

Case Study 1: Clustering the epileptic.qol Dataset

Bayesian consensus clustering using the BCClong package

```
library(joineRML)
library(ggplot2)
library(cowplot)
library(survminer)
library(survival)
data(epileptic.qol) # import data from joineRML library (use ?epileptic.qol to
epileptic.qol$time_month <- epileptic.qol$time/30.25 # convert days to months
epileptic.qol <- epileptic.qol[order(epileptic.qol$id,
                                     epileptic.qol$time_month),] # Sort by ID and time

# create a new ID variable with values from 1 to N;
subj <- unique(epileptic.qol$id)
N <- length(subj)
epileptic.qol$anxiety_scale <- scale(epileptic.qol$anxiety)
epileptic.qol$depress_scale <- scale(epileptic.qol$depress)
epileptic.qol$aep_scale <- scale(epileptic.qol$aep)
epileptic.qol$timec <- epileptic.qol$time_month -
  mean(epileptic.qol$time_month) # center the time variables
```

Bayesian consensus clustering (BCClong package)

```
library(BCClong)

## Warning: package 'BCClong' was built under R version 4.2.2

#-----#
# computed the mean adjusted adherence to determine the number of clusters
#set.seed(2023)
#alpha.adjust <- NULL
#for (k in 2:8){
#  fit.BCC <- BCC.multi (
#    mydat = list( epileptic.qol$anxiety_scale,
#                  epileptic.qol$depress_scale,
#                  epileptic.qol$aep_scale),
#    dist = c("gaussian","gaussian","gaussian"),
#    id = list(epileptic.qol$id,epileptic.qol$id,epileptic.qol$id),
#    time = list(epileptic.qol$timec,epileptic.qol$timec,epileptic.qol$timec),
#    formula =list(y ~ time + (1 + time/id),
#                  y ~ time + (1 + time/id),
#                  y ~ time + (1 + time/id)),
#    num.cluster = k,
#    initial.cluster.membership = "random", # "mixAK" or "random"
#    print.info="FALSE",
#    burn.in = 2000, # number of samples discarded
```

```

# thin = 10,                # thinning
# per = 100,                # output information every "per" iteration
# max.iter = 12000)         # maximum number of iteration
# alpha.adjust <- c(alpha.adjust, fit.BCC$alpha.adjust)
#}

# plot the mean adjusted adherence:
# the model has the largest mean adjusted adherence
# when the number of clusters is 2
# num.clust.BCC <- which.max(alpha.adjust) + 1
#-----#

num.clust.BCC <- 2 # optimal number of clusters based on mean adjusted adherence
# Fit the final model with the number of cluster 2 (largest mean adjusted adherence)
set.seed(222)
fit.BCC <- BCC.multi (
  mydat = list( epileptic.qol$anxiety_scale,
                epileptic.qol$depress_scale,
                epileptic.qol$aep_scale),
  dist = c("gaussian","gaussian","gaussian"),
  id = list(epileptic.qol$id,
            epileptic.qol$id,
            epileptic.qol$id),
  time = list(epileptic.qol$timec,epileptic.qol$timec,epileptic.qol$timec),
  formula =list(y ~ time + (1 + time|id),
                y ~ time + (1 + time|id),
                y ~ time + (1 + time|id)),
  num.cluster = num.clust.BCC,
  initial.cluster.membership = "mixAK", # "mixAK" or "random"
  print.info="FALSE",
  burn.in = 2000,                # number of samples discarded
  thin = 10,                    # thinning
  per = 1000,                   # output information every "per" iteration
  max.iter = 12000)             # maximum number of iteration

##
##
##
## iter = 1000
## iter = 2000
## iter = 3000
## iter = 4000
## iter = 5000
## iter = 6000
## iter = 7000
## iter = 8000
## iter = 9000
## iter = 10000
## iter = 11000
## iter = 12000

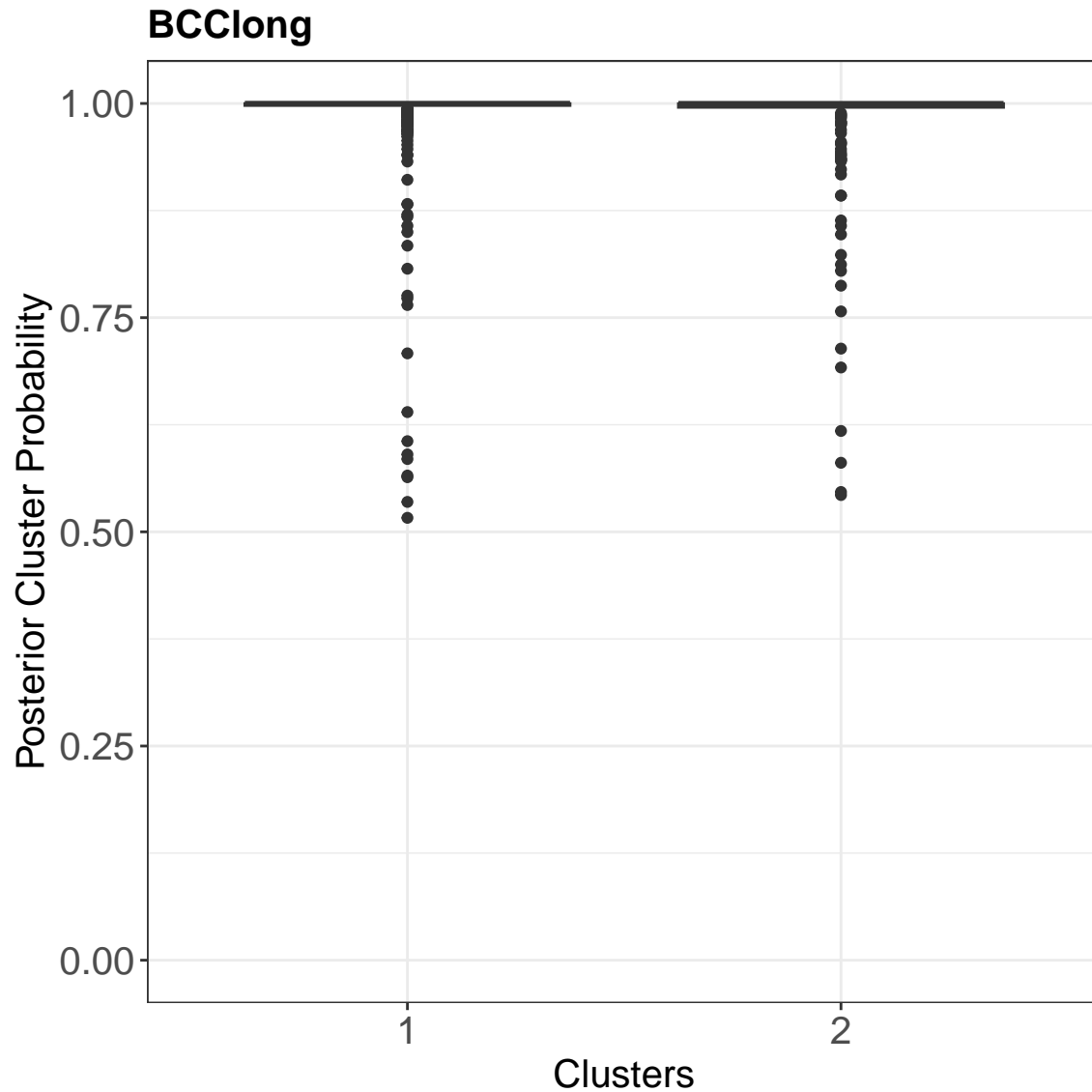
# str(fit.BCC)                # see the structure of the outputs
#-----#
# the id.org from object dat corresponds to
# the id values from the original dataset, which can be used

