

# Analisis Resultados

Luis Requeni Zamora

01/06/2023

## INTRODUCCIÓN

```
datos2 <- read.table(file = "ST_mel1_rep2_counts.tsv",
                     sep = '\t', header = TRUE)
rownames(datos2) <- datos2[, 1]
datos2 <- datos2[, -1]

x <- substr(colnames(datos2), 1, 3)
x <- as.numeric(gsub("x", "", gsub("X", "", x)))
y <- substr(colnames(datos2), 4, 6)
y <- as.numeric(gsub("x", "", gsub("X", "", y)))

coordenadas <- cbind(y, x)
spatialpoint <- SpatialPoints(coordenadas)
spatialpointdf <- SpatialPointsDataFrame(spatialpoint, datos)
```

## GENES ALTAMENTE VARIABLES

```
load("melanoma_spatialpolygon.rda")

coords <- coordinates(melanoma_spatialpolygon)
IDs <- row.names(as(melanoma_spatialpolygon, "data.frame"))
Vecinos <- tri2nb(coords, row.names = IDs)
Vecinos_List <- nb2WB(Vecinos)

Resultados <- data.frame(gen = NULL, p_valor = NULL)
for(i in 1:16148){
  print(i)
  a <- moran.test(as.numeric(datos2[i, ]), nb2listw(Vecinos))
  Resultados <- rbind(Resultados, c(i, a$p.value, a$estimate[1]))
}
colnames(Resultados) <- c("Gen", "P-Valor", "Indice")
save(Resultados, file = "ResultadosMoran_Polygons.rda")
```

```
load("ResultadosMoran_Polygons.rda")
ResultadosMoran_Polygons <- Resultados
```

```
ResultadosMoran_Polygons2 <- ResultadosMoran_Polygons[order(ResultadosMoran_Polygons$`P-Valor`), ]
DatosMoran_Polygons2 <- datos2[ResultadosMoran_Polygons2$Gen[1:16148], ]
topMoranPolygons <- rownames(DatosMoran_Polygons2)
```

```
ResultadosMoran_Polygons2$Gen <- topMoranPolygons
head(ResultadosMoran_Polygons2)
```

```
##              Gen      P-Valor    Indice
## 167      GAPDH ENSG00000111640 7.387894e-95 0.6360628
## 939     ATP1A1 ENSG00000163399 3.287378e-91 0.6231539
## 111    SERPINE2 ENSG00000135919 4.849672e-89 0.6162251
## 122       PMEL ENSG00000185664 3.753279e-87 0.6101906
## 810      RPS21 ENSG00000171858 8.883128e-85 0.6006930
## 6       RPL37A ENSG00000197756 1.429709e-84 0.5993723
```

```
tail(ResultadosMoran_Polygons2)
```

```
##              Gen      P-Valor    Indice
## 1519      DIS3L2 ENSG00000144535 0.996931 -0.086749197
## 16031 LLNLR-268E12.1 ENSG00000276445 0.999925 -0.005397913
## 16032       MUC20 ENSG00000176945 0.999925 -0.005397913
## 16033      CLTCL1 ENSG00000070371 0.999925 -0.005397913
## 16034       DDN ENSG00000181418 0.999925 -0.005397913
## 16035      ETV3L ENSG00000253831 0.999925 -0.005397913
```

```

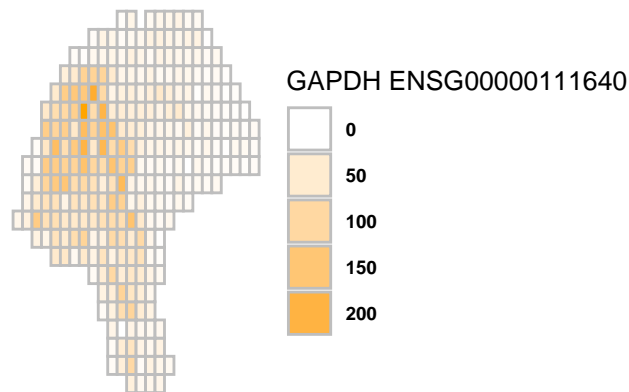
textcol <- "black"
colors <- c("red")
Valor1 <- t(datos2[topMoranPolygons[1], ])
Valor2 <- t(datos2[topMoranPolygons[2], ])
Valor3 <- t(datos2[topMoranPolygons[3], ])

Base1 <- data.frame(x, y, Valor1)
Base2 <- data.frame(x, y, Valor2)
Base3 <- data.frame(x, y, Valor3)

par(mfrow=c(1, 3))

ggplot(Base1, aes(x = y, y = x, fill = Valor1)) +
  geom_tile(colour = "grey", linewidth = 0.5) +
  guides(fill = guide_legend(title = topMoranPolygons[1])) +
  scale_fill_gradient2(low = "white", mid = "white", high = "orange", midpoint = .02) +
  theme_void(base_size = 10) +
  theme(legend.position = "right", legend.direction = "vertical",
        legend.title = element_text(colour = textcol),
        legend.text = element_text(colour = textcol, size = 7, face = "bold"),
        plot.background = element_blank(),
        panel.border = element_blank(),
        plot.margin = margin(0.7, 0.4, 0.1, 0.2, "cm"),
        plot.title = element_text(colour = textcol, hjust = 0, size = 14, face = "bold"))

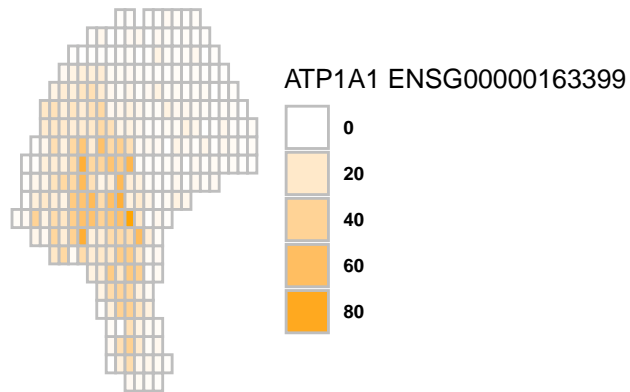
```



