# Annex C

RMarkdown document with all the analysis carried out for the paper "Some findings on zero-inflated and hurdle Poisson models for disease mapping"

## 1. Execution of models in WinBUGS using the library R2WinBUGS

### Load libraries, data and cartography

```
# Working directory
DirMain=" " # Set an appropriate directory
setwd(DirMain)
# Load library and data
library(R2WinBUGS)
load("datos/OE.rdata")
load("VR.rdata")
# For running the models in parallel calls to WinBUGS
source("Pbugs.0.4.4.r")
# Load cartography
Cvalenciana<-dget("datos/Cvalenciana.txt")

# Total number of diseases
ndiseases<-46
# Total number of municipalities
nareas<-540
# Total number of observed and expected cases during the whole period of study
Obs<-list()
Exp<-list()
for(i in 1:ndiseases){
  Obs[[i]]<-apply(Obs2[[i]],1,sum)
  Exp[[i]]<-apply(Esp[[i]],1,sum)
}
```

### BYM model

```
# BYM model, WinBUGS code
model.BYM<-function(){
  for(i in 1:n){
    O[i]~dpois(mu[i])
# Modeling of the mean for each municipality
    log(mu[i])<-log(E[i])+m+sd.phi*phi[i]+sd.theta*theta[i]
# SMR for each municipality
    SMR[i]<-exp(m+sd.phi*phi[i]+sd.theta*theta[i])
# Prior distribution for the non-spatial effect
    theta[i]~dnorm(0,1)
# Predictive distribution
    O.pred[i]~dpois(mu[i])
    pred.equal.0[i]<-equals(O.pred[i],0)
```

```
  }
# Predictive distribution for the number of zeroes
  zero.pred<-sum(pred.equal.0[])
# Prior distribution for the spatial effect
  phi[1:n]~car.normal(map[],w[],nvec[],1)
# Prior distribution for the mean risk for all municipalities
  m~dflat()
# Prior distribution for the standard deviations of the random effects
  sd.theta~dunif(0,5)
  sd.phi~dunif(0,5)
}

# Run BYM model for each disease
for(i in 1:ndiseases){
# Working directory
  setwd(paste(DirMain,"/resul/",LabelsCausas[i],sep=""))
# Initial values
  initials<-function(){list(m=rnorm(1,0,0.1),sd.theta=runif(1,0,1),sd.phi=runif(1,0,1),
                             theta=rnorm(nareas),phi=rnorm(nareas))}
# Data
  data<-list(n=nareas,O=Obs[[i]],E=Exp[[i]],map=Cvalenciana$map,w=Cvalenciana$w,
             nvec=Cvalenciana$nvec)
# Variables to retrieve
  param<-c("sd.phi","sd.theta","SMR","mu","zero.pred")
# Calls to WinBUGS
  t.ResulBYM<-system.time(ResulBYM<-Pbugs(data=data,inits=initials,
                                          parameters.to.save=param,model.file=model.BYM,
                                          n.chains=3,n.iter=50000,n.burnin=5000,DIC=F,
                                          working.directory=getwd()))
# Save results
  save(ResulBYM,t.ResulBYM,file="ResulBYM.Rdata")
  setwd<-paste(DirMain)
}
```

## Naive ZIP model

```
# Naive ZIP model, WinBUGS code
model.ZIP<-function(){
  for(i in 1:n){
    O[i]~dpois(mu[i])
    Z[i]~dbern(p)
# Modeling of the mean for each municipality
    log(mu[i])<-log(E[i])+m+sd.phi*phi[i]+sd.theta*theta[i]-1000*(1-Z[i])
# SMR for each municipality
    SMR[i]<-exp(m+sd.phi*phi[i]+sd.theta*theta[i]-1000*(1-Z[i]))
# Prior distribution for the non-spatial effect
    theta[i]~dnorm(0,1)
# Predictive distribution
    O.pred[i]~dpois(mu[i])
    pred.equal.0[i]<-equals(O.pred[i],0)
  }
# Predictive distribution for the number of zeroes
  zero.pred<-sum(pred.equal.0[])
```

```
# Prior distribution for the spatial effect
  phi[1:n]~car.normal(map[],w[],nvec[],1)
# Prior distribution for the mean risk for all municipalities
  m~dflat()
# Prior distribution for the standard deviations of the random effects
  sd.theta~dunif(0,5)
  sd.phi~dunif(0,5)
