

R code to obtain results

1. Execution of models in WinBUGS using the R2WinBUGS and pbugs libraries

Load libraries and data

```
# Working directory
DirMain = " " # Set an appropriate directory
setwd(DirMain)
# Load libraries and data
library(R2WinBUGS) # For running WinBUGS from R
library(knitr)
library(pbugs) # For running the models in parallel calls to WinBUGS
library(spdep) # For preparing information about spatial neighbors of each geographic
# unit to be used in WinBUGS (using the
# poly2nb and nb2WB functions)
load("data.RData")
# Loaded data Obs: 4-dimensional array
# with the observed mortality cases for
# each year of the study, sex, geographic
# unit and disease Exp: 4-dimensional
# array with the expected mortality cases
# for each year of the study, sex,
# geographic unit and disease carto:
# SpatialPolygonsDataFrame of the study
# region
carto.nb <- poly2nb(carto) # Neighbours list of each geographic unit with class nb
carto.wb <- nb2WB(carto.nb) # List with the adjacency vector (carto.wb$adj) and the
# number of neighbors of each geographic
# unit (carto.wb$num) to use in WinBUGS
index <- c(1, cumsum(carto.wb$num)) # Vector to identify the positions
# of the neighbors of each geographic
# unit
causes <- c(1:2, 5, 7, 9:12, 15:21) # Studied mortality causes
sex <- 1 # Mens
```

Multivariate adaptive BYM model

```
# Multivariate adaptive BYM model,
# WinBUGS code
AdaptiveBYM_model <- function() {
  # Likelihood
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      O[i, j] ~ dpois(lambda[i, j])
      # Modeling of the mean for each census
    }
  }
}
```

```

    # tract and disease
    log(lambda[i, j]) <- log(E[i,
      j]) + mu[j] + phi[i, j] +
      sd.theta[j] * theta[i, j]
    # SMR for each census tract and disease
    SMR[i, j] <- exp(mu[j] + phi[i,
      j] + sd.theta[j] * theta[i,
      j])
    # Prior distribution for spatial effects
    phi[i, j] ~ dnorm(mean.phi[i,
      j], prec.phi[i, j])
    # Prior distribution for non-spatial
    # effects
    theta[i, j] ~ dnorm(0, 1)
  }
}
for (i in 1:n.adj) {
  sqrt.c.adj[i] <- sqrt(c[adj[i]])
  for (j in 1:Ndiseases) {
    phi.adj[i, j] <- phi[adj[i],
      j]
  }
}
# Precision of the conditioned
# distribution of spatial effects
for (j in 1:Ndiseases) {
  prec.phi[1, j] <- pow(sd.phi[j],
    -2) * sqrt(c[1]) * sum(sqrt.c.adj[index[1]:index[2]])
  for (i in 2:Nareas) {
    prec.phi[i, j] <- pow(sd.phi[j],
      -2) * sqrt(c[i]) * sum(sqrt.c.adj[(index[i] +
      1):index[i + 1]])
  }
}
# Mean of the conditioned distribution of
# spatial effects
for (j in 1:Ndiseases) {
  mean.phi[1, j] <- inprod2(sqrt.c.adj[index[1]:index[2]],
    phi.adj[index[1]:index[2], j])/sum(sqrt.c.adj[index[1]:index[2]])
  for (i in 2:Nareas) {
    mean.phi[i, j] <- inprod2(sqrt.c.adj[(index[i] +
      1):index[i + 1]], phi.adj[(index[i] +
      1):index[i + 1], j])/sum(sqrt.c.adj[(index[i] +
      1):index[i + 1]])
  }
  # Sum-to-zero restriction for spatial
  # effects
  ceros[j] <- 0
  ceros[j] ~ dnorm(sum.phi[j], 10)
  sum.phi[j] <- sum(phi[, j])
}
# Prior distributions for c
for (i in 1:Nareas) {