```

```

# as an index to merge the class membership back to the original dataset
dat <- fit.BCC$dat
dat1_uq <- dat[[1]][!duplicated(dat[[1]]$id.org, fromLast=TRUE),]
dat1_uq$postprob <- fit.BCC$postprob
# Posterior cluster probability
bp.BCClong <- ggplot(dat1_uq, aes(x=factor(cluster.global ), y=postprob)) +
  geom_boxplot() + ggtitle("BCClong") +
  xlab("Clusters") + ylab("Posterior Cluster Probability") +
  ylim(c(0,1)) +
  theme_bw() +
  theme(legend.position = "none",
        plot.title = element_text(size = 15, face = "bold"),
        axis.text=element_text(size=15),
        axis.title=element_text(size=15),
        axis.text.x = element_text(angle = 0 ),
        strip.text.x = element_text(size = 15, angle = 0),
        strip.text.y = element_text(size = 15,face="bold"))
bp.BCClong

```



```

dat.cluster <- data.frame(id=dat1_uq$id.org,cluster.global=dat1_uq$cluster.global)
per <- round(100*table(dat.cluster$cluster.global)/length(dat.cluster$cluster.global),1)

dat.cluster$cluster.global <- cluster.BCC <- factor(dat.cluster$cluster.global,
  labels =paste("Cluster ",1:num.clust.BCC," (",per,"%)",sep=""))
dnew <- merge(epileptic.qol,dat.cluster, by="id")
dnew$time_month <- dnew$time/30.25

#-----#
# plotting the first feature (anxiety) by clusters
library(ggplot2)
library(cowplot)
p1.BCC <- ggplot(data = dnew, aes(x =time_month, y =anxiety,
  color=cluster.global,
  linetype=cluster.global,
  fill=cluster.global))+
  geom_smooth(method = "loess", linewidth = 3,se = FALSE,span=2)+