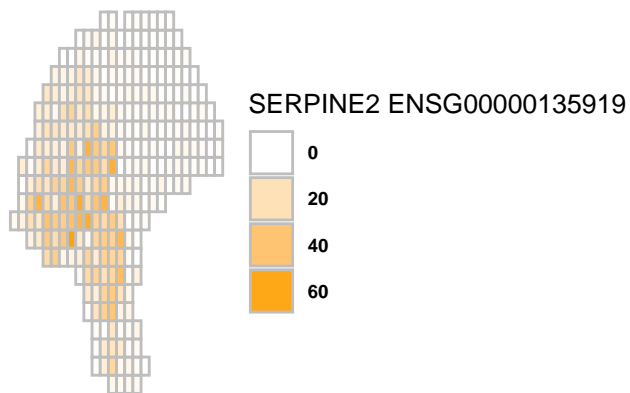
```

ggplot(Base2, aes(x = y, y = x, fill = Valor2)) +
  geom_tile(colour = "grey", linewidth = 0.5) +
  guides(fill = guide_legend(title = topMoranPolygons[2])) +
  scale_fill_gradient2(low = "white", mid = "white", high = "orange", midpoint = .02) +
  theme_void(base_size = 10) +
  theme(legend.position = "right", legend.direction = "vertical",
        legend.title = element_text(colour = textcol),
        legend.text = element_text(colour = textcol, size = 7, face = "bold"),
        plot.background = element_blank(),
        panel.border = element_blank(),
        plot.margin = margin(0.7, 0.4, 0.1, 0.2, "cm"),
        plot.title = element_text(colour = textcol, hjust = 0, size = 14, face = "bold"))

```



```
ggplot(Base3, aes(x = y, y = x, fill = Valor3)) +
  geom_tile(colour = "grey", linewidth = 0.5) +
  guides(fill = guide_legend(title = topMoranPolygons[3])) +
  scale_fill_gradient2(low = "white", mid = "white", high = "orange", midpoint = .02) +
  theme_void(base_size = 10) +
  theme(legend.position = "right", legend.direction = "vertical",
        legend.title = element_text(colour = textcol),
        legend.text = element_text(colour = textcol, size = 7, face = "bold"),
        plot.background = element_blank(),
        panel.border = element_blank(),
        plot.margin = margin(0.7, 0.4, 0.1, 0.2, "cm"),
        plot.title = element_text(colour = textcol, hjust = 0, size = 14, face = "bold"))
```



## CALCULO ACP

```
sce <- SingleCellExperiment(as.matrix(datos2))
sce@assays@data@listData$counts <- as.matrix(unlist(sce@assays@data@listData),
                                                nrow = 16148)
sce@assays@data@listData[["counts"]] <- sce@assays@data@listData[[1]]

sce <- logNormCounts(sce)
dec <- modelGeneVar(sce, assay.type = "logcounts")
top <- getTopHVGs(dec, n = 2000)
datosPCA <- runPCA(sce, ncomponents = 7, exprs_values = "logcounts",
                  subset_row = top)
datosPCA <- as.data.frame(datosPCA@int_colData@listData[["reducedDims"]][["PCA"]])

head(as.data.frame(t(datos2))[, c(1, 16148)])
```

```
##          PSME2 ENSG00000100911 PGM5P2 ENSG00000277778
## X7x15          2              0
## X7x16          0              0
## X7x17          0              0
## X7x18          0              0
## X8x13          0              0
## X8x14          1              0
```

```
head(datosPCA)
```

```
##          PC1          PC2          PC3          PC4          PC5          PC6
## X7x15  1.9829379  1.4512358 -7.227435  2.1068583 -2.1042511 -0.3544526
## X7x16  1.5391045  0.5299336  1.203405  2.0464473 -1.8843623  4.0249427
## X7x17 -0.2316224  1.0587171  3.356286  0.5383973  1.5228346  0.6354561
## X7x18  4.0109392  2.2564052  2.351316 -1.2636699  0.5921051 -0.1325016
## X8x13 -1.1661865  0.1063665 -2.860336  2.5964039 -6.1877103 -1.1612941
## X8x14 -1.8583911 -1.7624881 -11.021961  5.6702354 -1.4486031  1.4965119
##          PC7
## X7x15  1.5864330
## X7x16  0.4114744
## X7x17  3.0506769
## X7x18  2.3653149
## X8x13  3.3465836
## X8x14  1.2718308
```