# Prior distribution for p
  p~dunif(0,1)
}

# Run ZIP model for each disease
for(i in 1:ndiseases){
  setwd(paste(DirMain,"/resul/",LabelsCausas[i],sep=""))
# Initial values
  initials<-function(){list(m=rnorm(1,0,0.1),sd.theta=runif(1,0,1),sd.phi=runif(1,0,1),
                            theta=rnorm(nareas),phi=rnorm(nareas),
                            Z=as.numeric(Obs[[i]]>0))}
# Data
  data<-list(n=nareas,O=Obs[[i]],E=Exp[[i]],map=Cvalenciana$map,w=Cvalenciana$w,
             nvec=Cvalenciana$nvec)
# Variables to retrieve
  param<-c("sd.phi","sd.theta","SMR","mu","p","zero.pred")
# Calls to WinBUGS
  t.ResulZIP<-system.time(ResulZIP<-Pbugs(data=data,inits=initials,
                                          parameters.to.save=param,model.file=model.ZIP,
                                          n.chains=3,n.iter=50000,n.burnin=5000,DIC=F,
                                          working.directory=getwd()))
# Save results
  save(ResulZIP,t.ResulZIP,file="ResulZIP.Rdata")
  setwd<-paste(DirMain)
}
```

## Naive Hurdle model

```
# Naive Hurdle model, WinBUGS code
model.Hurdle<-function(){
# Modeling using the zero trick
  c<-10000
  for(i in 1:n){
    zeros[i]<-0
    zeros[i]~dpois(zeros.mean[i])
    zeros.mean[i]<- -L[i]+c
    Z[i]<-step(O[i]-1)
# Expression of the log-likelihood por i
    L[i]<-(1-Z[i])*log(1-p)+Z[i]*(log(p)+O[i]*log(lambda[i])-lambda[i]-logfact(O[i])-
                                  log(1-exp(-lambda[i])))
# Modeling of the mean Poisson for each municipality
    log(lambda[i])<-log(E[i])+m+sd.phi*phi[i]+sd.theta*theta[i]
# SMR for each municipality
    SMR[i]<-(p*lambda[i]/(1-exp(-lambda[i])))/E[i]
# Prior distribution for the non-spatial effect
    theta[i]~dnorm(0,1)
```

```
# Predictive distribution
    O.pred[i]~dbern(p)
    pred.equal.0[i]<-equals(O.pred[i],0)
  }
# Predictive distribution for the number of zeroes
  zero.pred<-sum(pred.equal.0[])

# Prior distribution for the spatial effect
  phi[1:n]~car.normal(map[],w[],nvec[],1)
# Prior distribution for the mean risk all every municipalities
  m~dflat()
# Prior distribution for the standard deviations of the random effects
  sd.theta~dunif(0,5)
  sd.phi~dunif(0,5)
# Prior distribution for p
  p~dunif(0,1)
}

# Run Hurdle model for each disease
for(i in 1:ndiseases){
  setwd(paste(DirMain,"/resul/",LabelsCausas[i],sep=""))
# Initial values
  initials<-function(){list(m=rnorm(1,0,0.1),sd.theta=runif(1,0,1),sd.phi=runif(1,0,1),
                             theta=rnorm(nareas),phi=rnorm(nareas))}
# Data
  data<-list(n=nareas,O=Obs[[i]],E=Exp[[i]],map=Cvalenciana$map,w=Cvalenciana$w,
             nvec=Cvalenciana$nvec)
# Variables to retrieve
  param<-c("sd.phi","sd.theta","SMR","lambda","p","zero.pred")
# Calls to WinBUGS
  t.ResulHurdle<-system.time(ResulHurdle<-Pbugs(data=data,inits=initials,
                                                parameters.to.save=param,
                                                model.file=model.Hurdle,n.chains=3,
                                                n.iter=50000,n.burnin=5000,DIC=F,
                                                working.directory=getwd()))
# Save results
  save(ResulHurdle,t.ResulHurdle,file="ResulHurdle.Rdata")
  setwd<-paste(DirMain)
}
```

### FE Hurdle model

```
# FE Hurdle model, WinBUGS code
model.HFE<-function(){
# Modeling using the zero trick
  c<-10000
  for(i in 1:n){
    zeros[i]<-0
    zeros[i]~dpois(zeros.mean[i])
    zeros.mean[i]<- (-L[i]+c)
    Z[i]<-step(O[i]-1)
