Case Study 2: Clustering the PBC Dataset

Bayesian consensus clustering model using the BCClong package

```
# install.packages("mixAK")
library(mixAK)
```

```
## Warning: package 'lme4' was built under R version 4.2.2
```

```
data(PBCseq)
# patients known to be alive  and without liver transplantation at 910 days of follow-up
idx <- unique(PBCseq[PBCseq$alive>910,]$id);
dnew910 <- PBCseq[PBCseq$id %in% idx,];
dnew910_uq <- dnew910[!duplicated(dnew910$id, fromLast=TRUE),] # Keep last observation per ID

dnew910$time <-  dnew910$month
dnew910$time <-  dnew910$month - mean(dnew910$month,na.rm=TRUE)
dnew910$time2 <-  dnew910$time^2

# use only data before 910 days (2.5 years)
dnew910.before <- dnew910[dnew910$day<=910,]; length(unique(dnew910.before$id))
```

```
## [1] 260
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

```
dnew910.before$lbili_scale <- as.numeric(scale(dnew910.before$lbili))
dnew910.before$lalbumin_scale <- as.numeric(scale(dnew910.before$lalbumin))
dnew910.before$lalk.phos_scale <- as.numeric(scale(dnew910.before$lalk.phos))
dnew910.before$lsgot_scale <- as.numeric(scale(dnew910.before$lsgot))
dnew910.before$lplatelet_scale <- as.numeric(scale(dnew910.before$lplatelet))
```

Bayesian consensus clustering (BCClong package)

```
library(BCClong)
```

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

```
# not run to reduce compiling time
# determining the number of clusters
#set.seed(2024)
#alpha.adjust <- NULL
#for (k in 2:5){
#  fit.BCC <- BCC.multi (
#    mydat = list(dnew910.before$lbili_scale,dnew910.before$lalbumin_scale,
#       dnew910.before$lalk.phos_scale,dnew910.before$lsgot_scale,dnew910.before$lplatelet_scale),
#    dist = c("gaussian","gaussian","gaussian","gaussian","gaussian"),
#  id = list(dnew910.before$id,dnew910.before$id,dnew910.before$id,
#                  dnew910.before$id,dnew910.before$id),
```

```
#  time = list(dnew910.before$time,dnew910.before$time,
#           dnew910.before$time,dnew910.before$time,dnew910.before$time),
#  formula =list(y ~ time +  (1 + time|id),
#                y ~ time +  (1 + time|id),
#                y ~ time +  (1 + time|id),
#                y ~ time +  (1 + time|id),
#                y ~ time +  (1 + time|id)),
#    num.cluster = k,
#    initials= NULL,              # initial values for model parameters
#    initial.cluster.membership = "random", # "mixAK" or "random"
#    print.info="FALSE",
#    burn.in = 1000,              # number of samples discarded
#    thin = 1,                   # thinning
#    per = 1000,                      # output information every "per" iteration
#    max.iter = 2000)           # maximum number of iteration
#  alpha.adjust <- c(alpha.adjust, fit.BCC$alpha.adjust)
#}
#num.clust.BCC <- which.max(alpha.adjust) + 1
```

```
num.clust.BCC <-  2  # optimal number of mean adjusted adherence
# to speed up the convergence of MCMC, here we first fit a
# single-feature model (for each feature, therefore 5 models)
# to obtain a better initial values for the
# cluster membership to be used in the consensus clustering
#-------------------------------------------------------------------------------------------#
set.seed(32034)
fit.BCC1 <- BCC.multi (
   mydat = list(dnew910.before$lbili_scale),
   dist = c("gaussian"),
  id = list(dnew910.before$id),
  time = list(dnew910.before$time),
  formula =list(y ~ time +  (1 + time|id)),
  num.cluster = num.clust.BCC,
  initials= NULL,              # initial values for model parameters
  initial.cluster.membership = "random", # "mixAK" or "random"
  print.info="FALSE",
  burn.in = 1000,              # number of samples discarded
  thin = 1,                      # thinning
  per = 1000,                      # output information every "per" iteration
  max.iter = 2000)           # maximum number of iteration
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.133948 (tol = 0.002, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00481855 (tol = 0.002, component 1)
```

```
## iter = 1000
## iter = 2000
```

```
fit.BCC2 <- BCC.multi (
   mydat = list(dnew910.before$lalbumin_scale),
   dist = c("gaussian"),
  id = list(dnew910.before$id),
  time = list(dnew910.before$time),
```