```

```

      c[i] ~ dgamma(tau, tau) %_~ I(0.001,
    )
  }
  tau <- pow(sd.c, -2)
  sd.c ~ dunif(0, 5)
  # Other prior distributions
  for (j in 1:Ndiseases) {
    sd.phi[j] ~ dunif(0, 5)
    sd.theta[j] ~ dunif(0, 5)
    mu[j] ~ dflat()
  }
}

# Object where the results for each set
# of diseases will be saved
results.AdaptiveBYM <- list()
# Run multivariate adaptive BYM model for
# each set of diseases
for (i in 1:15) {
  # Selection of mortality causes
  causes.id <- causes[-c(i)]
  # Data
  data <- list(0 = apply(Obs[, sex, , causes.id],
    c(2, 3), sum), E = apply(Exp[, sex,
    , causes.id], c(2, 3), sum), Nareas = dim(Obs)[3],
    Ndiseases = length(causes.id), n.adj = length(carto.wb$adj),
    adj = carto.wb$adj, index = index)
  # Initial values
  initials <- function() {
    list(mu = rnorm(data$Ndiseases, 0,
      1), sd.phi = runif(data$Ndiseases,
      0, 1), sd.theta = runif(data$Ndiseases,
      0, 1), phi = matrix(rnorm(data$Nareas *
      data$Ndiseases), nrow = data$Nareas,
      ncol = data$Ndiseases), theta = matrix(rnorm(data$Nareas *
      data$Ndiseases), nrow = data$Nareas,
      ncol = data$Ndiseases), c = runif(data$Nareas,
      0.9, 1.1), sd.c = runif(1, 0.5,
      0.6))
  }
  # Variables to retrieve
  param <- c("mu", "lambda", "sd.phi",
    "phi", "sd.theta", "theta", "SMR",
    "c", "sd.c", "tau")
  # Calls to WinBUGS
  results.AdaptiveBYM[[i]] <- pbugs(data = data,
    inits = initials, parameters.to.save = param,
    model = AdaptiveBYM_model, n.iter = 2e+05,
    n.burnin = 50000, n.chains = 3, DIC = F)
}

# Save results
save(results.AdaptiveBYM, file = "Results/results.AdaptiveBYM.RData")

```

Multivariate adaptive Leroux model

```

# Multivariate adaptive Leroux model,
# WinBUGS code
AdaptiveLeroux_model <- function() {
  # Likelihood
  for (i in 1:Nareas) {
    for (j in 1:Ndiseases) {
      O[i, j] ~ dpois(lambda[i, j])
      # Modeling of the mean for each census
      # tract and disease
      log(lambda[i, j]) <- log(E[i,
        j]) + mu[j] + eta[i, j]
      # SMR for each census tract and disease
      SMR[i, j] <- exp(mu[j] + eta[i,
        j])
      # Prior distribution for spatial effects
      eta[i, j] ~ dnorm(mean.eta[i,
        j], prec.eta[i, j])
    }
  }
  for (i in 1:n.adj) {
    sqrt.c.adj[i] <- sqrt(c[adj[i]])
    for (j in 1:Ndiseases) {
      eta.adj[i, j] <- eta[adj[i],
        j]
    }
  }
  # Precision of the conditioned
  # distribution of spatial effects
  for (j in 1:Ndiseases) {
    prec.eta[1, j] <- pow(sd.eta[j],
      -2) * sqrt(c[1]) * (rho[j] *
        sum(sqrt.c.adj[index[1]:index[2]]) +
        1 - rho[j])
    for (i in 2:Nareas) {
      prec.eta[i, j] <- pow(sd.eta[j],
        -2) * sqrt(c[i]) * (rho[j] *
          sum(sqrt.c.adj[(index[i] +
            1):index[i + 1]]) + 1 -
            rho[j])
    }
  }
  # Mean of the conditioned distribution of
  # spatial effects
  for (j in 1:Ndiseases) {
    mean.eta[1, j] <- (rho[j] * inprod2(sqrt.c.adj[index[1]:index[2]],
      eta.adj[index[1]:index[2], j]))/(rho[j] *
        sum(sqrt.c.adj[index[1]:index[2]]) +
        1 - rho[j])
    for (i in 2:Nareas) {
      mean.eta[i, j] <- (rho[j] * inprod2(sqrt.c.adj[(index[i] +
        1):index[i + 1]], eta.adj[(index[i] +
        1):index[i + 1], j]))/(rho[j] *