# Expression of the log-likelihood por i
    L[i]<-(1-Z[i])*log(1-p[i])+Z[i]*(log(p[i])+O[i]*log(lambda[i])-lambda[i]-logfact(O[i])
```

```
                                   -log(1-exp(-lambda[i])))
# Modeling of the mean Poisson for each municipality
    log(lambda[i])<-log(E[i])+m+sd.phi*phi[i]+sd.theta*theta[i]
# Modeling p for each municipality
    logit(p[i])<-alpha+beta*LE[i]
# SMR for each municipality
    SMR[i]<-(p[i]*lambda[i]/(1-exp(-lambda[i])))/E[i]
# Prior distribution for the non-spatial effect
    theta[i]~dnorm(0,1)
# Predictive distribution
    O.pred[i]~dbern(p[i])
    pred.equal.0[i]<-equals(O.pred[i],0)
  }
# Predictive distribution for the number of zeroes
  zero.pred<-sum(pred.equal.0[])

# Prior distribution for the spatial effect
  phi[1:n]~car.normal(map[],w[],nvec[],1)
# Prior distribution for the mean risk for all municipalities
  m~dflat()
# Prior distribution for the standard deviations of the random effects
  sd.theta~dunif(0,5)
  sd.phi~dunif(0,5)
# Prior distribution for the parameters logistic regression
  alpha~dflat()
  beta~dflat()
}

# Run FE Hurdle model for each disease
for(i in 1:ndiseases){
  setwd(paste(DirMain,"/resul/",LabelsCausas[i],sep=""))
# Initial values
  initials<-function(){list(m=rnorm(1,0,0.1),sd.theta=runif(1,0,1),sd.phi=runif(1,0,1),
                          theta=rnorm(nareas),phi=rnorm(nareas),alpha=rnorm(1,0,0.1),
                          beta=rnorm(1,0,0.1))}
# Data
  data<-list(n=nareas,O=Obs[[i]],E=Exp[[i]],LE=log(Exp[[i]])-mean(log(Exp[[i]])),
           map=Cvalenciana$map,w=Cvalenciana$w,nvec=Cvalenciana$nvec)
# Variables to retrieve
  param<-c("sd.phi","sd.theta","SMR","lambda","p","alpha","beta","zero.pred")
# Calls to WinBUGS
  t.ResulHFE<-system.time(ResulHFE<-Pbugs(data=data,inits=initials,
                                        parameters.to.save=param,model.file=model.HFE,
                                        n.chains=3,n.iter=50000,n.burnin=5000,DIC=F,
                                        working.directory=getwd()))
# Save results
  save(ResulHFE,t.ResulHFE,file="ResulHFE.Rdata")
  setwd<-paste(DirMain)
}
```

## NFE Hurdle model

```
# NFE Hurdle model, WinBUGS code
model.HNFE<-function(){
# Modeling using the zero trick
  c<-10000
  for(i in 1:n){
    zeros[i]<-0
    zeros[i]~dpois(zeros.mean[i])
    zeros.mean[i]<- (-L[i]+c)
    Z[i]<-step(O[i]-1)
# Expression of the log-likelihood por i
    L[i]<-(1-Z[i])*log(1-p[i])+Z[i]*(log(p[i])+O[i]*log(lambda[i])-lambda[i]-logfact(O[i])
                                     -log(1-exp(-lambda[i])))
# Modeling of the mean Poisson for each municipality
    log(lambda[i])<-log(E[i])+m+sd.phi*phi[i]+sd.theta*theta[i]
# Modeling p for each municipality
    logit(p[i])<-logit(1-exp(-lambda[i]))+gamma
# SMR for each municipality
    SMR[i]<-(p[i]*lambda[i]/(1-exp(-lambda[i])))/E[i]
# Prior distribution for the non-spatial effect
    theta[i]~dnorm(0,1)
# Predictive distribution
    O.pred[i]~dbern(p[i])
    pred.equal.0[i]<-equals(O.pred[i],0)
  }
# Predictive distribution for the number of zeroes
  zero.pred<-sum(pred.equal.0[])

# Prior distribution for the spatial effect
  phi[1:n]~car.normal(map[],w[],nvec[],1)
# Prior distribution for the mean risk for all municipalities
  m~dflat()
# Prior distribution for the standard deviations of the random effects
  sd.theta~dunif(0,5)
  sd.phi~dunif(0,5)
# Prior distribution for the parameters logistic regression
  gamma~dflat()
}

# Run NFE Hurdle model for each disease
for(i in 1:ndiseases){
  setwd(paste(DirMain,"/resul/",LabelsCausas[i],sep=""))