```
  formula =list(y ~ time +  (1|id)),
  num.cluster = num.clust.BCC,
  initials= NULL,              # initial values for model parameters
  initial.cluster.membership = "random", # "mixAK" or "random"
  print.info="FALSE",
  burn.in = 1000,              # number of samples discarded
  thin = 1,                      # thinning
  per = 1000,                      # output information every "per" iteration
  max.iter = 2000)           # maximum number of iteration
```

```
## iter = 1000
## iter = 2000
```

```
fit.BCC3 <- BCC.multi (
   mydat = list(dnew910.before$lalk.phos_scale),
   dist = c("gaussian"),
  id = list(dnew910.before$id),
  time = list(dnew910.before$time),
  formula =list(y ~ time +  (1 + time|id)),
  num.cluster = num.clust.BCC,
  initials= NULL,              # initial values for model parameters
  initial.cluster.membership = "random", # "mixAK" or "random"
  print.info="FALSE",
  burn.in = 1000,              # number of samples discarded
  thin = 1,                      # thinning
  per = 1000,                      # output information every "per" iteration
  max.iter = 2000)           # maximum number of iteration
```

```
## iter = 1000
## iter = 2000
```

```
fit.BCC4 <- BCC.multi (
   mydat = list(dnew910.before$lsgot_scale),
   dist = c("gaussian"),
  id = list(dnew910.before$id),
  time = list(dnew910.before$time),
  formula =list(y ~ time +  (1 + time|id)),
  num.cluster = num.clust.BCC,
  initials= NULL,              # initial values for model parameters
  initial.cluster.membership = "random", # "mixAK" or "random"
  print.info="FALSE",
  burn.in = 1000,              # number of samples discarded
  thin = 1,                      # thinning
  per = 1000,                      # output information every "per" iteration
  max.iter = 2000)
```

```
## iter = 1000
## iter = 2000
```

```
fit.BCC5 <- BCC.multi (
   mydat = list(dnew910.before$lplatelet_scale),
   dist = c("gaussian"),
  id = list(dnew910.before$id),
  time = list(dnew910.before$time),
  formula =list(y ~ time +  (1 + time|id)),
  num.cluster = num.clust.BCC,
```

```
  initials= NULL,              # initial values for model parameters
  initial.cluster.membership = "random", # "mixAK" or "random"
  print.info="FALSE",
  burn.in = 1000,              # number of samples discarded
  thin = 1,                      # thinning
  per = 1000,                      # output information every "per" iteration
  max.iter = 2000)
```