```

```

        sum(sqrt.c.adj[(index[i] +
            1):index[i + 1]]) + 1 -
        rho[j])
    }
    # Sum-to-zero restriction for spatial
    # effects
    ceros[j] <- 0
    ceros[j] ~ dnorm(sum.eta[j], 10)
    sum.eta[j] <- sum(eta[, j])
}
# Prior distributions for c
for (i in 1:Nareas) {
    c[i] ~ dgamma(tau, tau) %_ I(0.001,
    )
}
tau <- pow(sd.c, -2)
sd.c ~ dunif(0, 5)
# Other prior distributions
for (j in 1:Ndiseases) {
    mu[j] ~ dflat()
    sd.eta[j] ~ dunif(0, 5)
    rho[j] ~ dunif(0, 1)
}
}
# Object where the results for each set
# of diseases will be saved
results.AdaptiveLeroux <- list()
# Run multivariate adaptive Leroux model
# for each set of diseases
for (i in 1:15) {
    # Selection of mortality causes
    causes.id <- causes[-c(i)]
    # Data
    data <- list(0 = apply(Obs[, sex, , causes.id],
        c(2, 3), sum), E = apply(Exp[, sex,
        , causes.id], c(2, 3), sum), Nareas = dim(Obs)[3],
        Ndiseases = length(causes.id), n.adj = length(carto.wb$adj),
        adj = carto.wb$adj, index = index)
    # Initial values
    initials <- function() {
        list(mu = rnorm(data$Ndiseases, 0,
            1), sd.eta = runif(data$Ndiseases,
            0, 1), rho = runif(data$Ndiseases,
            0, 1), eta = matrix(rnorm(data$Nareas *
            data$Ndiseases), nrow = data$Nareas,
            ncol = data$Ndiseases), c = runif(data$Nareas,
            0.9, 1.1), sd.c = runif(1, 0.5,
            1.5))
    }
    # Variables to retrieve
    param <- c("mu", "lambda", "sd.eta",
        "SMR", "c", "sd.c", "rho")
    # Calls to WinBUGS

```

```

    results.AdaptiveLeroux[[i]] <- pbugs(data = data,
      inits = initials, parameters.to.save = param,
      model = AdaptiveLeroux_model, n.iter = 2e+05,
      n.burnin = 50000, n.chains = 3, DIC = F)
  }
  # Save results
  save(results.AdaptiveLeroux, file = "Results/results.AdaptiveLeroux.RData")

```

Univariate BYM model with spatial weights matrices of either unitary weights or using the values obtained from the multivariate analysis of 14 diseases

```

# Univariate BYM model, WinBUGS code
BYM_model <- function() {
  # Likelihood
  for (i in 1:Nareas) {
    O[i] ~ dpois(lambda[i])
    # Modeling of the mean for each census
    # tract
    log(lambda[i]) <- log(E[i]) + mu +
      sd.phi * phi[i] + sd.theta *
      theta[i]
    # SMR for each census tract
    SMR[i] <- exp(mu + sd.phi * phi[i] +
      sd.theta * theta[i])
    # Prior distribution for non-spatial
    # effects
    theta[i] ~ dnorm(0, 1)
  }
  # Prior distribution for spatial effects
  phi[1:Nareas] ~ car.normal(adj[], w[],
    num[], 1)
  # Other prior distributions
  sd.phi ~ dunif(0, 5)
  sd.theta ~ dunif(0, 5)
  mu ~ dflat()
}
# Object where the results for each
# disease will be saved
results.BYM <- list()
# Run BYM model for each disease
for (i in 1:15) {
  # ATTENTION: Specify spatial weights
  # For unitary weights
  w <- rep(1, length(carto.wb$adj))
  # For adaptive weights:
  index_neighbors <- cbind(rep(1:dim(carto)[1],
    carto.wb$num), carto.wb$adj)
  w <- c(sqrt(results.AdaptiveBYM[[i]]$mean$c[index_neighbors[,
    1]]) * sqrt(results.AdaptiveBYM[[i]]$mean$c[index_neighbors[,
    2]]))
  # Data
  data <- list(O = apply(Obs[, sex, , causes[i]],
    2, sum), E = apply(Exp[, sex, , causes[i]],