```

```

theme_bw() +
ggtitle("BCClong") + xlab("Time (months)") + ylab("anxiety") +
theme(legend.position = "none", legend.title = element_blank(),
      plot.title = element_text(size = 15, face = "bold"),
      #legend.text = element_text(size = 15),
      axis.text = element_text(size = 15),
      axis.title = element_text(size = 15),
      axis.text.x = element_text(size = 15, angle = 0, hjust = 0.95, vjust = 0.2),
      strip.background = element_blank(),
      strip.text.x = element_blank(),
      strip.text.y = element_blank()) +
#scale_color_manual(values = c("red", "blue", "green")) +
guides(color = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       linetype = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       shape = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       fill = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE)) +
ylim(c(min(epileptic.qol$anxiety, na.rm = TRUE),
        max(epileptic.qol$anxiety, na.rm = TRUE))) +
scale_color_manual(values = c("green", "black")) +
scale_fill_manual(values = c("green", "black"))

#-----#
# plotting the second feature (depress) by clusters
p2.BCC <- ggplot(data = dnew, aes(x = time_month, y = depress,
                                color = cluster.global,
                                linetype = cluster.global,
                                fill = cluster.global)) +
geom_smooth(method = "loess", linewidth = 3, se = FALSE, span = 2) +
theme_bw() +
ggtitle("BCClong") + xlab("Time (months)") + ylab("depress") +
theme(legend.position = "none", legend.title = element_blank(),
      plot.title = element_text(size = 15, face = "bold"),
      #legend.text = element_text(size = 15),
      axis.text = element_text(size = 15),
      axis.title = element_text(size = 15),
      axis.text.x = element_text(size = 15, angle = 0, hjust = 0.95, vjust = 0.2),
      strip.background = element_blank(),
      strip.text.x = element_blank(),
      strip.text.y = element_blank()) +
guides(color = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       linetype = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       shape = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       fill = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE)) +
ylim(c(min(dnew$depress, na.rm = TRUE),
        max(dnew$depress, na.rm = TRUE))) +
scale_color_manual(values = c("green", "black")) +
scale_fill_manual(values = c("green", "black"))

#-----#
# plotting the third feature (aep) by clusters
p3.BCC <- ggplot(data = dnew, aes(x = time_month, y = aep,
                                color = cluster.global,
                                linetype = cluster.global,
                                fill = cluster.global)) +
geom_smooth(method = "loess", linewidth = 3, se = FALSE, span = 2) +

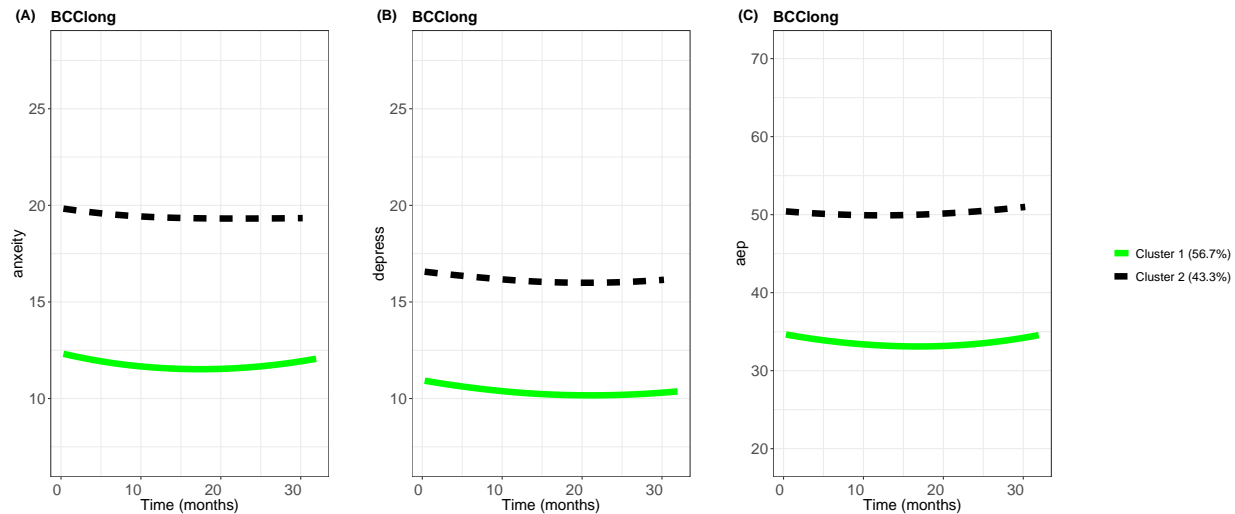
```

```

theme_bw() +
ggtitle("BCClong") + xlab("Time (months)") + ylab("aep") +
theme(legend.position = "none", legend.title = element_blank(),
      plot.title = element_text(size = 15, face = "bold"),
      #legend.text = element_text(size = 15),
      axis.text = element_text(size = 15),
      axis.title = element_text(size = 15),
      axis.text.x = element_text(size = 15, angle = 0, hjust = 0.95, vjust = 0.2),
      strip.background = element_blank(),
      strip.text.x = element_blank(),
      strip.text.y = element_blank()) +
guides(color = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       linetype = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       shape = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE),
       fill = guide_legend(nrow = 1, byrow = TRUE, reverse = FALSE)) +
ylim(c(min(dnew$aep, na.rm = TRUE),
        max(dnew$aep, na.rm = TRUE))) +
scale_color_manual(values = c("green", "black")) +
scale_fill_manual(values = c("green", "black"))

#-----#
# extract a legend
legend.BCC <- get_legend(ggplot(data = dnew, aes(x = time_month, y = depress,
          color = cluster.global,
          linetype = cluster.global,
          fill = cluster.global)) +
  geom_smooth(method = "loess", linewidth = 3, se = FALSE, span = 2) +
  theme_bw() +
  ggtitle("BCClong") + xlab("Time (months)") + ylab("depress") +
  theme(legend.position = c(0.5, 0.5),
        legend.text = element_text(size = 12),
        legend.title = element_blank(),
        plot.title = element_text(size = 15, face = "bold"),
        axis.text = element_text(size = 15),
        axis.title = element_text(size = 15),
        axis.text.x = element_text(size = 15, angle = 0, hjust = 0.95, vjust = 0.2),
        strip.background = element_blank(),
        strip.text.x = element_blank(),
        strip.text.y = element_blank()) +
  guides(color = guide_legend(nrow = 2, byrow = TRUE, reverse = FALSE),
         linetype = guide_legend(nrow = 2, byrow = TRUE, reverse = FALSE),
         shape = guide_legend(nrow = 2, byrow = TRUE, reverse = FALSE),
         fill = guide_legend(nrow = 2, byrow = TRUE, reverse = FALSE)) +
  ylim(c(min(dnew$depress, na.rm = TRUE), max(dnew$depress, na.rm = TRUE))) +
  scale_color_manual(values = c("green", "black")) +
  scale_fill_manual(values = c("green", "black"))
)
plot_grid(p1.BCC, NULL, p2.BCC, NULL, p3.BCC, NULL, legend.BCC,
  labels = c("(A)", "", "(B)", "", "(C)", "", ""),
  ncol = 7,
  rel_widths = c(1, 0.1, 1, 0.1, 1, 0.1, 0.5))