```
## iter = 1000
## iter = 2000
```

```
# fit the final model based on the initial cluster membership obtained
# from the previous steps
set.seed(2023)
ptm <- proc.time()
fit.BCC <- BCC.multi (
   mydat = list(dnew910.before$lbili_scale,dnew910.before$lalbumin_scale,
                dnew910.before$lalk.phos_scale,dnew910.before$lsgot_scale,
                dnew910.before$lplatelet_scale),
   dist = c("gaussian","gaussian","gaussian","gaussian","gaussian"),
  id = list(dnew910.before$id,dnew910.before$id,
            dnew910.before$id,dnew910.before$id,dnew910.before$id),
  time = list(dnew910.before$time,dnew910.before$time,dnew910.before$time,
              dnew910.before$time,dnew910.before$time),
  formula =list(y ~ time +  (1 + time|id),
                y ~ time +  (1 + time|id),
                y ~ time +  (1 + time|id),
                y ~ time +  (1 + time|id),
                y ~ time +  (1 + time|id)),
  num.cluster = num.clust.BCC,
  initials= NULL,              # initial values for model parameters
  initial.cluster.membership = "input",
  input.initial.cluster.membership = list(fit.BCC1$cluster.global,
                                          fit.BCC2$cluster.global,fit.BCC3$cluster.global,
                                          fit.BCC4$cluster.global,fit.BCC5$cluster.global),
  initial.global.cluster.membership = fit.BCC1$cluster.global,
  print.info="FALSE",
  burn.in = 10000,             # number of samples discarded
  thin = 10,                     # thinning
  per = 10000,                     # output information every "per" iteration
  max.iter = 20000)              # maximum number of iteration
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0191714 (tol = 0.002, component 1)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00489004 (tol = 0.002, component 1)
```

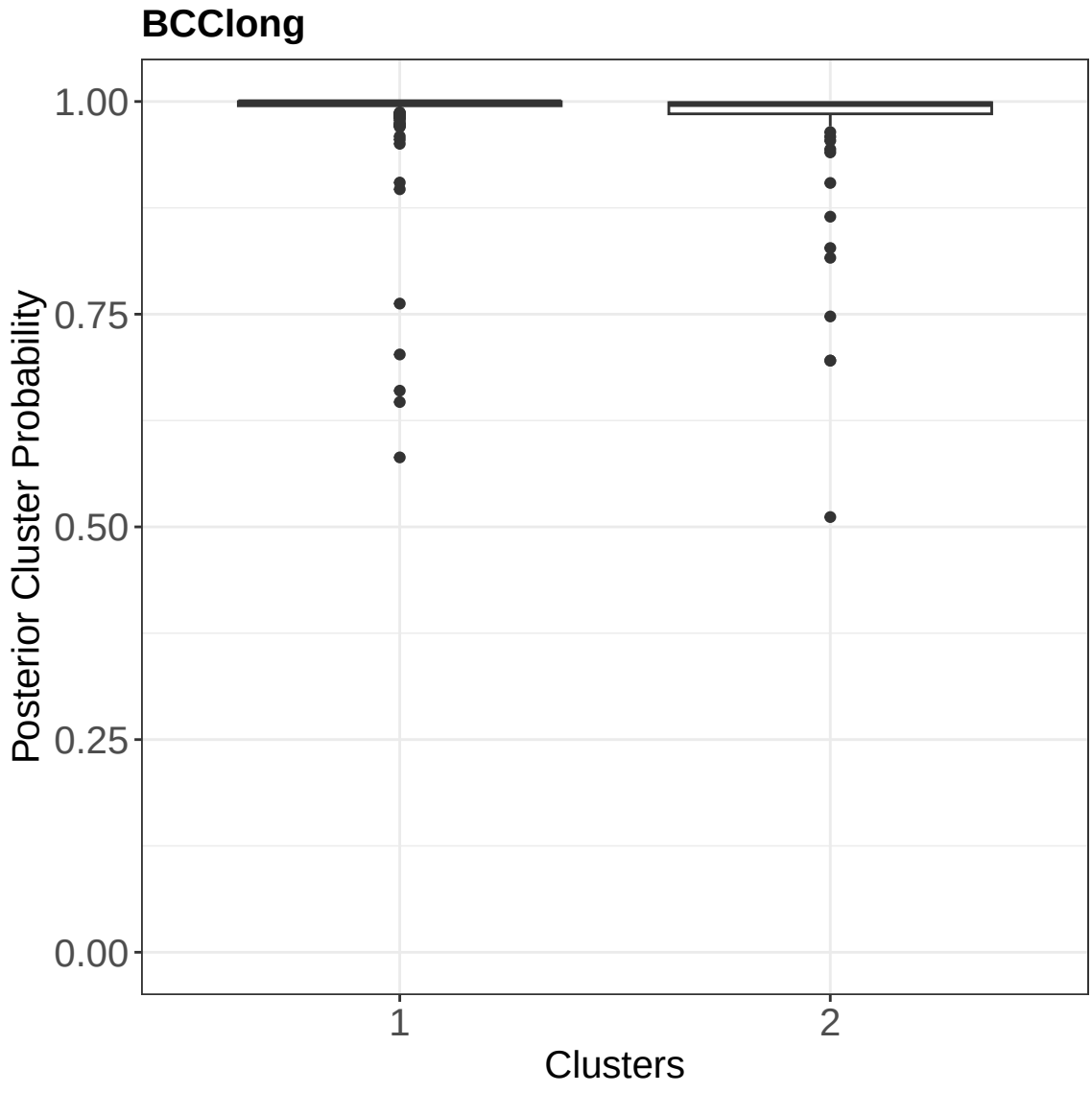
```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.00476882 (tol = 0.002, component 1)
```

```
## iter = 10000
## iter = 20000
```

```
dat <- fit.BCC$dat
dnew_uq <- dnew910.before[!duplicated(dnew910.before$id, fromLast=TRUE),]
dnew_uq$cluster.global <- fit.BCC$cluster.global
```