```

```

    2, sum), Nareas = dim(Obs)[3], adj = carto.wb$adj,
    w = w, num = carto.wb$num)
# Initial values
initials <- function() {
  list(mu = rnorm(1, 0, 1), sd.phi = runif(1,
    0, 1), sd.theta = runif(1, 0,
    1), phi = rnorm(data$Nareas),
    theta = rnorm(data$Nareas))
}
# Variables to retrieve
param <- c("mu", "lambda", "sd.phi",
  "phi", "sd.theta", "theta", "SMR")
results.BYM[[i]] <- pbugs(data = data,
  inits = initials, parameters.to.save = param,
  model = BYM_model, n.iter = 1e+05,
  n.burnin = 30000, n.chains = 3, DIC = F)
}
# For the model with unitary weights
save(results.BYM, file = "Results/results.BYM.unitaryw.RData")
# For the model with adaptive weights
save(results.BYM, file = "Results/results.BYM.adaptw.RData")

```

Univariate Leroux model with spatial weights matrices of either unitary weights or using the values obtained from the multivariate analysis of 14 diseases

```

# Univariate Leroux model, WinBUGS code
Leroux_model <- function() {
  # Likelihood
  for (i in 1:Nareas) {
    O[i] ~ dpois(lambda[i])
    # Modeling of the mean for each census
    # tract
    log(lambda[i]) <- log(E[i]) + mu +
      sd.eta * eta[i]
    # SMR for each census tract
    SMR[i] <- exp(mu + sd.eta * eta[i])
    # Prior distribution for spatial effects
    eta[i] ~ dnorm(mean.eta[i], prec.eta[i])
  }
  for (i in 1:n.adj) {
    sqrt.c.adj[i] <- sqrt(c[adj[i]])
    eta.adj[i] <- eta[adj[i]]
  }
  # Precision of conditioned distribution
  # eta[i]
  prec.eta[1] <- (rho * sqrt(c[1]) * sum(sqrt.c.adj[index[1]:index[2]]) +
    1 - rho)
  for (i in 2:Nareas) {
    prec.eta[i] <- (rho * sqrt(c[i]) *
      sum(sqrt.c.adj[(index[i] + 1):index[i +
        1]]) + 1 - rho)
  }
  # Mean of conditioned distribution eta[i]

