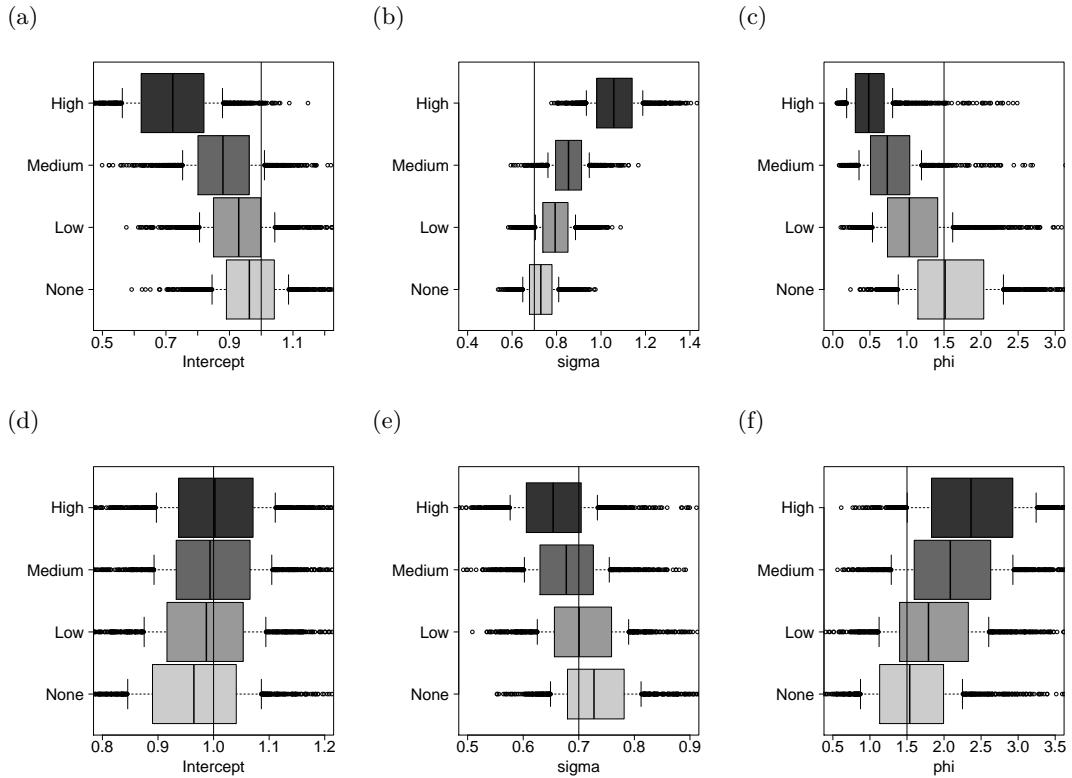


Fig. 7.1. Estimation of the intercept and the covariance parameters for the distance-based method MCMCLH (see Table 1) fitted on over-dispersed (plots (a),(b),(c)) and under-dispersed (plots (d)-(e)-(f)) count data. Three increasing degrees of extra-spatial over-dispersion and under-dispersion were generated as well as one non-dispersed case. One thousand datasets based on the irregularly-spaced scenario UNGRID.4 (see Table 2) were generated for each case. Plots (a)-(d), (b)-(d) and (c)-(f) are box-plots of the 1000 mean estimates for the intercept, the standard deviation of the random effects and the scale parameter respectively. Plots (a)-(e) are box-plots of the 1000 mean estimates of the intercept for the different regression methods; true values for the regression parameters are indicated by the vertical lines.

Appendix 8: Link between the level of RMSE and the shifts among methods

The methods that had a minimal or quasi-minimal degree of RMSE showed only slight shifts among each other. The shifts among methods reflect the differences in RMSE, which clearly illustrates the link between the relative behaviour of regression methods (shifts) and their intrinsic inferential performance (RMSE).

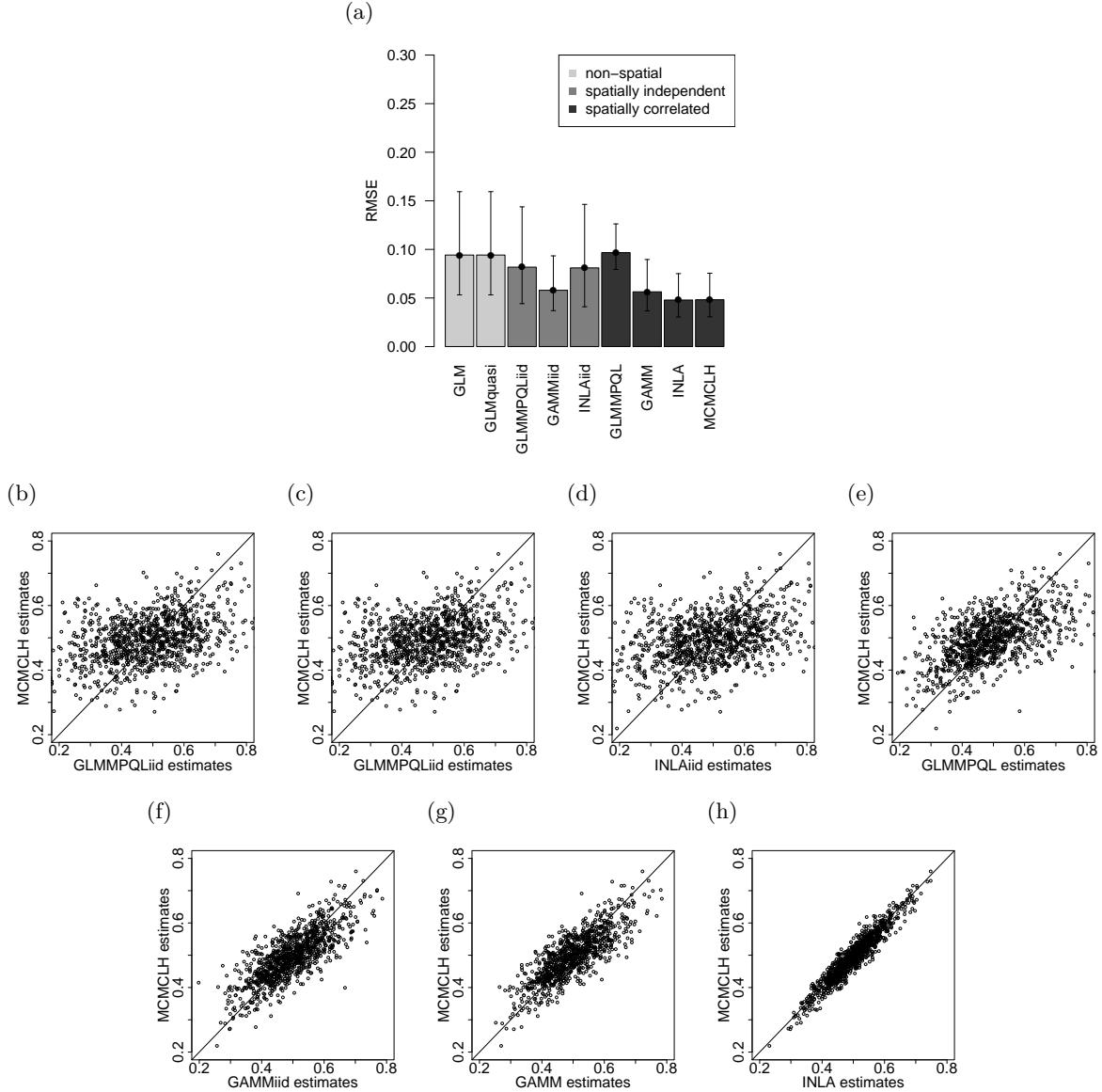


Fig. 8.1. Plot (a) shows the summarized indicator over all parameters (excluding the intercept) of the Root Mean Squared Error (RMSE) for the different regression methods (see Table 1) fitted to 1000 simulated datasets from the representative gridded scenario GRID.4. (see Table 2). Plots (b) to (h) illustrate the differences in mean estimates (shifts) among the different regression methods applied to a single dataset from scenario GRID.4 for one quantitative spatial covariate.

Appendix 9: The effect of sample size on statistical inference

We ran simulations on regular grids, based on scenario GRID.4, to study the effect of sample size on statistical inference. In the first case ($n=400$), the 20×20 grid is completely filled with simulated values. In the next cases ($n=200$, $n=100$), we chose 200 and 100 random locations on the grid, so that respectively half and three-quarters of the observations are considered as missing values. We chose five representative methods (GLM, GLMquasi, INLAiid, INLA, MCMCLH) to illustrate the effect of sample size on statistical inference, in terms of type I errors. For the non-spatial methods (GLM, GLMquasi) and the spatially-independent method (INLAiid), the errors decay with sample size; as the number of missing values increases, the standard error model estimates naturally increase too. For the two spatial methods (INLA, MCMCLH), the type I errors remain stable.