## VECINDADES

```
load("melanoma_spatialpolygon.rda")

# Vecinos Basados En Contigüedad
# Un unico punto
Vecinos1 <- poly2nb(melanoma_spatialpolygon, queen = T)
nbInfo1 <- nb2WB(Vecinos1)
# Mas de un punto
Vecinos2 <- poly2nb(melanoma_spatialpolygon, queen = F)
nbInfo2 <- nb2WB(Vecinos2)

# Vecinos Basados En Grafos
coords <- coordinates(melanoma_spatialpolygon)
IDs <- row.names(as(melanoma_spatialpolygon, "data.frame"))
# Triangulación de Delaunay
Vecinos3 <- tri2nb(coords, row.names = IDs)
nbInfo3 <- nb2WB(Vecinos3)
# Esfera de Influencias
Vecinos4 <- graph2nb(soi.graph(Vecinos3, coords), row.names = IDs)
nbInfo4 <- nb2WB(Vecinos4)
# Vecinos de Gabriel
Vecinos5 <- graph2nb(gabrielneigh(coords, nnmult = 4), row.names = IDs, sym = T)
nbInfo5 <- nb2WB(Vecinos5)
# Vecinos Relativos
Vecinos6 <- graph2nb(relativeneigh(coords, nnmult = 4), row.names = IDs, sym = T)
nbInfo6 <- nb2WB(Vecinos6)

# Vecinos Basados En Distancias
# Vecino más próximo
Vecinos7 <- knn2nb(knearneigh(coords, k = 1), sym = T)
nbInfo7 <- nb2WB(Vecinos7)
# 2 Vecinos más próximos
Vecinos8 <- knn2nb(knearneigh(coords, k = 2), sym = T)
nbInfo8 <- nb2WB(Vecinos8)
# 4 Vecinos más próximos
Vecinos9 <- knn2nb(knearneigh(coords, k = 4), sym = T)
nbInfo9 <- nb2WB(Vecinos9)
```

```

y <- datosPCA
Modelo_Ejercicio <- nimbleCode(
{
  for(k in 1:4){
    sigma[k] ~ dgamma(1, 1)
    tautau[k] <- 1 / sigma[k]^2
  }
  for(k in 1:4){
    s[1:293, k] ~ dcar_normal(adj[1:L], weights[1:L], num[1:293], tautau[k], zero_mean = 1)
  }

  for(i in 1:293)
  {
    y[i, 1:7] ~ dmnorm(mus[1:7, z[i]], wi_tau[1:7, 1:7, i])
    wi_tau[1:7, 1:7, i] <- w[i]*tau[1:7, 1:7]
    w[i] ~ dgamma(2, 2)
    z[i] ~ dcat(omega[1:4, i])
    for(k in 1:4){
      alpha[i, k] ~ dnorm(0, 100)
      omega[k, i] <- Phi[k, i]/sum(Phi[1:4, i])
      log(Phi[k, i]) <- alpha[i, k] + s[i, k]
    }
  }
  for (h in 1:4)
  {
    mus[1:7, h] ~ dmnorm(mu_0[1:7], tau_0[1:7, 1:7])
  }
  tau[1:7, 1:7] ~ dwish(wish_V[1:7, 1:7], 8)
}
)

Iniciales_Ejercicio <- function()
{
  list(tau = diag(1,7),
       z = sample(c(1, 2, 3, 4), 293, replace = T),
       mus = matrix(rep(0, 7*4), ncol = 4),
       omega = matrix(rep(c(1/4, 1/4, 1/4, 1/4), 293), ncol = 293),
       w = rep(1, 293),
       alpha = matrix(rep(log(0.25), 293*4), nrow = 293),
       sigma = rep(50, 4),
       s = matrix(rep(0, 293*4), nrow = 293))
}

Parametros_Ejercicio <- c("mus", "z", "tau", "w", "s", "sigma",
                          "tautau", "omega", "Phi", "alpha")

nimbleOptions(showCompilerOutput = F)

```

```

ModeloNimbleVecindad1 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo1$adj),
    adj = nbInfo1$adj,
    weights = nbInfo1$weights,
    num = nbInfo1$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)

ModeloNimbleVecindad2 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo2$adj),
    adj = nbInfo2$adj,
    weights = nbInfo2$weights,
    num = nbInfo2$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)

ModeloNimbleVecindad3 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo3$adj),
    adj = nbInfo3$adj,
    weights = nbInfo3$weights,
    num = nbInfo3$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)

ModeloNimbleVecindad4 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo4$adj),
    adj = nbInfo4$adj,
    weights = nbInfo4$weights,
    num = nbInfo4$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)

```



```

ModeloNimbleVecindad5 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo5$adj),
    adj = nbInfo5$adj,
    weights = nbInfo5$weights,
    num = nbInfo5$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)

ModeloNimbleVecindad6 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo6$adj),
    adj = nbInfo6$adj,
    weights = nbInfo6$weights,
    num = nbInfo6$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)

ModeloNimbleVecindad7 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo7$adj),
    adj = nbInfo7$adj,
    weights = nbInfo7$weights,
    num = nbInfo7$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)

ModeloNimbleVecindad8 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo8$adj),
    adj = nbInfo8$adj,
    weights = nbInfo8$weights,
    num = nbInfo8$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)

ModeloNimbleVecindad9 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),