```

```

mean.eta[1] <- (rho * inprod2(sqrt.c.adj[index[1]:index[2]],
  eta.adj[index[1]:index[2]]))/(rho *
  sum(sqrt.c.adj[index[1]:index[2]]) +
  1 - rho)
for (i in 2:Nareas) {
  mean.eta[i] <- (rho * inprod2(sqrt.c.adj[(index[i] +
    1):index[i + 1]], eta.adj[(index[i] +
    1):index[i + 1]]))/(rho * sum(sqrt.c.adj[(index[i] +
    1):index[i + 1]]) + 1 - rho)
}
# Sum-to-zero restriction for spatial
# effects
ceros <- 0
ceros ~ dnorm(sum.eta, 10)
sum.eta <- sum(eta[])
# Other prior distributions
mu ~ dflat()
sd.eta ~ dunif(0, 5)
rho ~ dunif(0, 1)
}
# Object where the results for each
# disease will be saved
results.Leroux <- list()
# Run Leroux model for each disease
for (i in 1:15) {
  # ATTENTION: Specify spatial weights
  # For unitary weights:
  w <- rep(1, length(carto.wb$adj))
  # For adaptive weights:
  w <- c(results.AdaptiveLeroux[[i]]$mean$c)
  # Data
  data <- list(0 = apply(Obs[, sex, , causes[i]],
    2, sum), E = apply(Exp[, sex, , causes[i]],
    2, sum), Nareas = dim(Obs)[3], adj = carto.wb$adj,
    n.adj = length(carto.wb$adj), num = carto.wb$num,
    index = index, c = w)
  # Initial values
  initials <- function() {
    list(mu = rnorm(1, 0, 1), sd.eta = runif(1,
      0, 1), eta = rnorm(data$Nareas))
  }
  # Variables to retrieve
  param <- c("mu", "lambda", "sd.eta",
    "eta", "SMR", "rho")
  results.Leroux[[i]] <- pbugs(data = data,
    inits = initials, parameters.to.save = param,
    model = Leroux_model, n.iter = 1e+05,
    n.burnin = 30000, n.chains = 3, DIC = F)
}
# For the model with unitary weights
save(results.Leroux, file = "Results/results.Leroux.unitaryw.RData")
# For the model with adaptive weights
save(results.Leroux, file = "Results/results.Leroux.adaptw.RData")

```


2. Estimated spatial weights c_i with multivariate adaptive BYM and Leroux models for each census tract of Valencia according to all 15 diseases in the data set (Section 4.1, Figure 1 in paper)

```
library(RColorBrewer)
library(sp)
# Results of the multivariate adaptive
# BYM and Leroux models
load("Results/results.AdaptiveBYM.15diseases.RData")
load("Results/results.AdaptiveLeroux.15diseases.RData")
palette <- brewer.pal(7, "YlOrRd")[7:1]
intervals_c <- c(0, 0.3, 0.7, 1, 1.2, 1.4,
  1.6, 2.05)
par(oma = c(2, 0, 2, 0), mar = c(1, 0, 0,
  0), mfrow = c(1, 2), xpd = NA)
# Estimated spatial weights with the
# multivariate adaptive BYM model
# according to all 15 diseases in the
# data set
plot(carto, col = palette[cut(results.AdaptiveBYM$mean$c,
  intervals_c)], xlim = c(-0.4430475, -0.2739941),
  ylim = c(39.45547, 39.55039), main = "Adaptive BYM model")
# Estimated spatial weights with the
# multivariate adaptive Leroux model
# according to all 15 diseases in the
# data set
plot(carto, col = palette[cut(results.AdaptiveLeroux$mean$c,
  intervals_c)], xlim = c(-0.4430475, -0.2739941),
  ylim = c(39.45547, 39.55039), main = "Adaptive Leroux model")
legend(-0.4870006, 39.49428, levels(cut(results.AdaptiveBYM$mean$c,
  intervals_c)), title = " ", border = NULL,
  fill = palette, bty = "n")
```