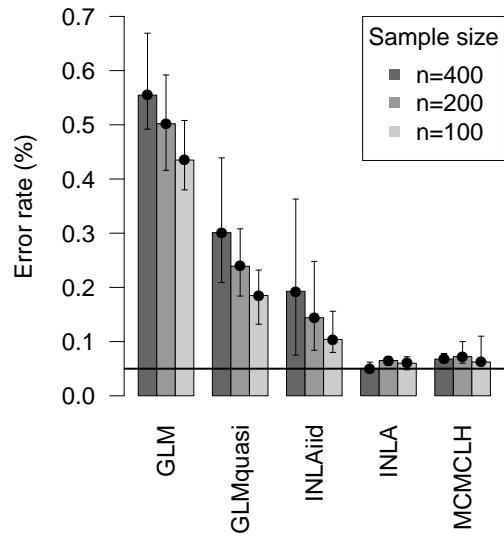


Fig. 9.1. The effect of sample size on statistical inference in terms of type I errors

Appendix 10: Complete results for the gridded scenarios

The following figures present the full results for all the gridded scenarios (GRID.1-GRID.10; see Table 2) for the regression methods described in Table 1. In each case, the results were obtained by fitting the methods to 1000 simulated datasets from each scenario. Plots (a),(b),(d) and (e) are box-plots for each covariate of the 1000 mean estimates for the different regression methods; true regression parameters are indicated by the vertical lines. Plots (c),(f) and (g) respectively illustrate the summarized indicators over all parameters (excluding the intercept) of the standard error model estimates, the Root Mean Squared Error (RMSE) and the type I error rates. The minimum and maximum value among all parameters of the three indicators are enclosed in square brackets. In each plot, non-spatial methods are pale grey, spatially-independent methods are intermediate grey and spatially-correlated methods are dark grey.