```



```
## Use Cox Model to Evaluate the Association between Clusters and Time to Treatment Failure
dnew_uq <- epileptic.qol[!duplicated(epileptic.qol$id, fromLast=TRUE),] # Keep last observation per id
dnew_uq$id.org <- dnew_uq$id
dnew_uq <- merge(dnew_uq, dat1_uq[,c("id.org", "cluster.global", "postprob")])
dnew_uq$with.time.month <- dnew_uq$with.time/30.25
fit <- survfit(Surv(with.time.month, with.status2) ~ cluster.global,
               data = dnew_uq)
res.cox <- coxph(Surv(with.time.month, with.status2) ~ cluster.global,
                 weights=postprob, data = dnew_uq)
pvalue <- ifelse(summary(res.cox)$sctest[3] >= 0.0001,
                 summary(res.cox)$sctest[3], '<0.0001')

# Visualize with survminer
# install.packages("survminer")
library(survminer)
library(survival)
names(fit$strata) <- paste("Cluster ", 1:num.clust.BCC, " (", per,"%", ")", sep="")
gp_survival.BCC.global <- ggsurvplot(fit, data = dnew_uq, title="BCClong",
                                     risk.table = TRUE, risk.table.y.text.col = TRUE, pval = pvalue,
                                     legend = "bottom", # conf.int = TRUE,
                                     xlab = "Time (months)",
                                     ggtheme = theme_bw() +
                                       theme(legend.position = "none", legend.title = element_blank(),
                                             plot.title = element_text(size = 15, face = "bold"),
                                             axis.text = element_text(size = 15),
                                             axis.title = element_text(size = 15),
                                             strip.text.x = element_text(size = 15),
                                             strip.text.y = element_text(size = 15)))
gp_survival.BCC.global$plot <- gp_survival.BCC.global$plot +
  guides(fill = guide_legend(title = NULL, nrow = 1),
         color = guide_legend(title = NULL, nrow = 1),
         linetype = guide_legend(title = NULL, nrow = 1)) +
  scale_color_manual(values = c("green", "black")) +
  scale_fill_manual(values = c("green", "black"))
gp_survival.BCC.global$plot
```