3. Standardized Mortality Ratios for studied mortality causes in Valencia estimated with the BYM (upper row) and Leroux (lower row) models and with spatial weights matrices of either unitary weights (left) or using the values obtained from the multivariate analysis of 14 diseases (all mortality causes of study except the evaluated cause) (Section 1, supplementary material in paper)

```
# Results of the BYM model with unitary
# weights
load("Results/results.BYM.unitaryw.RData")
BYM <- results.BYM
# Results of the Leroux model with
# unitary weights
load("Results/results.Leroux.unitaryw.RData")
Leroux <- results.Leroux
# Results of the BYM model with adaptive
```

```

# weights
load("Results/results.BYM.adaptw.RData")
BYM_adapt <- results.BYM
# Results of the Leroux model with
# adaptive weights
load("Results/results.Leroux.adaptw.RData")
Leroux_adapt <- results.Leroux
# Studied mortality causes
causes_name <- c("AIDS", "Stomach cancer",
  "Colorectal cancer", "Lung cancer", "Prostate cancer",
  "Bladder cancer", "Hematological cancer",
  "Mellitus diabetes", "Dementia", "Ischemic heart disease",
  "Ictus", "COPD", "Liver cirrhosis", "Suicides",
  "Traffic accidents")
Palette.RR <- brewer.pal(7, "BrBG")[7:1]
par(mfrow = c(2, 2), xpd = TRUE)
for (i in 1:15) {
  par(mfrow = c(2, 2), xpd = TRUE)
  aux <- cut(BYM[[i]]$mean$SMR, c(-100,
    0.67, 0.8, 0.91, 1.1, 1.25, 1.5,
    100))
  plot(carto, col = Palette.RR[aux], main = paste0("BYM model"),
    xlim = c(-0.4430475, -0.2739941),
    ylim = c(39.45547, 39.55039), cex.main = 1.5)
  aux <- cut(BYM_adapt[[i]]$mean$SMR, c(-100,
    0.67, 0.8, 0.91, 1.1, 1.25, 1.5,
    100))
  plot(carto, col = Palette.RR[aux], main = paste0("Adaptive BYM model"),
    xlim = c(-0.4430475, -0.2739941),
    ylim = c(39.45547, 39.55039), cex.main = 1.5)
  aux <- cut(Leroux[[i]]$mean$SMR, c(-100,
    0.67, 0.8, 0.91, 1.1, 1.25, 1.5,
    100))
  plot(carto, col = Palette.RR[aux], main = paste0("Leroux model"),
    xlim = c(-0.4430475, -0.2739941),
    ylim = c(39.45547, 39.55039), cex.main = 1.5)
  aux <- cut(Leroux_adapt[[i]]$mean$SMR,
    c(-100, 0.67, 0.8, 0.91, 1.1, 1.25,
    1.5, 100))
  plot(carto, col = Palette.RR[aux], main = paste0("Adaptive Leroux model"),
    xlim = c(-0.4430475, -0.2739941),
    ylim = c(39.45547, 39.55039), cex.main = 1.5)
  par(xpd = NA)
  legend(-0.5253527, 39.61729, 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.RR,
    bty = "n")
  mtext(causes_name[i], side = 3, cex = 2,
    line = 0, outer = TRUE)
}

```

4. DIC for the BYM and Leroux models with adaptive and unweighed spatial weights matrices (Section 4.2, Table 1 in paper)

```
# Studied mortality causes
causes <- c(1:2, 5, 7, 9:12, 15:21)
# Observed mortality cases
Observados <- apply(Obs[, , , causes], c(2,
  3, 4), sum)
# Function for DICs calculation
CalculaDIC <- function(mu, 0, save = FALSE) {
  D <- apply(mu, 1, function(x) {
    -2 * sum(dpois(0, x, log = T))
  })
  Dmedia <- mean(D)
  mumedia <- apply(mu, 2, mean)
  DenMedia <- -2 * sum(dpois(0, mumedia,
    log = T))
  if (save == TRUE) {
    return(c(Dmedia, Dmedia - DenMedia,
      2 * Dmedia - DenMedia))
  }
  cat("D = ", Dmedia, "pD = ", Dmedia -
    DenMedia, "DIC = ", 2 * Dmedia -
    DenMedia, " \n")
}
# Objects where the DIC of the models for
# each disease will be saved
DIC_BYM <- c()
DIC_BYMadapt <- c()
DIC_Leroux <- c()
DIC_Lerouxadapt <- c()
for (j in 1:15) {
  # DIC BYM model with unitary weights
  DIC_BYM[j] <- CalculaDIC(mu = BYM[[j]]$sims.list$lambda,
    0 = Observados[sex, , j], save = TRUE)[3]
  # DIC BYM model with adaptive weights
  DIC_BYMadapt[j] <- CalculaDIC(mu = BYM_adapt[[j]]$sims.list$lambda,
    0 = Observados[sex, , j], save = TRUE)[3]
  # DIC Leroux model with unitary weights
  DIC_Leroux[j] <- CalculaDIC(mu = Leroux[[j]]$sims.list$lambda,
    0 = Observados[sex, , j], save = TRUE)[3]
  # DIC Leroux model with adaptive weights
  DIC_Lerouxadapt[j] <- CalculaDIC(mu = Leroux_adapt[[j]]$sims.list$lambda,
    0 = Observados[sex, , j], save = TRUE)[3]
}
kable(data.frame(causes_name, DIC_BYM, DIC_BYMadapt,
  DIC_Leroux, DIC_Lerouxadapt), digits = 2,
  col.names = c("Causes", "BYM model Adjacency",
    "BYM model Adaptive", "Leroux model Adjacency",
    "Leroux model Adaptive"))
```