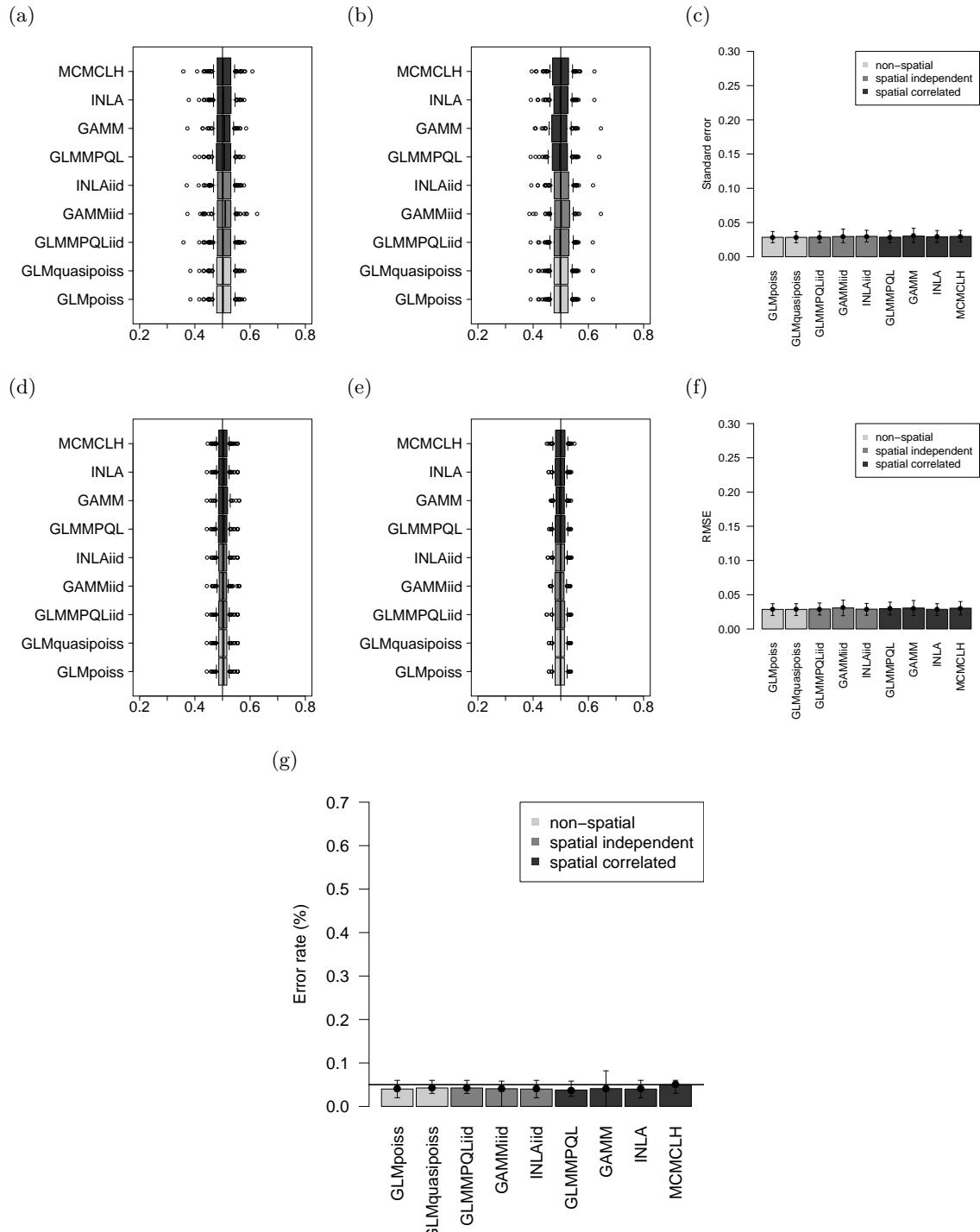


Fig. 10.1. Results for scenario GRID.1

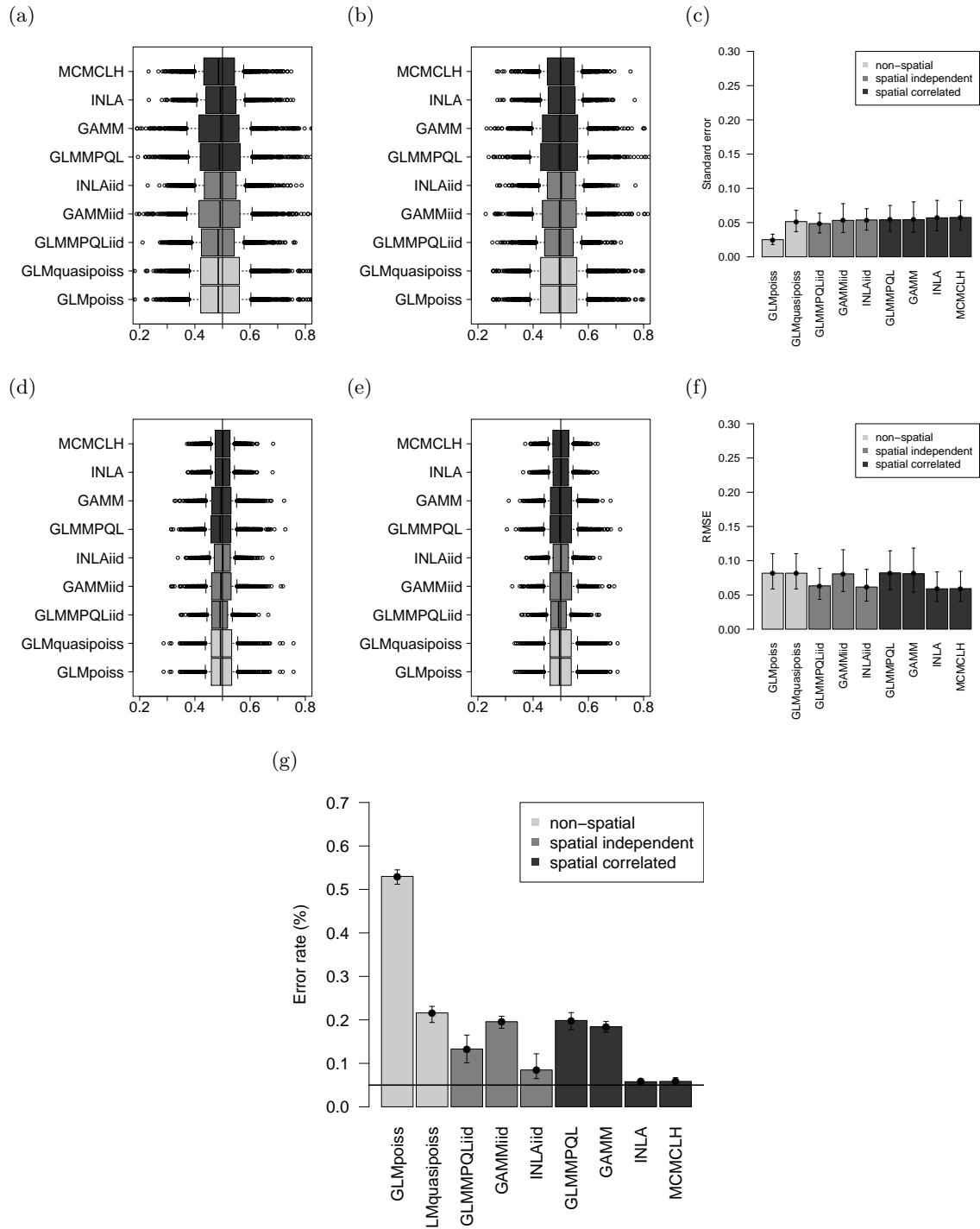


Fig. 10.2. Results for scenario GRID.2

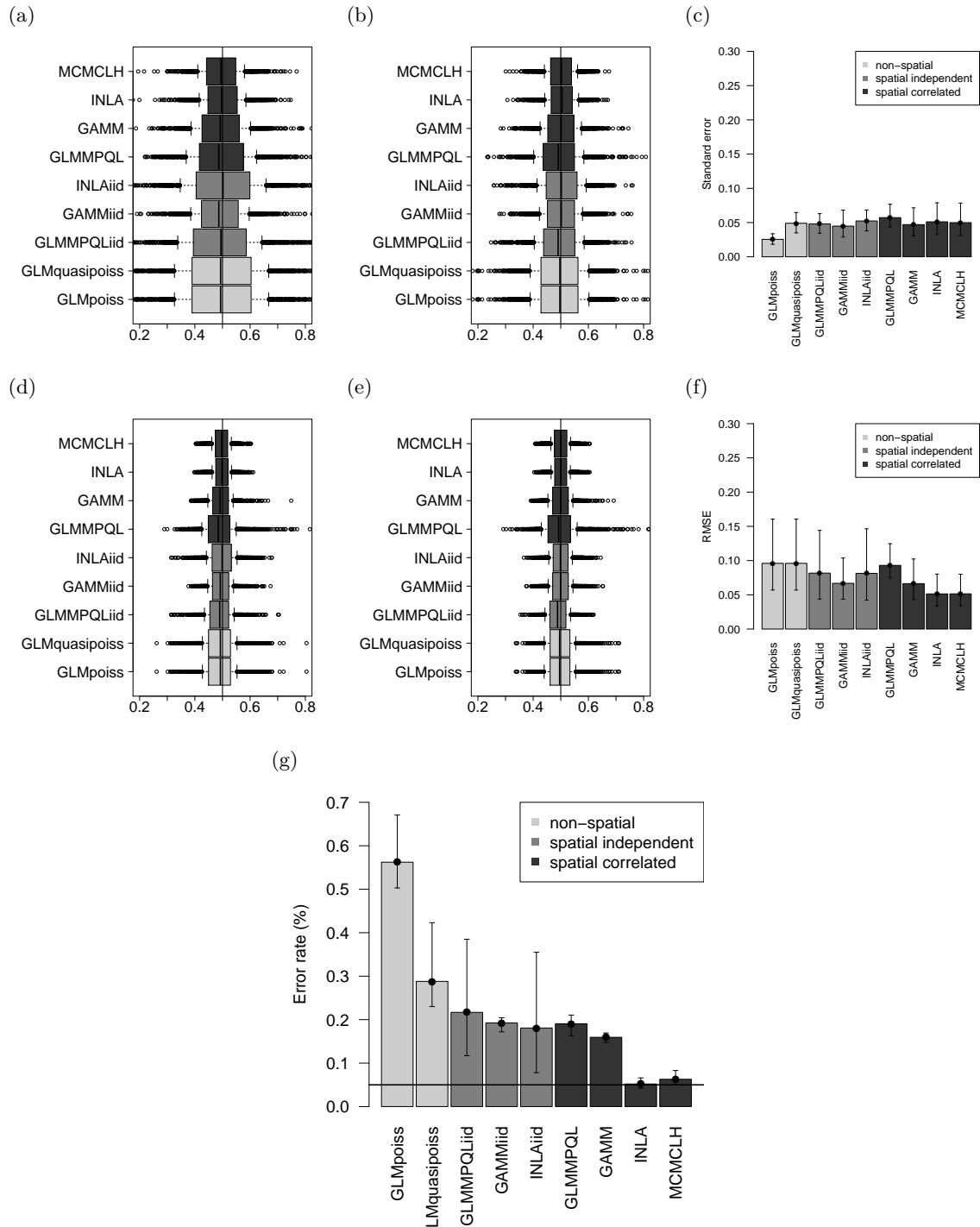


Fig. 10.3. Results for scenario GRID.3

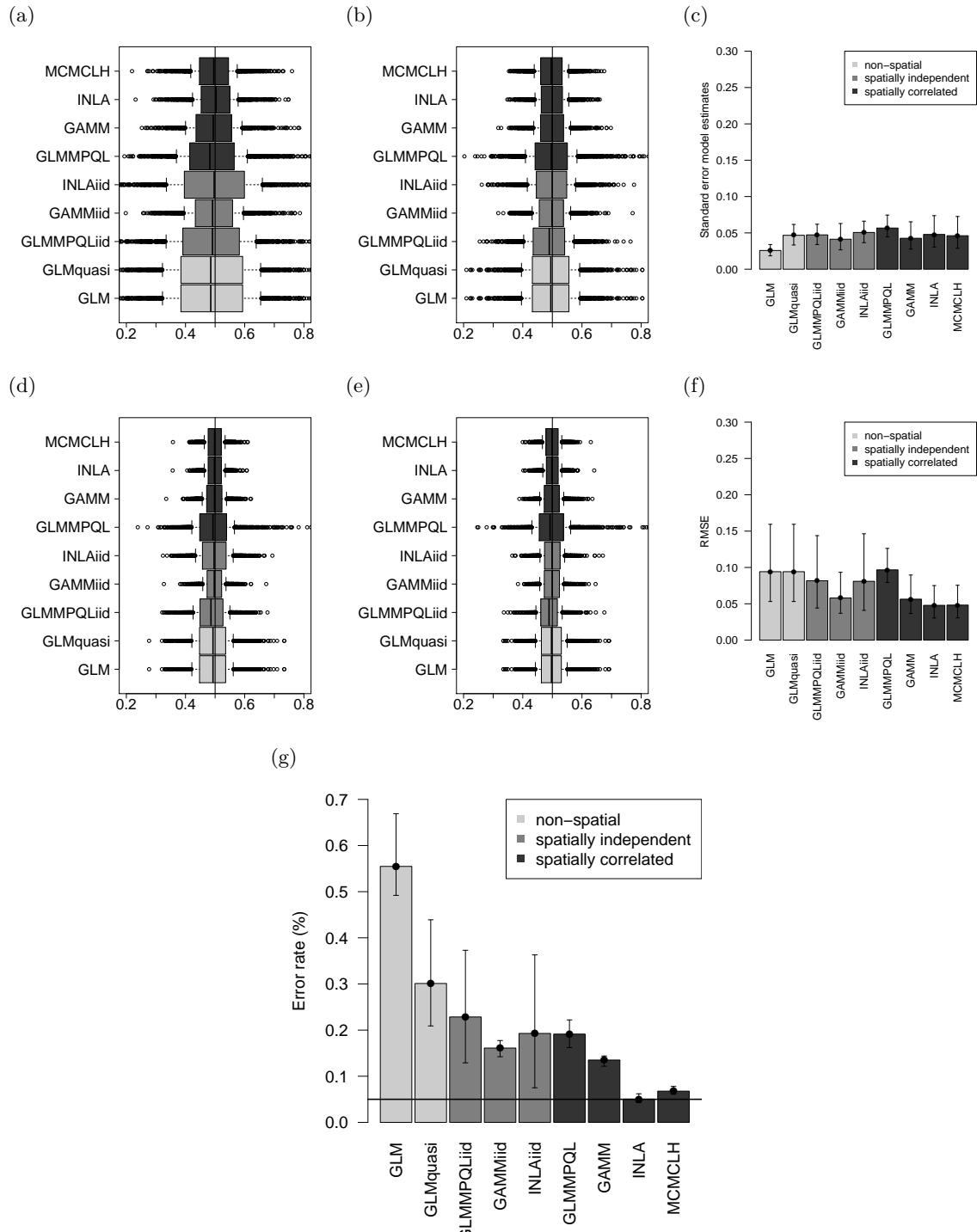


Fig. 10.4. Results for scenario GRID.4

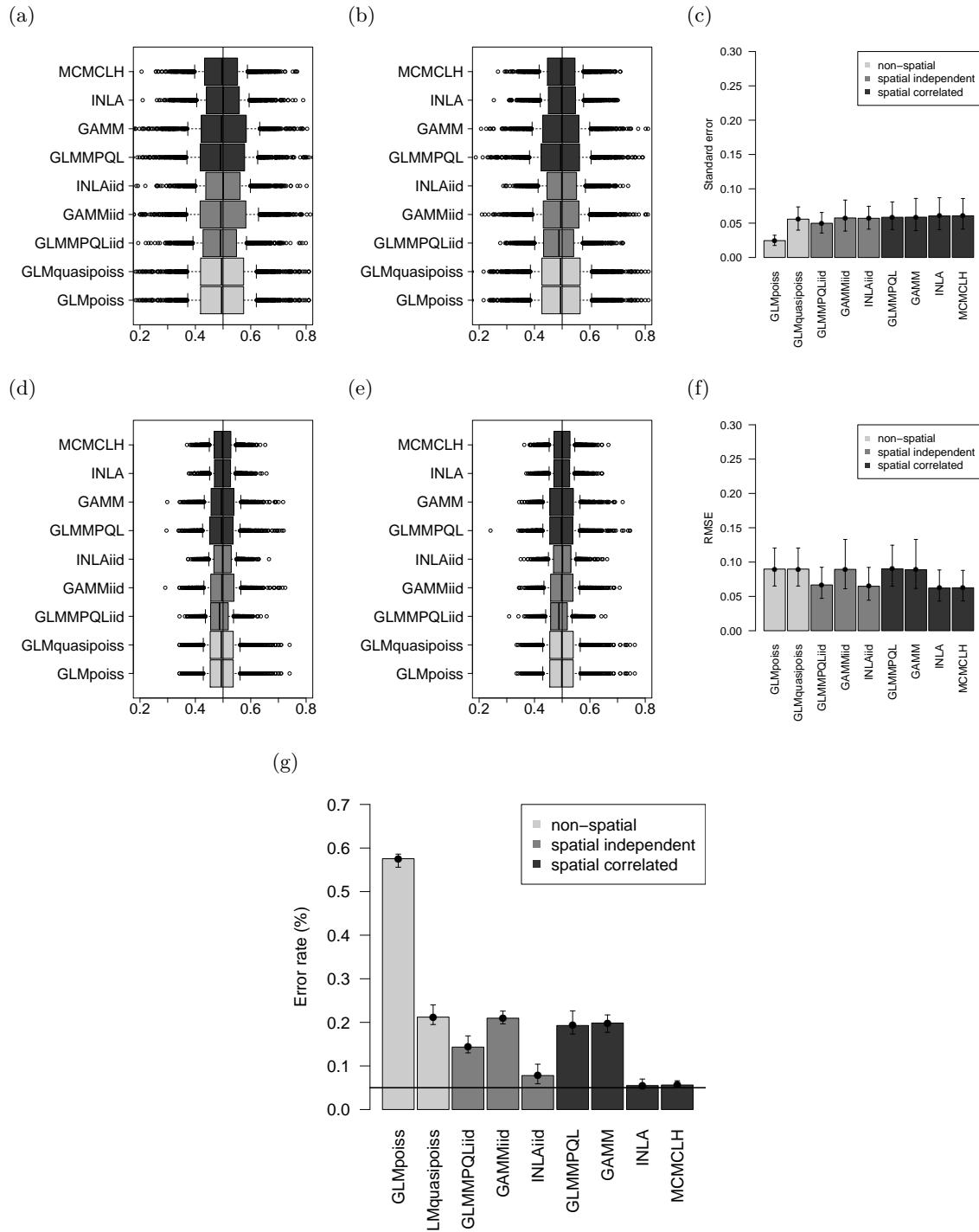


Fig. 10.5. Results for scenario GRID.5

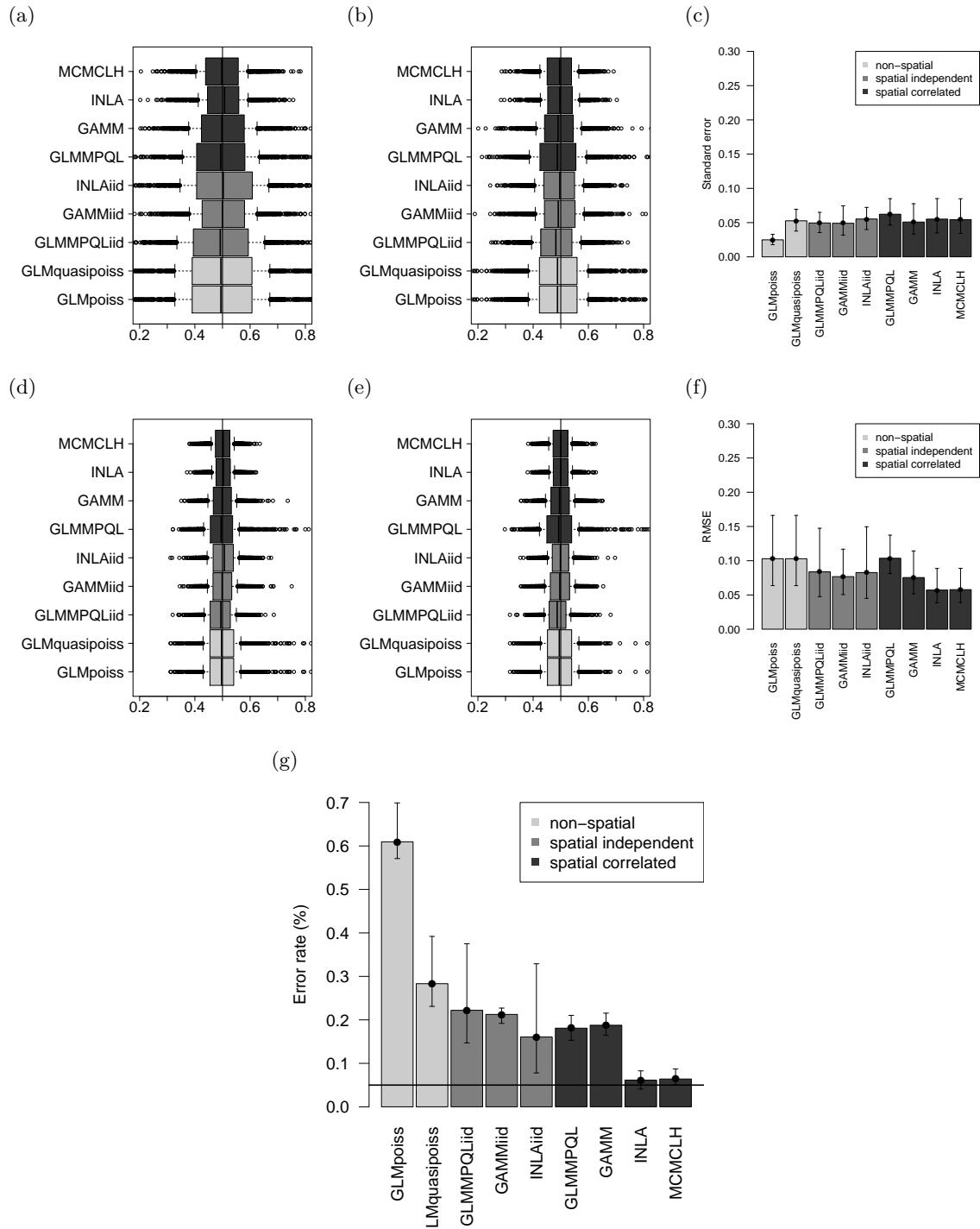