# Initial values
  initials<-function(){list(m=rnorm(1,0,0.1),sd.theta=runif(1,0,1),sd.phi=runif(1,0,1),
                            theta=rnorm(nareas),phi=rnorm(nareas),gamma=rnorm(1,0,0.1))}
# Data
  data<-list(n=nareas,O=Obs[[i]],E=Exp[[i]],map=Cvalenciana$map,w=Cvalenciana$w,
             nvec=Cvalenciana$nvec)
# Variables to retrieve
  param<-c("sd.phi","sd.theta","SMR","lambda","p","gamma","zero.pred")
# Calls to WinBUGS
  t.ResulHNFE<-system.time(ResulHNFE<-Pbugs(data=data,inits=initials,
                                            parameters.to.save=param,
```

```
                                              model.file=model.HNFE,n.chains=3,
                                              n.iter=50000,n.burnin=5000,DIC=F,
                                              working.directory=getwd()))
# Save results
  save(ResulHNFE,t.ResulHNFE,file="ResulHNFE.Rdata")
  setwd<-paste(DirMain)
}
```

### HGeo model

```
# HGeo model, WinBUGS code
model.HGeo<-function(){
# Modeling using the zero trick
  c<-10000
  for(i in 1:n){
    zeros[i]<-0
    zeros[i]~dpois(zeros.mean[i])
    zeros.mean[i]<- (-L[i]+c)
    Z[i]<-step(O[i]-1)
# Expression of the log-likelihood por i
    L[i]<-(1-Z[i])*log(1-p[i])+Z[i]*(log(p[i])+O[i]*log(lambda[i])-lambda[i]-logfact(O[i])
                                     -log(1-exp(-lambda[i])))
# Modeling of the mean Poisson for each municipality
    log(lambda[i])<-log(E[i])+m+sd.phi*phi[i]+sd.theta*theta[i]
# Modeling p for each municipality
    p[i]<-1-pow((1-pi),E[i])
# SMR for each municipality
    SMR[i]<-(p[i]*lambda[i]/(1-exp(-lambda[i])))/E[i]
# Prior distribution for the non-spatial effect
    theta[i]~dnorm(0,1)
# Predictive distribution
    O.pred[i]~dbern(p[i])
    pred.equal.0[i]<-equals(O.pred[i],0)
    }
# Predictive distribution for the number of zeroes
  zero.pred<-sum(pred.equal.0[])

# Prior distribution for the spatial effect
  phi[1:n]~car.normal(map[],w[],nvec[],1)
# Prior distribution for the mean risk for all municipalities
  m~dflat()
# Prior distribution for the standard deviations of the random effects
  sd.theta~dunif(0,5)
  sd.phi~dunif(0,5)
# Prior distribution for pi
  pi~dunif(0,1)
}

# Run HGeo model for each disease
for(i in 2:ndiseases){
  setwd(paste(DirMain,"/resul/",LabelsCausas[i],sep=""))
# Initial values
  initials<-function(){list(m=rnorm(1,0,0.1),sd.theta=runif(1,0,1),sd.phi=runif(1,0,1),
```

```
                              theta=rnorm(nareas),phi=rnorm(nareas),pi=runif(1,0,1))}
# Data
  data<-list(n=nareas,O=Obs[[i]],E=Exp[[i]],map=Cvalenciana$map,w=Cvalenciana$w,
             nvec=Cvalenciana$nvec)
# Variables to retrieve
  param<-c("sd.phi","sd.theta","SMR","lambda","p","pi","zero.pred")
# Calls to WinBUGS
  t.ResulHGeo<-system.time(ResulHGeo<-Pbugs(data=data,inits=initials,
                                            parameters.to.save=param,model.file=model.HGeo,
                                            n.chains=3,n.iter=50000,n.burnin=5000,DIC=F,
                                            working.directory=getwd()))
# Save results
  save(ResulHGeo,t.ResulHGeo,file="ResulHGeo.Rdata")
  setwd<-paste(DirMain)
}
```

### ZGeo model

```
# ZIP model, WinBUGS code
model.ZGeo<-function(){
  for(i in 1:n){
    O[i]~dpois(mu[i])
    Z[i]~dbern(p[i])
# Modeling p for each municipality
    p[i]<-1-pow((1-pi),E[i])
#Modeling of the mean for each municipality
    log(mu[i])<-log(E[i])+m+sd.phi*phi[i]+sd.theta*theta[i]-1000*(1-Z[i])
    lambda[i]<-E[i]*exp(m+sd.phi*phi[i]+sd.theta*theta[i])