```

```

constants = list(mu_0 = rep(0, 7),
                 wish_V = diag(1, 7),
                 tau_0 = diag(0.01, 7),
                 L = length(nbInfo9$adj),
                 adj = nbInfo9$adj,
                 weights = nbInfo9$weights,
                 num = nbInfo9$num),
inits = Iniciales_Ejercicio,
monitors = Parametros_Ejercicio,
thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
summary = TRUE, WAIC = TRUE)

```

```
ModeloNimbleVecindad1$WAIC$WAIC
```

```
## [1] 9182.631
```

```
ModeloNimbleVecindad2$WAIC$WAIC
```

```
## [1] 9180.646
```

```
ModeloNimbleVecindad3$WAIC$WAIC
```

```
## [1] 9180.18
```

```
ModeloNimbleVecindad4$WAIC$WAIC
```

```
## [1] 9182.11
```

```
ModeloNimbleVecindad5$WAIC$WAIC
```

```
## [1] 9182.574
```

```
ModeloNimbleVecindad6$WAIC$WAIC
```

```
## [1] 9181.427
```

```
ModeloNimbleVecindad7$WAIC$WAIC
```

```
## [1] 9214.962
```

```
ModeloNimbleVecindad8$WAIC$WAIC
```

```
## [1] 9190.264
```

```
ModeloNimbleVecindad9$WAIC$WAIC
```

```
## [1] 9183.034
```

```
Vecinos10 <- nblag_cumul(nblag(Vecinos3, 2))
nbInfo10 <- nb2WB(Vecinos10)
Vecinos11 <- nblag_cumul(nblag(Vecinos3, 3))
nbInfo11 <- nb2WB(Vecinos11)
```

```
ModeloNimbleVecindadOrden2 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo10$adj),
    adj = nbInfo10$adj,
    weights = nbInfo10$weights,
    num = nbInfo10$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)
```

```
ModeloNimbleVecindadOrden3 <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
  constants = list(mu_0 = rep(0, 7),
    wish_V = diag(1, 7),
    tau_0 = diag(0.01, 7),
    L = length(nbInfo11$adj),
    adj = nbInfo11$adj,
    weights = nbInfo11$weights,
    num = nbInfo11$num),
  inits = Iniciales_Ejercicio,
  monitors = Parametros_Ejercicio,
  thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
  summary = TRUE, WAIC = TRUE)
```

```
ModeloNimbleVecindad3$WAIC$WAIC
```

```
## [1] 9180.18
```

```
ModeloNimbleVecindadOrden2$WAIC$WAIC
```

```
## [1] 9175.508
```

```
ModeloNimbleVecindadOrden3$WAIC$WAIC
```

```
## [1] 9177.537
```

## CLUSTERS

```
y <- datosPCA
Modelo_Ejercicio <- nimbleCode(
{
  for(k in 1:K){
    sigma[k] ~ dgamma(1, 1)
    tautau[k] <- 1 / sigma[k]^2
  }
  for(k in 1:K){
    s[1:293, k] ~ dcar_normal(adj[1:L], weights[1:L], num[1:293], tautau[k], zero_mean = 1)
  }

  for(i in 1:293)
  {
    y[i, 1:7] ~ dmnorm(mus[1:7, z[i]], wi_tau[1:7, 1:7, i])
    wi_tau[1:7, 1:7, i] <- w[i]*tau[1:7, 1:7]
    w[i] ~ dgamma(2, 2)
    z[i] ~ dcat(omega[1:K, i])
    for(k in 1:K){
      alpha[i, k] ~ dnorm(0, 100)
      omega[k, i] <- Phi[k, i]/sum(Phi[1:K, i])
      log(Phi[k, i]) <- alpha[i, k] + s[i, k]
    }
  }
  for (h in 1:K)
  {
    mus[1:7, h] ~ dmnorm(mu_0[1:7], tau_0[1:7, 1:7])
  }
  tau[1:7, 1:7] ~ dwish(wish_V[1:7, 1:7], 8)
}
)

Iniciales_Ejercicio <- function()
{
  list(tau = diag(1,7),
       z = sample(c(1:K), 293, replace = T),
       mus = matrix(rep(0, 7*K), ncol = K),
       omega = matrix(rep(c(rep(1/K, K)), 293), ncol = 293),
       w = rep(1, 293),
       alpha = matrix(rep(log(1/K), 293*K), nrow = 293),
       sigma = rep(50, K),
       s = matrix(rep(0, 293*K), nrow = 293))
}

Parametros_Ejercicio <- c("mus", "z", "tau", "w", "s", "sigma",
                          "tautau", "omega", "Phi", "alpha")

nimbleOptions(showCompilerOutput = F)
```

```

for(K in 3:9){
  print(K)
  ModeloNimbleVecindad <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
                                     constants = list(mu_0 = rep(0, 7),
                                                       wish_V = diag(1, 7),
                                                       tau_0 = diag(0.01, 7),
                                                       L = length(nbInfo10$adj),
                                                       adj = nbInfo10$adj,
                                                       weights = nbInfo10$weights,
                                                       K = K,
                                                       num = nbInfo10$num),
                                     inits = Iniciales_Ejercicio,
                                     monitors = Parametros_Ejercicio,
                                     thin = 10, niter = 20000, nburnin = 10000, nchains = 1,
                                     summary = TRUE, WAIC = TRUE)
  assign(paste0("ModeloNimbleCluster", K), ModeloNimbleVecindad)
}