Fig. 10.6. Results for scenario GRID.6

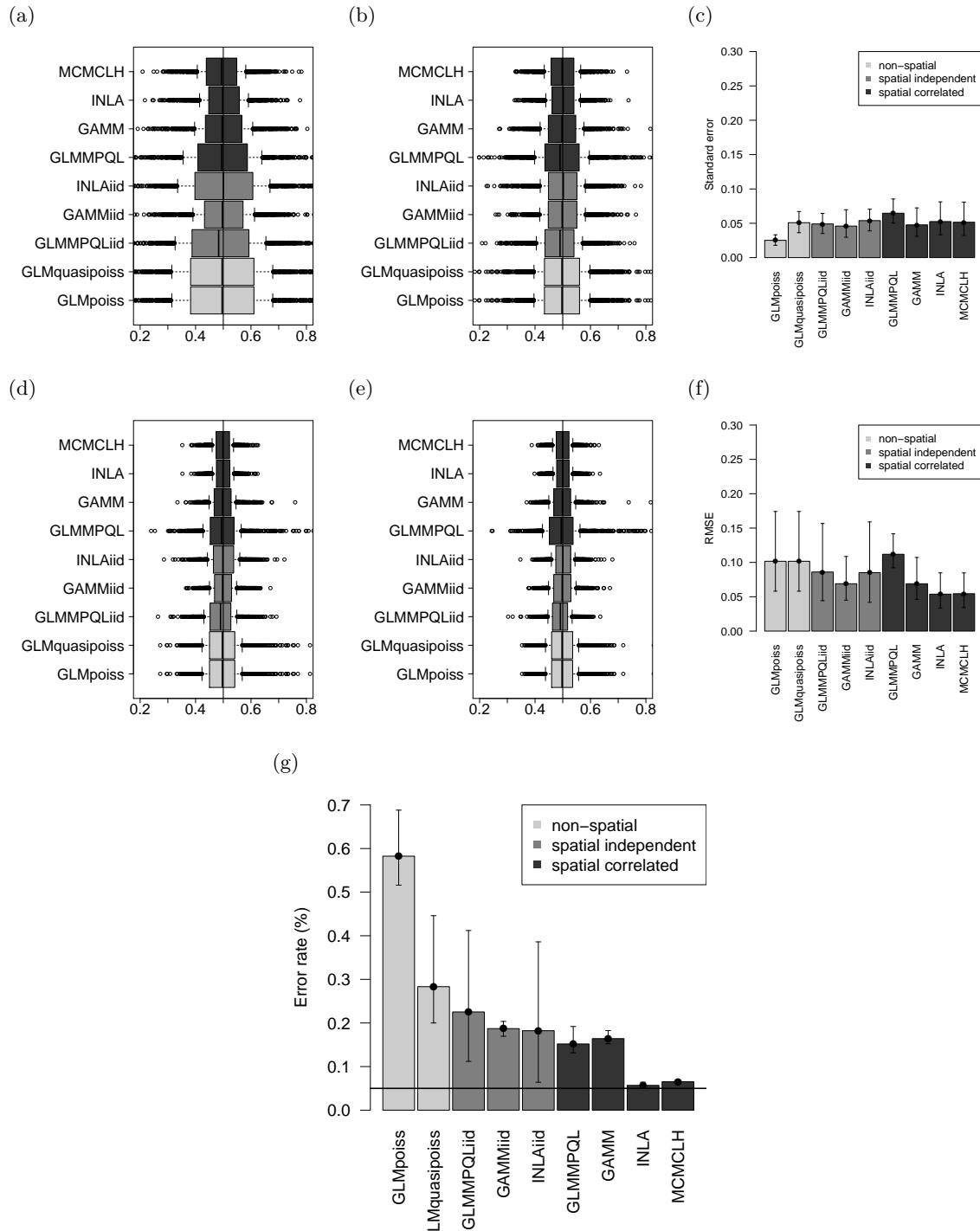


Fig. 10.7. Results for scenario GRID.7

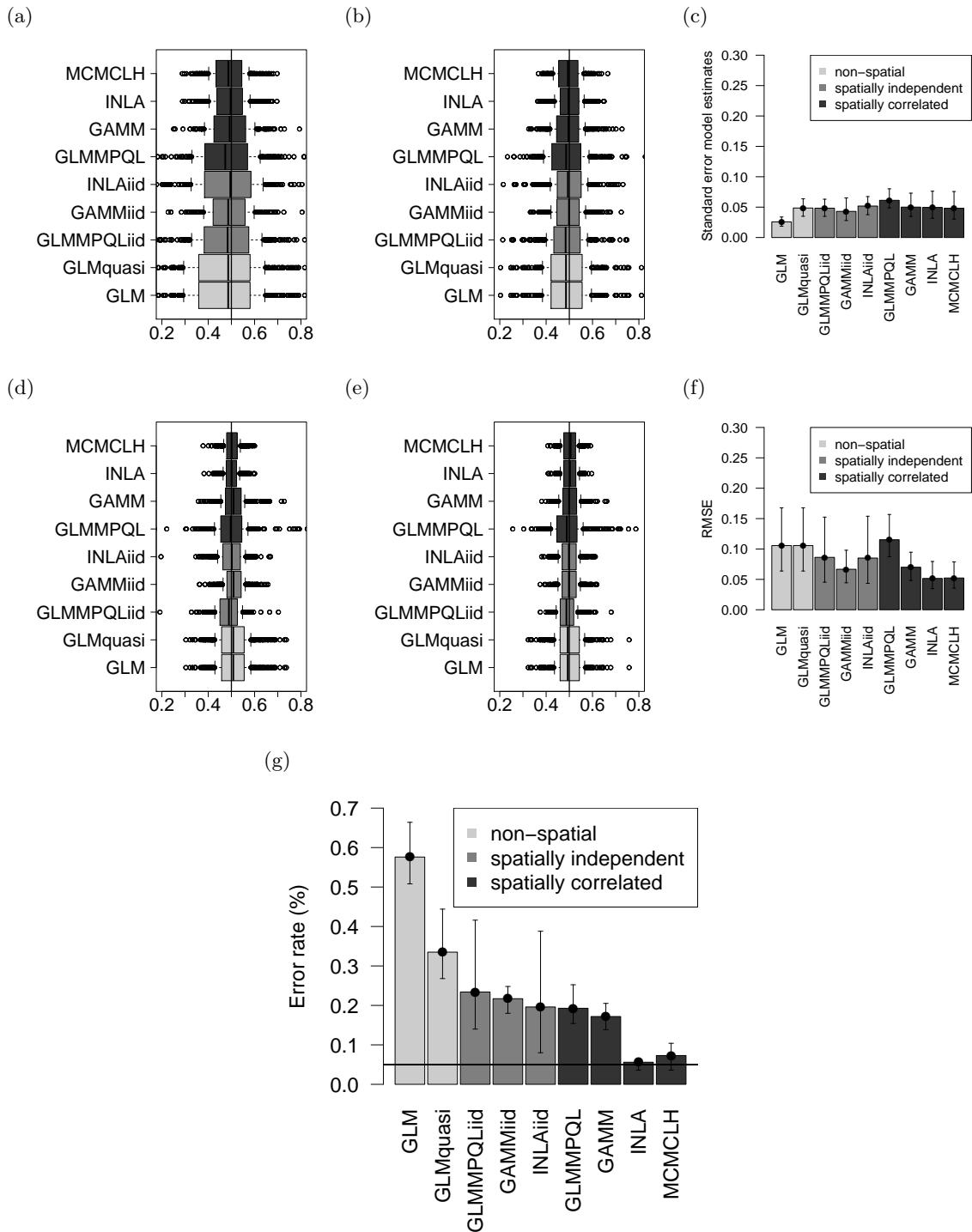


Fig. 10.8. Results for scenario GRID.8

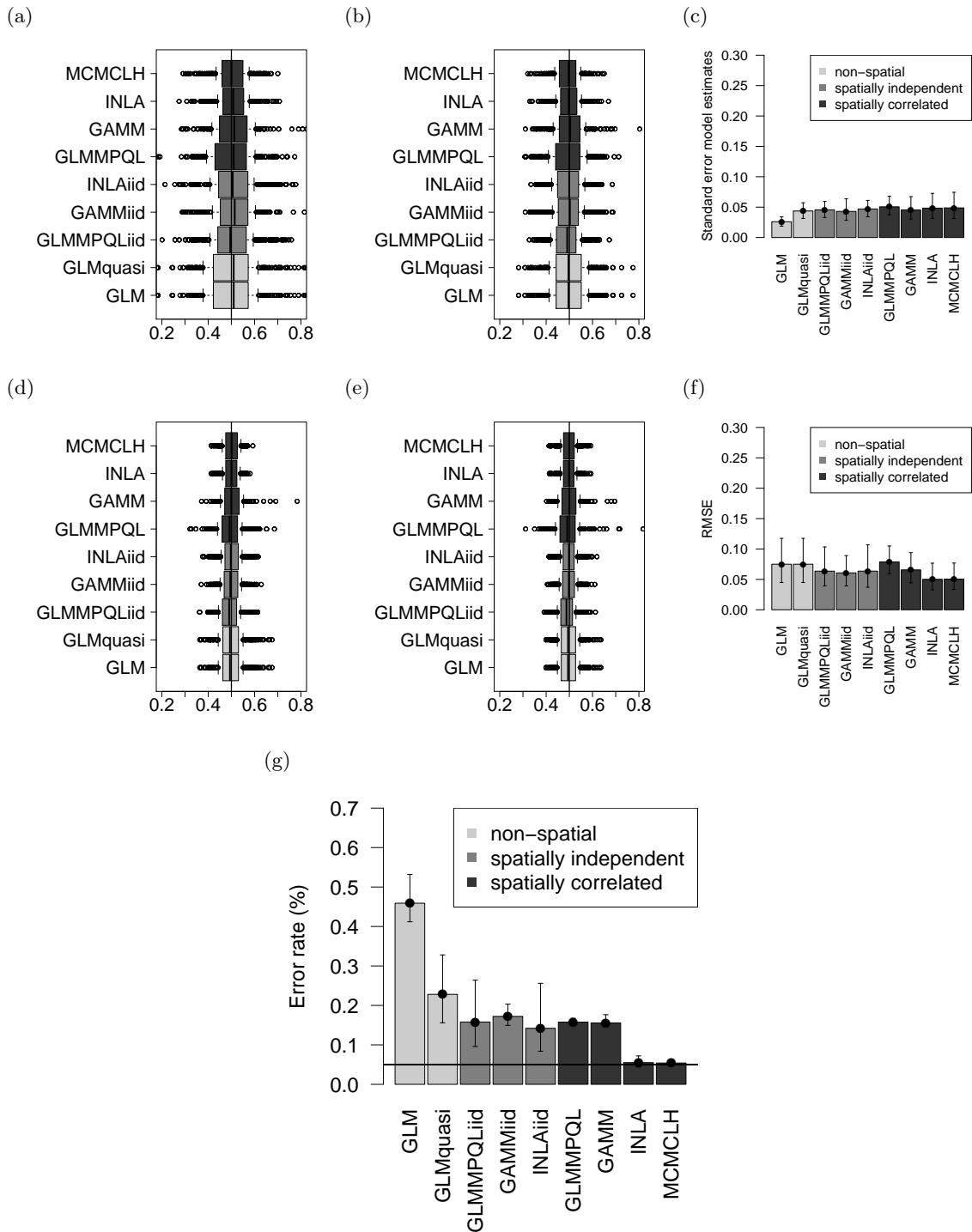


Fig. 10.9. Results for scenario GRID.9

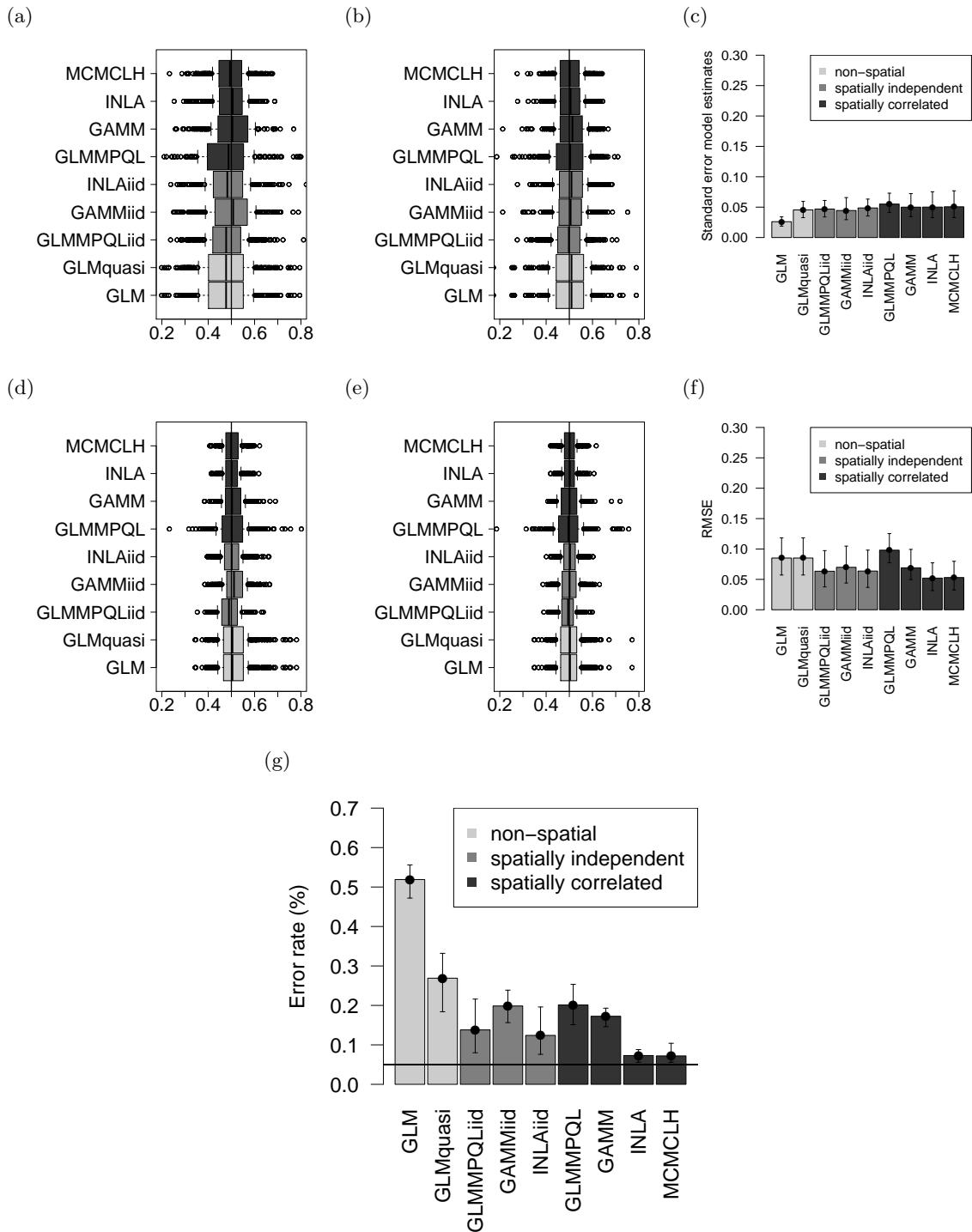


Fig. 10.10. Results for scenario GRID.10

Appendix 11: Complete results for the irregularly-spaced scenarios

The following figures present the full results for all irregularly-spaced scenarios (UNGRID.1-UNGRID.6; see Table 2) for the regression methods described in Table 1. In each case, the results come from fitting the methods to 1000 simulated datasets of each scenario. Plots (a),(b),(d) and (e) are box-plots for each covariate of the 1000 mean estimates for the different regression methods; true regression parameters are indicated by the vertical lines. Plots (c),(f) and (g) respectively illustrate the summarized indicators over all parameters (excluding the intercept) of the standard error model estimates, the Root Mean Squared Error (RMSE) and the type I error rates. The minimum and maximum value among all parameters of the three indicators are enclosed in square brackets. In each plot, non-spatial methods are pale grey, spatially-independent methods are intermediate grey and spatially-correlated methods are dark grey.