# SMR for each municipality
    SMR[i]<-exp(m+sd.phi*phi[i]+sd.theta*theta[i]-1000*(1-Z[i]))
# Prior distribution for the non-spatial effect
    theta[i]~dnorm(0,1)
# Predictive distribution
    O.pred[i]~dpois(mu[i])
    pred.equal.0[i]<-equals(O.pred[i],0)
  }
# Predictive distribution for the number of zeroes
  zero.pred<-sum(pred.equal.0[])

# Prior distribution for the spatial effect
  phi[1:n]~car.normal(map[],w[],nvec[],1)
# Prior distribution for the mean risk for all municipalities
  m~dflat()
# Prior distribution for the standard deviations of the random effects
  sd.theta~dunif(0,5)
  sd.phi~dunif(0,5)
# Prior distribution for pi
  pi~dunif(0,1)
}

# Run ZGeo model for each disease
for(i in 1:ndiseases){
  setwd(paste(DirMain,"/resul/",LabelsCausas[i],sep=""))
```

```
# Initial values
  initials<-function(){list(m=rnorm(1,0,0.1),sd.theta=runif(1,0,1),sd.phi=runif(1,0,1),
                        theta=rnorm(nareas),phi=rnorm(nareas),Z=as.numeric(Obs[[i]]>0),
                        pi=runif(1,0,1))}
# Data
  data<-list(n=nareas,O=Obs[[i]],E=Exp[[i]],map=Cvalenciana$map,w=Cvalenciana$w,
             nvec=Cvalenciana$nvec)
# Variables to retrieve
  param<-c("sd.phi","sd.theta","SMR","mu","lambda","p","pi","zero.pred")
# Calls to WinBUGS
  t.ResulZGeo<-system.time(ResulZGeo<-Pbugs(data=data,inits=initials,
                                            parameters.to.save=param,model.file=model.ZGeo,
                                            n.chains=3,n.iter=50000,n.burnin=5000,DIC=F,
                                            working.directory=getwd()))
# Save results
  save(ResulZGeo,t.ResulZGeo,file="ResulZGeo.Rdata")
  setwd<-paste(DirMain)
}
```

## 2. Comparison observed zeroes for each data set and posterior predicted zeroes for each model (Tables 1 and 2 in Annex B)

```
# Load libraries
library(xtable)
library(pander)
library(rmarkdown)
library(knitr)

# Posterior predicted zeroes for each model
zeros_BYM<-character()
zeros_ZIP<-character()
zeros_Hurdle<-character()
zeros_HFE<-character()
zeros_HNFE<-character()
zeros_HGeo<-character()
zeros_ZGeo<-character()

for(i in 1:ndiseases){
# Load WinBUGS results
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulBYM.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulZIP.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHurdle.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHFE.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHNFE.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHGeo.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulZGeo.Rdata",sep=""))

# Posterior predicted medians for zeroes for each model run
# and corresponding unilateral 95% posterior predictive intervals
zeros_BYM[i]<-paste0(round(summary(ResulBYM$sims.list$zero.pred)[3])," [0,",
                     round(quantile(ResulBYM$sims.list$zero.pred,p=0.95)),"]")
zeros_ZIP[i]<-paste0(round(summary(ResulZIP$sims.list$zero.pred)[3])," [0,",
```

```
                      round(quantile(ResulZIP$sims.list$zero.pred,p=0.95)),"]")
zeros_Hurdle[i]<-paste0(round(summary(ResulHurdle$sims.list$zero.pred)[3])," [0,",
                         round(quantile(ResulHurdle$sims.list$zero.pred,p=0.95)),"]")
zeros_HFE[i]<-paste0(round(summary(ResulHFE$sims.list$zero.pred)[3])," [0,",
                      round(quantile(ResulHFE$sims.list$zero.pred,p=0.95)),"]")
zeros_HNFE[i]<-paste0(round(summary(ResulHNFE$sims.list$zero.pred)[3])," [0,",
                       round(quantile(ResulHNFE$sims.list$zero.pred,p=0.95)),"]")