5. CPO for the BYM and Leroux models with adaptive and unweighed spatial weights matrices (Section 4.2, Table 1 in paper)

```

# Objects where the likelihood of the
# models for each simulation, geographic
# unit and disease will be saved
likelihood_BYM <- array(NA, dim = c(1002,
  531, 15))
likelihood_BYMadapt <- array(NA, dim = c(1002,
  531, 15))
likelihood_Leroux <- array(NA, dim = c(1002,
  531, 15))
likelihood_Lerouxadapt <- array(NA, dim = c(1002,
  531, 15))

# Objects where the CPO of the models for
# each geographic unit and disease will
# be saved
CPO_BYM <- array(NA, dim = c(531, 15))
CPO_BYMadapt <- array(NA, dim = c(531, 15))
CPO_Leroux <- array(NA, dim = c(531, 15))
CPO_Lerouxadapt <- array(NA, dim = c(531,
  15))

# Likelihood of the models for each
# disease, geographic unit and MCMC
# simulation
for (i in 1:15) {
  for (j in 1:531) {
    for (k in 1:1002) {
      likelihood_BYM[k, j, i] <- dpois(Observados[sex,
        j, i], BYM[[i]]$sims.list$lambda[k,
        j])
      likelihood_BYMadapt[k, j, i] <- dpois(Observados[sex,
        j, i], BYM_adapt[[i]]$sims.list$lambda[k,
        j])
      likelihood_Leroux[k, j, i] <- dpois(Observados[sex,
        j, i], Leroux[[i]]$sims.list$lambda[k,
        j])
      likelihood_Lerouxadapt[k, j,
        i] <- dpois(Observados[sex,
        j, i], Leroux_adapt[[i]]$sims.list$lambda[k,
        j])
    }
  }
}

# CPO of the models for each disease and
# geographic unit
for (i in 1:15) {
  for (j in 1:531) {
    CPO_BYM[j, i] <- 1/(mean(1/likelihood_BYM[,
      j, i]))
    CPO_BYMadapt[j, i] <- 1/(mean(1/likelihood_BYMadapt[,
      j, i]))
    CPO_Leroux[j, i] <- 1/(mean(1/likelihood_Leroux[,

```

```

      j, i]))
    CPO_Lerouxadapt[j, i] <- 1/(mean(1/likelihood_Lerouxadapt[,
      j, i]))
  }
}
# Total CPO of the models
CPO_TOTAL_BYM <- apply(apply(CPO_BYM, 2,
  function(x) {
    log(x)
  }), 2, sum)
CPO_TOTAL_Leroux <- apply(apply(CPO_Leroux,
  2, function(x) {
    log(x)
  }), 2, sum)
CPO_TOTAL_BYMadapt <- apply(apply(CPO_BYMadapt,
  2, function(x) {
    log(x)
  }), 2, sum)
CPO_TOTAL_Lerouxadapt <- apply(apply(CPO_Lerouxadapt,
  2, function(x) {
    log(x)
  }), 2, sum)
kable(data.frame(causes_name, CPO_TOTAL_BYM,
  CPO_TOTAL_BYMadapt, CPO_TOTAL_Leroux,
  CPO_TOTAL_Lerouxadapt), col.names = c("Causes",
  "BYM model Adjacency", "BYM model Adaptive",
  "Leroux model Adjacency", "Leroux model Adaptive"),
  digits = 2)

```