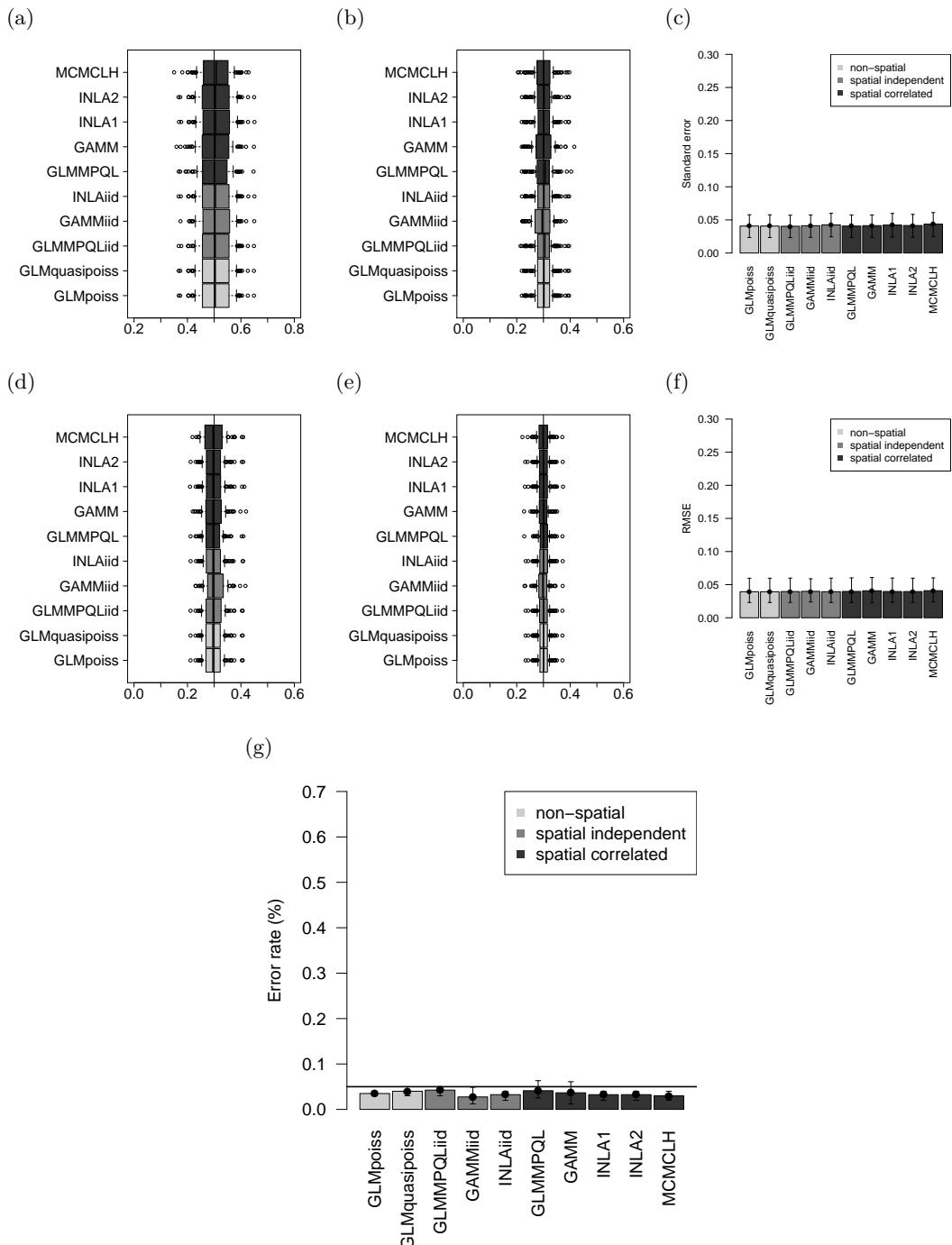


Fig. 11.1. Results for scenario UNGRID.1

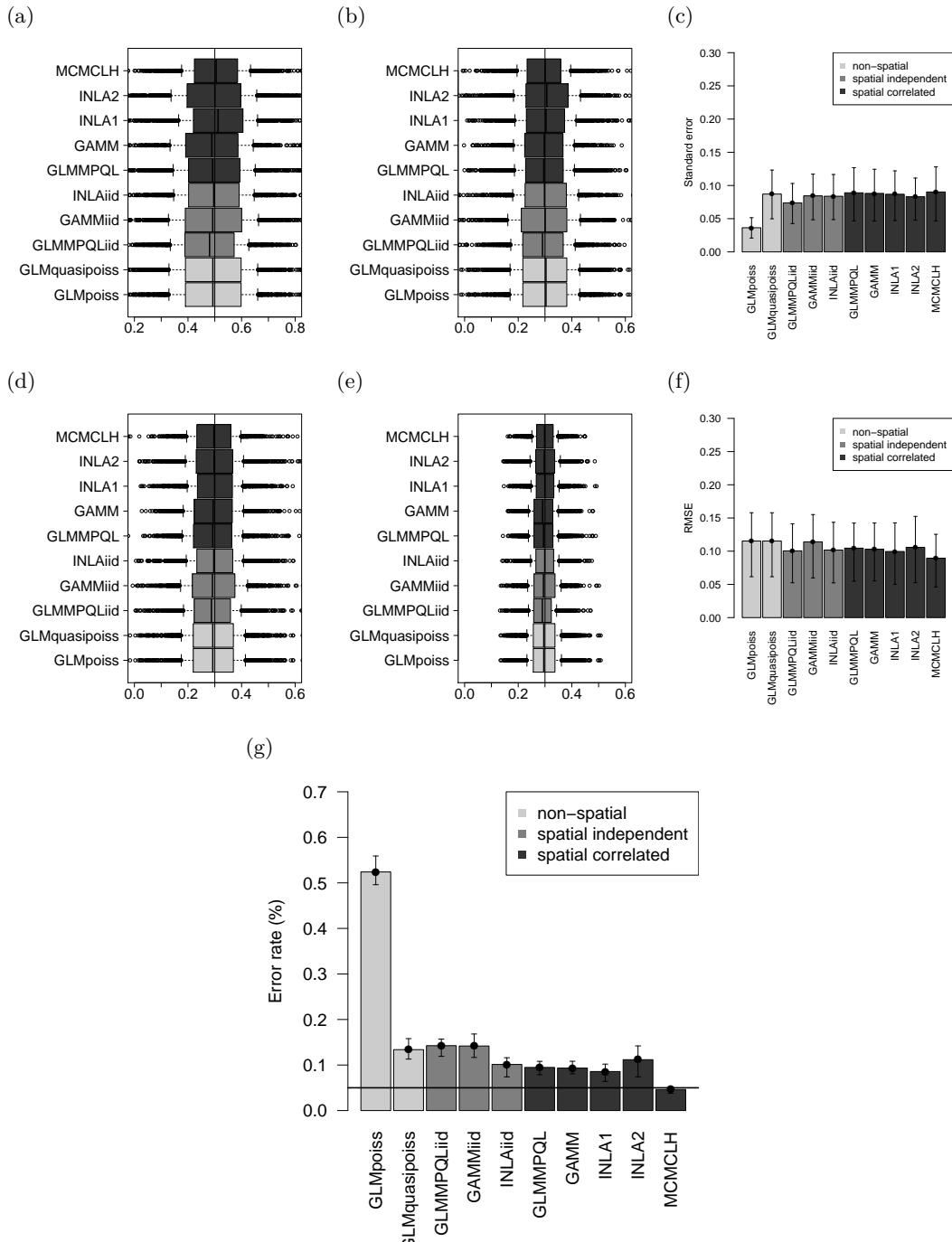


Fig. 11.2. Results for scenario UNGRID.2

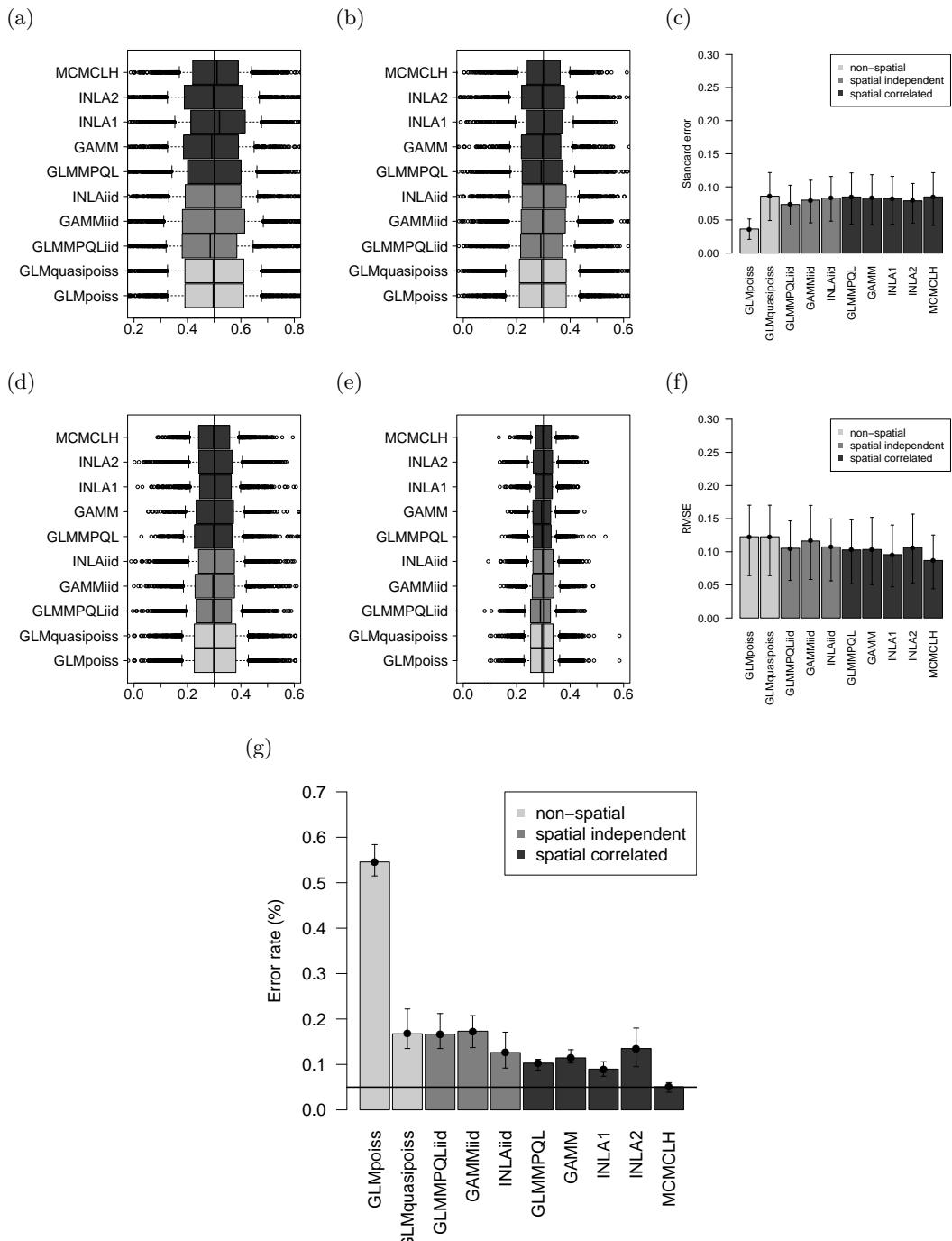


Fig. 11.3. Results for scenario UNGRID.3