zeros_HGeo[i]<-paste0(round(summary(ResulHGeo$sims.list$zero.pred)[3])," [0,",
                       round(quantile(ResulHGeo$sims.list$zero.pred,p=0.95)),"]")
zeros_ZGeo[i]<-paste0(round(summary(ResulZGeo$sims.list$zero.pred)[3])," [0,",
                       round(quantile(ResulZGeo$sims.list$zero.pred,p=0.95)),"]")
}

Disease<-c("(Men, All tumours)","(Women, All tumours)","(Men, Mouth)","(Men, Stomach)",
           "(Women, Stomach)","(Men, Colorectal)","(Women, Colorectal)","(Men, Colon)",
           "(Women, Colon)","(Men, Rectum)","(Women, Rectum)","(Men, Liver)",
           "(Women, Liver)","(Women, Vesicle)","(Men, Pancreas)","(Women, Pancreas)",
           "(Men, Larynx)","(Men, Lung)","(Women, Lung)","(Women, Breast)",
           "(Women, Uterus)","(Women, Ovary)","(Men, Prostate)","(Men, Bladder)",
           "(Men, Lymphatic)","(Women, Lymphatic)","(Men, Leukemia)",
           "(Women, Leukemia)","(Men, Diabetes)","(Women, Diabetes)",
           "(Men, Hypertensive)","(Women, Hypertensive)","(Men, Ischemic)",
           "(Women, Ischemic)","(Men, Cerebrovascular)","(Women, Cerebrovascular)",
           "(Men, Atherosclerosis)","(Women, Atherosclerosis)",
           "(Men, Other Cardiovascular)","(Women, Other Cardiovascular)",
           "(Men, Pneumonia)","(Women, Pneumonia)","(Men, COPD)","(Women, COPD)",
           "(Men, Cirrhosis)","(Women, Cirrhosis)")

Table<-cbind(Disease,unlist(lapply(Obs,function(x){sum(x==0)})),zeros_BYM,zeros_ZIP,
             zeros_Hurdle,zeros_HFE,zeros_HNFE,zeros_HGeo,zeros_ZGeo)
colnames(Table)<-c("Sex & Cause","Obs. zeroes","BYM","ZIP","Hurdle","HFE","HNFE","HGeo",
                   "ZGeo")

kable(Table,split.table=Inf,row.names=FALSE,align='c',caption="Observed zeroes for each
      data set and posterior predicted zeroes for each model. Values in the Obs. zeroes
      column correspond to the real observed zeroes for each data set. For the 5 columns
      on the right, numbers correspond to the posterior predictive median for this same
      quantity for each model run and the corresponding unilateral 95% posterior
      predictive interval.")
```

# 3. DIC for each model (Table 3 in Annex B)

```
# DIC BYM model

CalculaDIC_BYM<-function(Simu,O,E,save=FALSE){
  mu<-t(apply(Simu$sims.list$SMR,1,function(x){x*E}))
  D<-apply(mu,1,function(x){-2*sum(O*log(x)-x-lfactorial(O))})
  Dmedia<-mean(D)
  mumedia<-apply(Simu$sims.list$SMR,2,mean)*E
  DenMedia<- -2*sum(O*log(mumedia)-mumedia-lfactorial(O))
  if(save==TRUE){return(c(Dmedia,Dmedia-DenMedia,2*Dmedia-DenMedia))}
  cat("D=",Dmedia,"pD=",Dmedia-DenMedia,"DIC=",2*Dmedia-DenMedia,"\n")
```

```
}

# DIC Hurdle FE, Hurdle NFE and HGeo models

CalculaDIC_Hurdle<-function(Simu,O,E,save=FALSE){
  log.verosim<-matrix(nrow=Simu$n.sims,ncol=length(O))
  Z<-as.numeric(O>0)
    for(j in 1:Simu$n.sims){
      for(k in 1:length(O)){
        if(Z[k]==0){log.verosim[j,k]<-log(1-Simu$sims.list$p[j,k])}
        if(Z[k]==1){log.verosim[j,k]<-log(Simu$sims.list$p[j,k])+
                      O[k]*log(Simu$sims.list$lambda[j,k])-Simu$sims.list$lambda[j,k]-
                      lfactorial(O[k])-log(1-exp(-Simu$sims.list$lambda[j,k]))}
      }
      }
  D<--2*apply(log.verosim,1,sum)
  Dmedia<-mean(D)

    log.verosimMedia<-c()
    for(k in 1:length(O)){
      if(Z[k]==0){log.verosimMedia[k]<-log(1-Simu$mean$p[k])}
      if(Z[k]==1){log.verosimMedia[k]<-log(Simu$mean$p[k])+
                    O[k]*log(Simu$mean$lambda[k])-Simu$mean$lambda[k]-lfactorial(O[k])-
                    log(1-exp(-Simu$mean$lambda[k]))}