```

```
ModeloNimbleCluster3$WAIC$WAIC
```

```
## [1] 9400.195
```

```
ModeloNimbleCluster4$WAIC$WAIC
```

```
## [1] 9180.563
```

```
ModeloNimbleCluster5$WAIC$WAIC
```

```
## [1] 9033.013
```

```
ModeloNimbleCluster6$WAIC$WAIC
```

```
## [1] 8927.074
```

```
ModeloNimbleCluster7$WAIC$WAIC
```

```
## [1] 8827.721
```

```
ModeloNimbleCluster8$WAIC$WAIC
```

```
## [1] 8787.857
```

```
ModeloNimbleCluster9$WAIC$WAIC
```

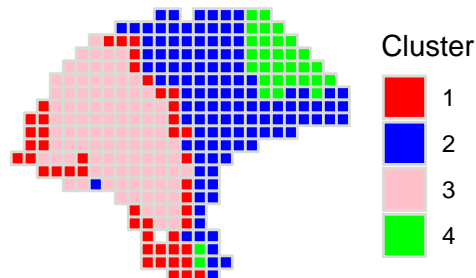
```
## [1] 8676.707
```

## MODELO FINAL

```
K = 4
ModeloNimbleFinal <- nimbleMCMC(Modelo_Ejercicio, data = list(y = y),
                                constants = list(mu_0 = rep(0, 7),
                                                  wish_V = diag(1, 7),
                                                  tau_0 = diag(0.01, 7),
                                                  L = length(nbInfo10$adj),
                                                  adj = nbInfo10$adj,
                                                  weights = nbInfo10$weights,
                                                  K = K,
                                                  num = nbInfo10$num),
                                inits = Iniciales_Ejercicio,
                                monitors = Parametros_Ejercicio,
                                thin = 10, niter = 100000, nburnin = 50000, nchains = 1,
                                summary = TRUE, WAIC = TRUE)
}
```

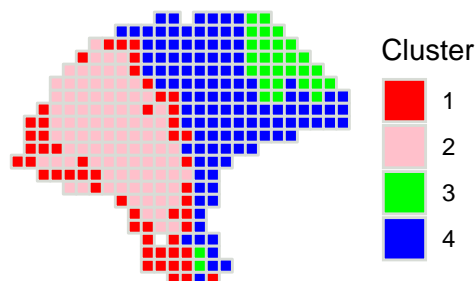
- Clasificación Original

```
load("ClasificacionMelanomaZhao.rda")
clusterPlot(ClasificacionMelanomaZhao, palette = c("red", "blue", "pink", "green"))
```



- Clasificación Modelo

```
zs <- ModeloNimbleFinal$samples[, grep("z", colnames(ModeloNimbleFinal$samples))]  
clasif <- apply(zs, 2, function(j) as.numeric(names(sort(table(j), decreasing = T))[1]))  
ClasificacionMelanomaZhao$spatial.cluster <- clasif  
clusterPlot(ClasificacionMelanomaZhao, palette = c("red", "pink", "green", "blue"))
```



```
load("ClasificacionMelanomaZhao.rda")
BayesSpace <- ClasificacionMelanomaZhao$spatial.cluster
table(BayesSpace)

zs <- ModeloNimbleFinal$samples[, grep("z", colnames(ModeloNimbleFinal$samples))]
Nosotros <- apply(zs, 2, function(j) as.numeric(names(sort(table(j), decreasing = T))[1]))
table(Nosotros)
```

```
Nosotros[Nosotros == 2] <- 30
Nosotros[Nosotros == 3] <- 40
Nosotros[Nosotros == 4] <- 20
Nosotros[Nosotros == 30] <- 3
Nosotros[Nosotros == 40] <- 4
Nosotros[Nosotros == 20] <- 2
```

```
table(BayesSpace)
```

```
## BayesSpace
##   1   2   3   4
## 47 117 98 31
```

```
table(Nosotros)
```

```
## Nosotros
##   1   2   3   4
## 54 117 92 30
```

```
names(which(BayesSpace != Nosotros))
```

```
## [1] "z[3]" "z[4]" "z[32]" "z[35]" "z[45]" "z[59]" "z[88]" "z[158]"
## [9] "z[215]" "z[239]"
```

```
zs <- ModeloNimbleFinal$samples[, names(which(BayesSpace != Nosotros))]
resultados <- apply(zs, 2, function(j) table(j))
resultados
```

```
## $'z[3] '
## j
##   1   2   3   4
## 2459 30 10 2501
##
## $'z[4] '
## j
##   1   2   3   4
## 3345 728 3 924
##
## $'z[32] '
## j
##   1   2
## 3630 1370
```

```
##
## $'z[35]'
## j
##      1      2      4
## 4814  183      3
##
## $'z[45]'
## j
##      1      2      3      4
## 2544    64      2 2390
##
## $'z[59]'
## j
##      1      2
## 2880 2120
##
## $'z[88]'
## j
##      1      2      4
## 2924 2075      1
##
## $'z[158]'
## j
##      1      2      3      4
## 4029  957      8      6
##
## $'z[215]'
## j
##      1      2      3      4
##      27      3 1774 3196
##
## $'z[239]'
## j
##      1      2      4
## 4475  513      12
```

```
datos2 <- read.table(file = "ST_mel1_rep2_counts.tsv",
                     sep = '\t', header = TRUE)
rownames(datos2) <- datos2[, 1]
datos2 <- datos2[, -1]

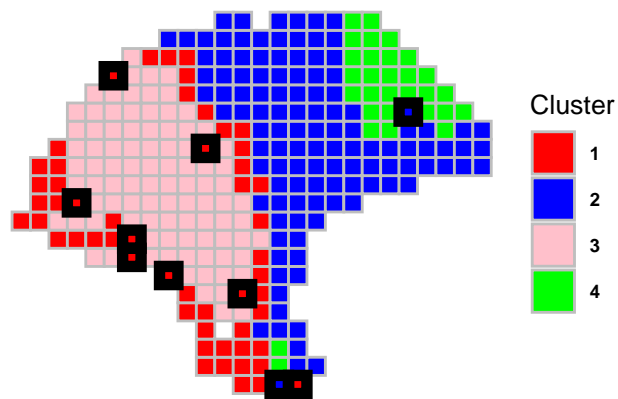
x <- substr(colnames(datos2), 1, 3)
x <- as.numeric(gsub("x", "", gsub("X", "", x)))
y <- substr(colnames(datos2), 4, 6)
y <- as.numeric(gsub("x", "", gsub("X", "", y)))

textcol <- "black"
colors <- c("red")
Base <- data.frame(x, y, Nosotros)

a <- names(which(BayesSpace != Nosotros))
a <- as.numeric(substr(a, 3, nchar(a)-1))
frames <- data.frame(Var1 = y[a], Var2 = x[a])
```