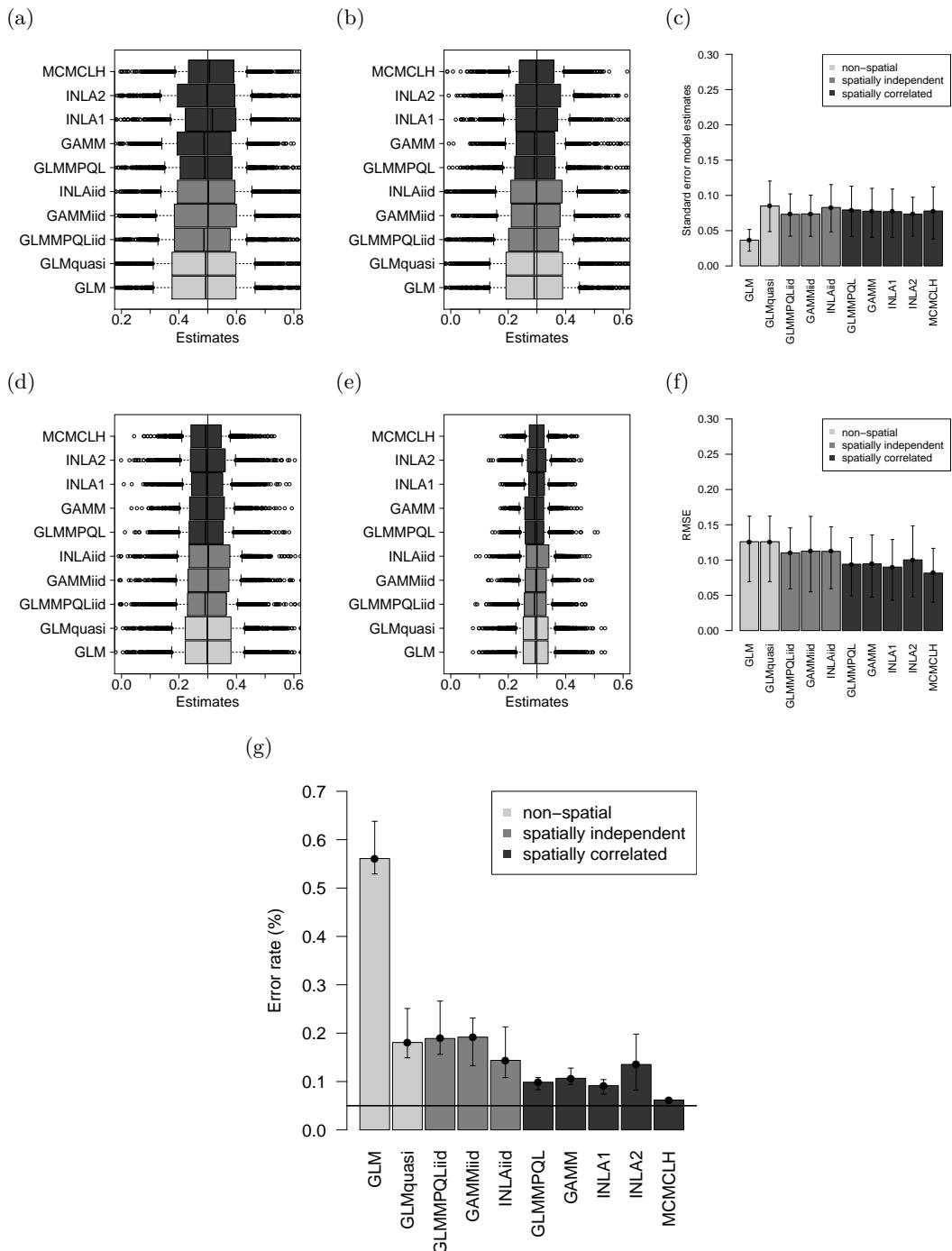


Fig. 11.4. Results for scenario UNGRID.4

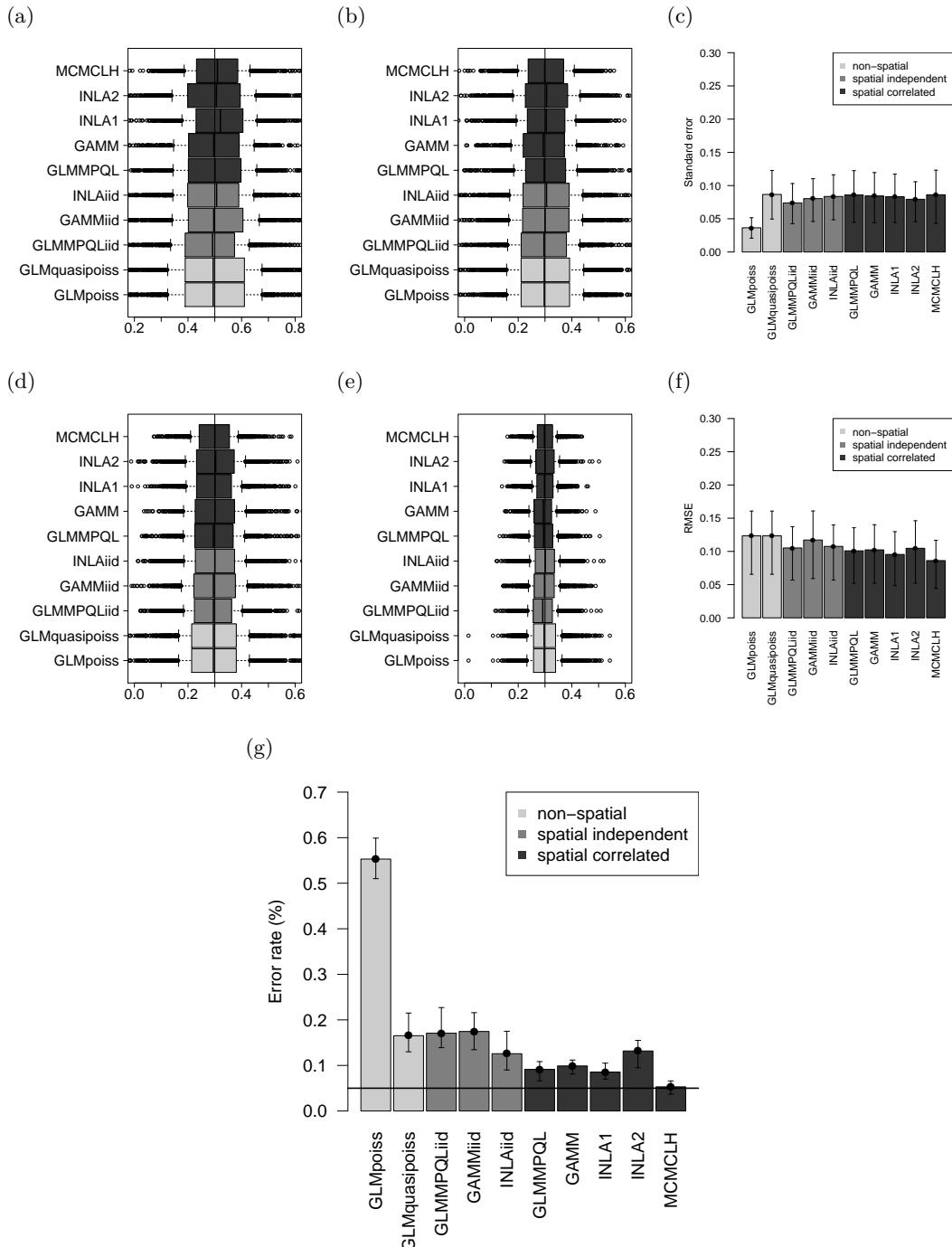


Fig. 11.5. Results for scenario UNGRID.5

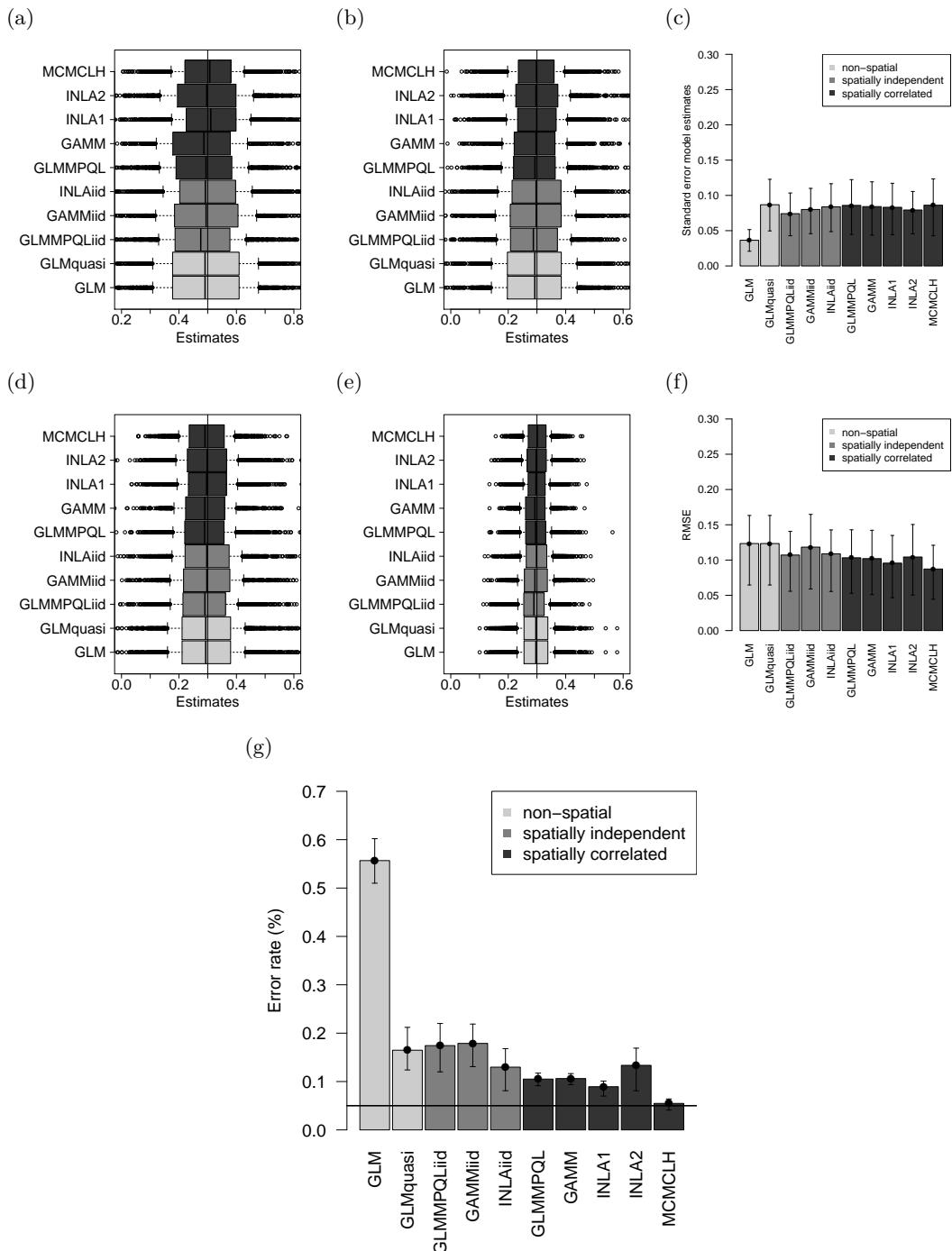


Fig. 11.6. Results for scenario UNGRID.6