    }

  DenMedia<- -2*sum(log.verosimMedia)
  if(save==TRUE){return(c(Dmedia,Dmedia-DenMedia,2*Dmedia-DenMedia))}
  cat("D=",Dmedia,"pD=",Dmedia-DenMedia,"DIC=",2*Dmedia-DenMedia,"\n")
}

# DIC ZGeo model

CalculaDIC_ZIP<-function(Simu,O,E,save=FALSE){
  log.verosim<-matrix(nrow=Simu$n.sims,ncol=length(O))
  Z<-as.numeric(O>0)
  for(j in 1:Simu$n.sims){
      for(k in 1:length(O)){
        if(Z[k]==0){log.verosim[j,k]<-log((1-Simu$sims.list$p[j,k])+
                                      Simu$sims.list$p[j,k]*dpois(x=O[k],
                                      lambda=Simu$sims.list$lambda[j,k]))}
        if(Z[k]==1){log.verosim[j,k]<-log(Simu$sims.list$p[j,k]*dpois(x=O[k],
                                      lambda=Simu$sims.list$lambda[j,k]))}
      }
      }
  D<--2*apply(log.verosim,1,sum)
  Dmedia<-mean(D)

    log.verosimMedia<-c()
    for(k in 1:length(O)){
      if(Z[k]==0){log.verosimMedia[k]<-log((1-Simu$mean$p[k])+
                                      Simu$mean$p[k]*dpois(x=O[k],
                                      lambda=Simu$mean$lambda[k]))}
```

```r
      if(Z[k]==1){log.verosimMedia[k]<-log(Simu$mean$p[k]*dpois(x=O[k],
                                        lambda=Simu$mean$lambda[k]))}
    }

  DenMedia<- -2*sum(log.verosimMedia)
  if(save==TRUE){return(c(Dmedia,Dmedia-DenMedia,2*Dmedia-DenMedia))}
  cat("D=",Dmedia,"pD=",Dmedia-DenMedia,"DIC=",2*Dmedia-DenMedia,"\n")
}

DIC_BYM<-matrix(nrow=ndiseases,ncol=3)
DIC_HFE<-matrix(nrow=ndiseases,ncol=3)
DIC_HNFE<-matrix(nrow=ndiseases,ncol=3)
DIC_HGeo<-matrix(nrow=ndiseases,ncol=3)
DIC_ZGeo<-matrix(nrow=ndiseases,ncol=3)

for(i in 1:ndiseases){
# Load WinBUGS results
    load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulBYM.Rdata",sep=""))
    load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHFE.Rdata",sep=""))
    load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHNFE.Rdata",sep=""))
    load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHGeo.Rdata",sep=""))
    load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulZGeo.Rdata",sep=""))

# DIC for each model and cause
    DIC_BYM[i,]<-CalculaDIC_BYM(ResulBYM,Obs[[i]],Exp[[i]],save=TRUE)
    DIC_HFE[i,]<-CalculaDIC_Hurdle(ResulHFE,Obs[[i]],Exp[[i]],save=TRUE)
    DIC_HNFE[i,]<-CalculaDIC_Hurdle(ResulHNFE,Obs[[i]],Exp[[i]],save=TRUE)
    DIC_HGeo[i,]<-CalculaDIC_Hurdle(ResulHGeo,Obs[[i]],Exp[[i]],save=TRUE)
    DIC_ZGeo[i,]<-CalculaDIC_ZIP(ResulZGeo,Obs[[i]],Exp[[i]],save=TRUE)
}

Table<-cbind(Disease,round(DIC_BYM,1),round(DIC_HFE,1),round(DIC_HNFE,1),
             round(DIC_HGeo,1),round(DIC_ZGeo,1))
colnames(Table)<-c("Disease",rep(c("D","pD","DIC"),5))
rownames(Table)<-as.character(1:46)
cab<-c("Disease",rep(c("D","pD","DIC"),5))

Table2<-rbind(cab,Table)
rownames(Table2)<-c("",rownames(Table))

addtorow<-list()
addtorow$pos<-list(0)
addtorow$command<-paste0('\\multicolumn{1}{c}{}',paste0(' & \\multicolumn{3}{c}{',
                                                  c("BYM", "FE", "NFE", "HGeo", "ZGeo"),
                                                  '}', collapse=''), '\\\\')

print(xtable(Table2,caption="DIC for each model.",align=rep("c",17)),add.to.row=addtorow,
      include.colnames=F,hline.after=c(-1,0,1,nrow(tabla2)),include.rownames=F,
      comment=FALSE)
```

## 4. Posterior distribution of $\gamma$ in the Hurdle NFE model (Table 4 in Annex B)

```
gamma<-character()

for(i in 1:ndiseases){
# Load WinBUGS NFE results
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHNFE.Rdata",sep=""))