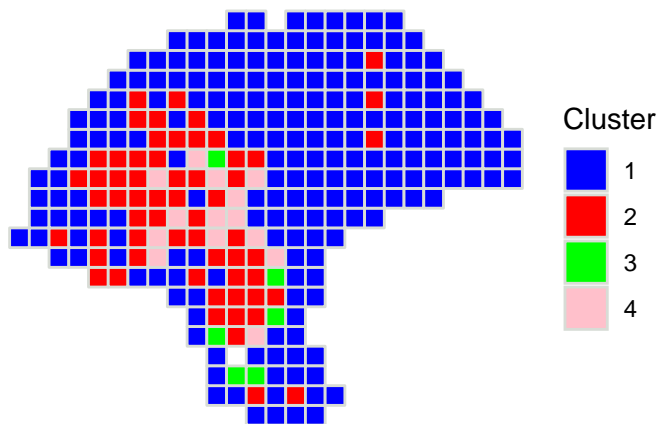


```
ggplot(Base, aes(x = y, y = x, fill = as.factor(Nosotros))) +
  geom_tile(colour = "grey", linewidth = 0.5) +
  guides(fill = guide_legend(title = "Cluster")) +
  scale_fill_manual(values = c("red", "blue", "pink", "green")) +
  geom_rect(data = frames, aes(x = Var1, y = Var2), linewidth = 2, fill = NA, colour = "black",
    xmin = frames$Var1 - 0.5, xmax = frames$Var1 + 0.5, ymin = frames$Var2 - 0.5, ymax = f
  ) +
  theme_void(base_size = 10) +
  theme(legend.position = "right", legend.direction = "vertical",
    legend.title = element_text(colour = textcol),
    legend.text = element_text(colour = textcol, size = 7, face = "bold"),
    plot.background = element_blank(),
    panel.border = element_blank(),
    plot.margin = margin(0.7, 0.4, 0.1, 0.2, "cm"),
    plot.title = element_text(colour = textcol, hjust = 0, size = 14, face = "bold"))
```



```
# HCLUST
HCLUST1 <- hclust(dist(t(datos2)))
HCLUST1 <- cutree(HCLUST1, k = 4)

ClasificacionMelanomaZhao$spatial.cluster <- HCLUST1
clusterPlot(ClasificacionMelanomaZhao, palette = c("blue", "red", "green", "pink"))
```

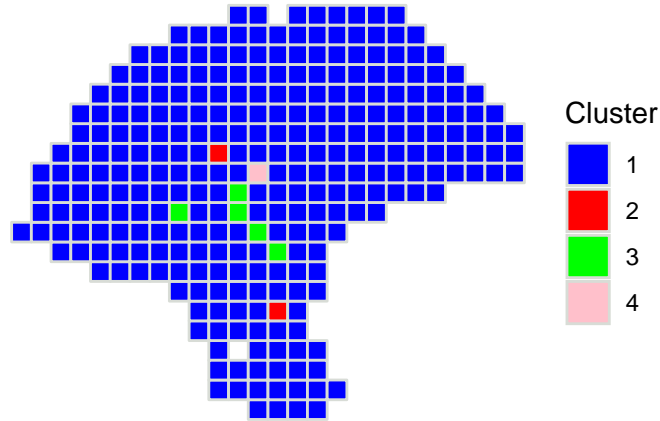


```
# AGNES
AGNES1 <- agnes(t(datos2))
```

```
AGNES1 <- cutree(AGNES1, k = 4)
```

```
ClasificacionMelanomaZhao$spatial.cluster <- AGNES1
```

```
clusterPlot(ClasificacionMelanomaZhao, palette = c("blue", "red", "green", "pink"))
```



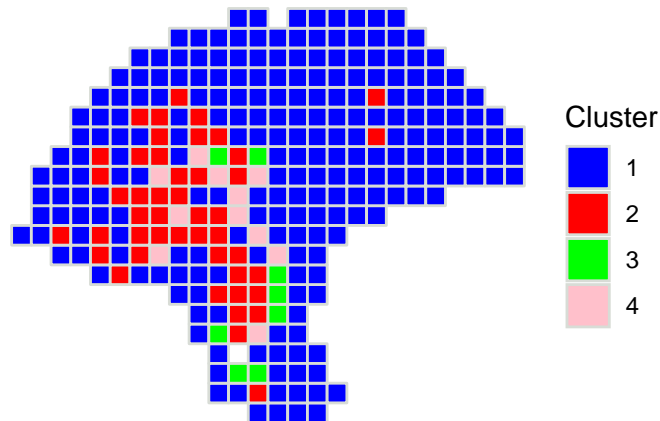
```
# DIANA
```

```
DIANA1 <- diana(t(datos2))
```

```
DIANA1 <- cutree(DIANA1, k = 4)
```

```
ClasificacionMelanomaZhao$spatial.cluster <- DIANA1
```

```
clusterPlot(ClasificacionMelanomaZhao, palette = c("blue", "red", "green", "pink"))
```

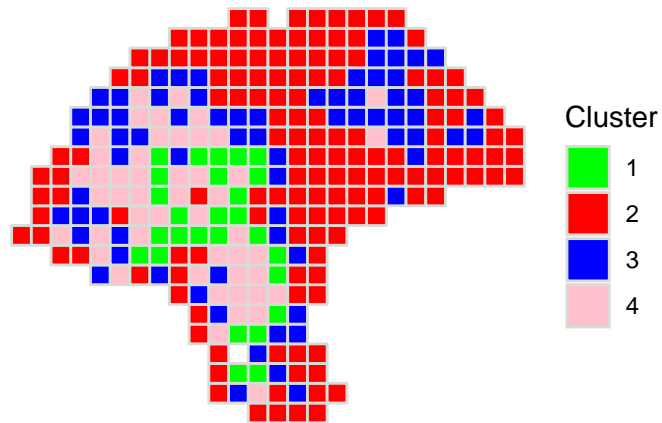


```
# KMEANS
```

```
KMEANS1 <- kmeans(t(datos2), centers = 4)
```

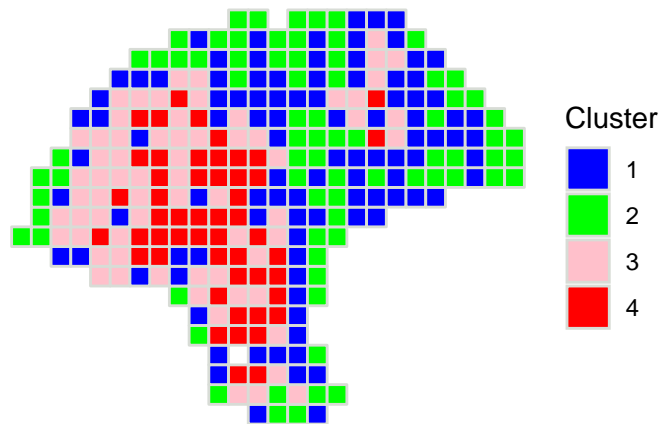
```
ClasificacionMelanomaZhao$spatial.cluster <- KMEANS1$cluster
```

```
clusterPlot(ClasificacionMelanomaZhao, palette = c("green", "red", "blue", "pink"))
```



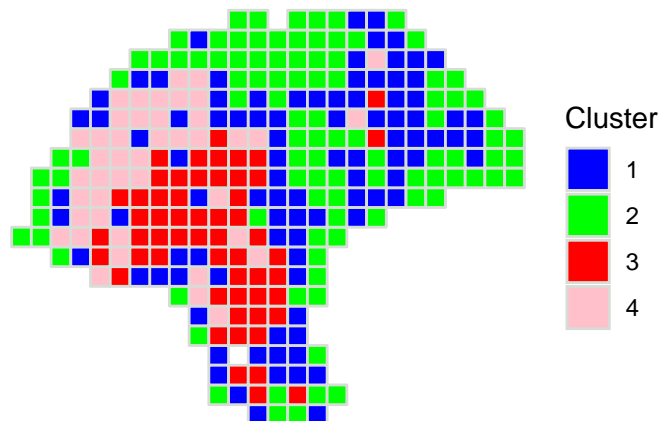
```
# PAM
PAM1 <- pam(t(datos2), k = 4)

ClasificacionMelanomaZhao$spatial.cluster <- PAM1$cluster
clusterPlot(ClasificacionMelanomaZhao, palette = c("blue", "green", "pink", "red"))
```