# Posterior mean for gamma in the NFE model and the corresponding 95% posterior interval.
gamma[i]<-paste0(round(ResulHNFE$summary["gamma",1],2)," [",
                 round(ResulHNFE$summary["gamma",3],2)," - ",
                 round(ResulHNFE$summary["gamma",7],2),"]")
}

Table<-cbind(Disease,gamma)
colnames(Table)<-c("Sex & Cause","$\\gamma$")

kable(Table,split.table=Inf,row.names=FALSE,align='c',caption="Posterior distribution of
      $\\gamma$ in the NFE model")
```

## 5. Choropleth maps for all models (Figure 1 in Annex B)

```
# Load libraries
library(RColorBrewer)

cuts_SMR<-c(0,0.67,0.80,0.91,1.10,1.25,1.50)
palette<-brewer.pal(7,"BrBG")[7:1]

for(i in 1:ndiseases){
# Load WinBUGS results
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulBYM.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulZIP.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHurdle.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHFE.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHNFE.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulHGeo.Rdata",sep=""))
load(paste(getwd(),"/resul/",LabelsCausas[i],"/ResulZGeo.Rdata",sep=""))

# SMR estimates, BYM model
plot(VR.cart,col=palette[findInterval(ResulBYM$mean$SMR,cuts_SMR)],
     main=paste0("BYM - ", Disease[i]))
legend("bottomright",c("< 0.67","0.67 - 0.80","0.80 - 0.91","0.91 - 1.10","1.10 - 1.25",
                       "1.25 - 1.50","> 1.50"),title="SMR",border=NULL,fill=palette,
       bty="n")

# SMR estimates, naive ZIP model
plot(VR.cart,col=palette[findInterval(ResulZIP$mean$SMR,cuts_SMR)],
     main=paste0("ZIP - ", Disease[i]))
legend("bottomright",c("< 0.67","0.67 - 0.80","0.80 - 0.91","0.91 - 1.10","1.10 - 1.25",
                       "1.25 - 1.50","> 1.50"),title="SMR",border=NULL,fill=palette,
       bty="n")
```

```
# SMR estimates, naive Hurdle model
plot(VR.cart,col=palette[findInterval(ResulHurdle$mean$SMR,cuts_SMR)],
     main=paste0("Hurdle - ", Disease[i]))
legend("bottomright",c("< 0.67","0.67 - 0.80","0.80 - 0.91","0.91 - 1.10","1.10 - 1.25",
                       "1.25 - 1.50","> 1.50"),title="SMR",border=NULL,fill=palette,
       bty="n")

# SMR estimates, FE model
plot(VR.cart,col=palette[findInterval(ResulHFE$mean$SMR,cuts_SMR)],
     main=paste0("HFE - ", Disease[i]))
legend("bottomright",c("< 0.67","0.67 - 0.80","0.80 - 0.91","0.91 - 1.10","1.10 - 1.25",
                       "1.25 - 1.50","> 1.50"),title="SMR",border=NULL,fill=palette,
       bty="n")

# SMR estimates, NFE model
plot(VR.cart,col=palette[findInterval(ResulHNFE$mean$SMR,cuts_SMR)],
     main=paste0("HNFE - ", Disease[i]))
legend("bottomright",c("< 0.67","0.67 - 0.80","0.80 - 0.91","0.91 - 1.10","1.10 - 1.25",
                       "1.25 - 1.50","> 1.50"),title="SMR",border=NULL,fill=palette,
       bty="n")

# SMR estimates, HGeo model
plot(VR.cart,col=palette[findInterval(ResulHGeo$mean$SMR,cuts_SMR)],
     main=paste0("HGeo - ", Disease[i]))
legend("bottomright",c("< 0.67","0.67 - 0.80","0.80 - 0.91","0.91 - 1.10","1.10 - 1.25",
                       "1.25 - 1.50","> 1.50"),title="SMR",border=NULL,fill=palette,
       bty="n")

# SMR estimates, ZGeo model
plot(VR.cart,col=palette[findInterval(ResulZGeo$mean$SMR,cuts_SMR)],
     main=paste0("ZGeo - ", Disease[i]))
legend("bottomright",c("< 0.67","0.67 - 0.80","0.80 - 0.91","0.91 - 1.10","1.10 - 1.25",
                       "1.25 - 1.50","> 1.50"),title="SMR",border=NULL,fill=palette,
       bty="n")
}
```