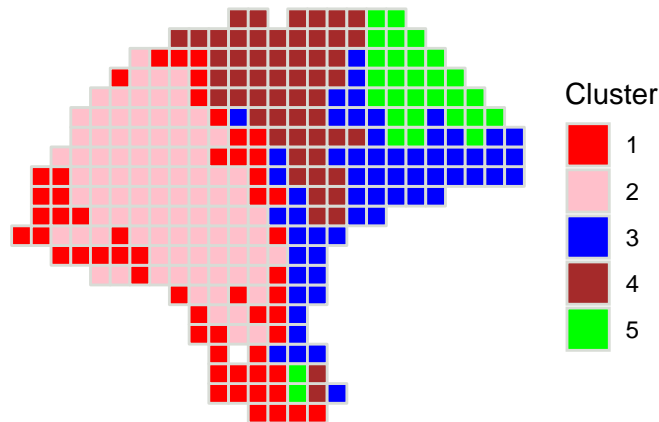
```
# CLARA
CLARA1 <- clara(t(datos2), k = 4)

ClasificacionMelanomaZhao$spatial.cluster <- CLARA1$cluster
clusterPlot(ClasificacionMelanomaZhao, palette = c("blue", "green", "red", "pink"))
```

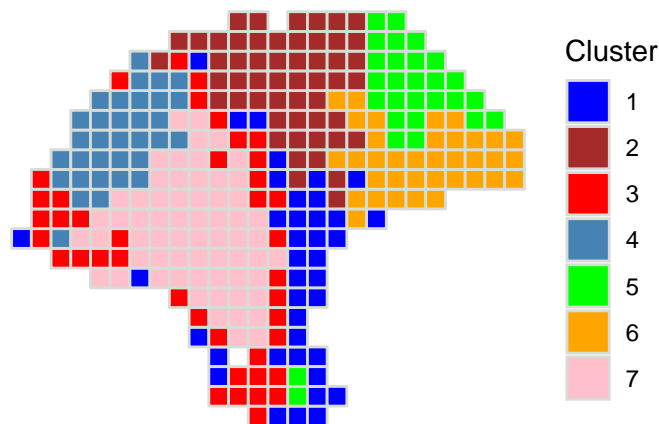


```
load("ModeloNimbleCluster5.rda")
load("ModeloNimbleCluster7.rda")
load("ModeloNimbleCluster9.rda")
```

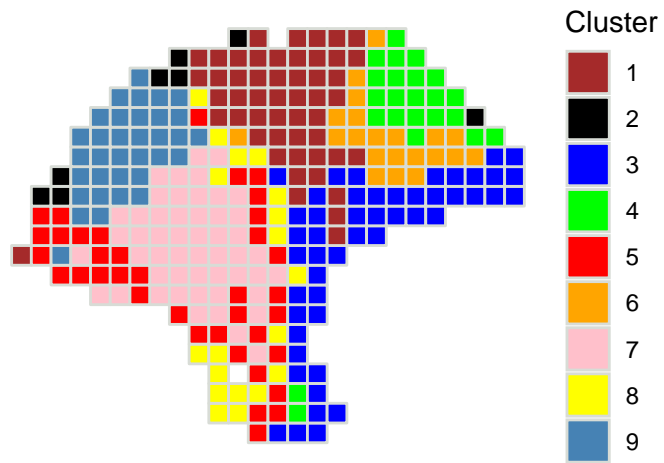
```
zs <- ModeloNimbleCluster5$samples[, grep("z", colnames(ModeloNimbleCluster5$samples))]
clasif <- apply(zs, 2, function(j) as.numeric(names(sort(table(j), decreasing = T))[1]))
ClasificacionMelanomaZhao$spatial.cluster <- clasif
clusterPlot(ClasificacionMelanomaZhao, palette = c("red", "pink", "blue", "brown", "green"))
```



```
zs <- ModeloNimbleCluster7$samples[, grep("z", colnames(ModeloNimbleCluster7$samples))]
clasif <- apply(zs, 2, function(j) as.numeric(names(sort(table(j), decreasing = T))[1]))
ClasificacionMelanomaZhao$spatial.cluster <- clasif
clusterPlot(ClasificacionMelanomaZhao, palette = c("blue", "brown", "red", "steelblue",
"green", "orange", "pink"))
```



```
zs <- ModeloNimbleCluster9$samples[, grep("z", colnames(ModeloNimbleCluster9$samples))]
clasif <- apply(zs, 2, function(j) as.numeric(names(sort(table(j), decreasing = T))[1]))
ClasificacionMelanomaZhao$spatial.cluster <- clasif
clusterPlot(ClasificacionMelanomaZhao, palette = c("brown", "black", "blue", "green", "red",
"orange", "pink", "yellow", "steelblue"))
```



## OTRAS FUNCIONALIDADES

```
Clasificaciones <- as.data.frame(ModeloNimbleFinal$samples[,
                                grep("z", colnames(ModeloNimbleFinal$samples))])

Probabilidades <- matrix(nrow = 293, ncol = 293)
for(i in 1:293){
  for(j in 1:293){
    Probabilidades[i, j] <- sum(Clasificaciones[, i] == Clasificaciones[, j])/5000
  }
}
Valor <- c()
for(i in 1:293){
  Valor <- c(Valor, mean(Probabilidades[i, Vecinos3[[i]]]))
}
```