```
dnew_uq$postprob <- fit.BCC$postprob

# Posterior cluster probability
bp.BCClong <- ggplot(dnew_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=unique(dat[[1]]$id.org),
                          cluster.global=fit.BCC$cluster.global)
per <- round(100*table(dat.cluster$cluster.global)/length(dat.cluster$cluster.global),1)
dnew_uq$cluster.global.re <-  factor(dnew_uq$cluster.global,
                                     labels =paste("Cluster ",
                                                   1:num.clust.BCC," (",per,"%)",sep=""))
dat.cluster$cluster.global.re <-  dat.cluster$cluster.global
dat.cluster$cluster.global.re <- cluster.BCC <-  factor(dat.cluster$cluster.global.re,
                                    labels =paste("Cluster ",
                                                  1:num.clust.BCC," (",per,"%)",sep=""))
dnew <- merge(dnew910.before,dat.cluster,by="id")
```

```
library(ggplot2)
library(cowplot)
p1.BCC <- ggplot(data =dnew, aes(x = month, y = lbili,
                                 color=cluster.global.re,
                                 linetype=cluster.global.re,fill=cluster.global.re))+
  ggtitle("BCClong") +
        geom_smooth(aes(x =month, y = lbili,
                        color=cluster.global.re,
                        linetype=cluster.global.re,fill=cluster.global.re),
                    method = "loess", linewidth = 3,se = FALSE,span=2)+
        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")) +
        guides(fill=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
               color=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
                linetype=guide_legend(title=NULL,nrow = 2,byrow=TRUE)) +
        xlab("Time (months)") + ylab("lbili")  +
        ylim(c(min(dnew$lbili,na.rm=TRUE),max(dnew$lbili,na.rm=TRUE)))+
        scale_color_manual(values=c("green", "black"))+
        scale_fill_manual(values=c("green", "black"))

p2.BCC <- ggplot(data =dnew, aes(x = month, y = lalbumin,
                                 color=cluster.global.re,
                                 linetype=cluster.global.re,fill=cluster.global.re))+
  ggtitle("BCClong") +
        geom_smooth(aes(x =month, y = lalbumin,
                        color=cluster.global.re,
                        linetype=cluster.global.re,fill=cluster.global.re),
                    method = "loess", linewidth = 3,se = FALSE,span=2)+
        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")) +
```

```
            guides(fill=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
                   color=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
                linetype=guide_legend(title=NULL,nrow = 2,byrow=TRUE)) +
        xlab("Time (months)") + ylab("lalbumin")  +
        ylim(c(min(dnew$lalbumin,na.rm=TRUE),
               max(dnew$lalbumin,na.rm=TRUE)))+
        scale_color_manual(values=c("green", "black"))+
        scale_fill_manual(values=c("green", "black"))

p3.BCC <- ggplot(data =dnew, aes(x = month, y = lalk.phos,
                                 color=cluster.global.re,
                                 linetype=cluster.global.re,fill=cluster.global.re))+
  ggtitle("BCClong") +
        geom_smooth(aes(x = month, y = lalk.phos,
                        color=cluster.global.re,
                        linetype=cluster.global.re,fill=cluster.global.re),
                    method = "loess", linewidth = 3,se = FALSE,span=2)+
        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")) +
        guides(fill=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
               color=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
                linetype=guide_legend(title=NULL,nrow = 2,byrow=TRUE)) +
        xlab("Time (months)") + ylab("lalbumin")  +
        ylim(c(min(dnew$lalk.phos,na.rm=TRUE),
               max(dnew$lalk.phos,na.rm=TRUE)))+
        scale_color_manual(values=c("green", "black"))+
        scale_fill_manual(values=c("green", "black"))
p4.BCC <- ggplot(data =dnew, aes(x = month, y = lsgot,
                                 color=cluster.global.re,
                                 linetype=cluster.global.re,fill=cluster.global.re))+
  ggtitle("BCClong") +
        geom_smooth(aes(x = month, y = lsgot,
                        color=cluster.global.re,
                        linetype=cluster.global.re,fill=cluster.global.re),
                    method = "loess", linewidth = 3,se = FALSE,span=2)+
        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")) +
        guides(fill=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
               color=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
                linetype=guide_legend(title=NULL,nrow = 2,byrow=TRUE)) +
        xlab("Time (months)") + ylab("lalbumin")  +
```

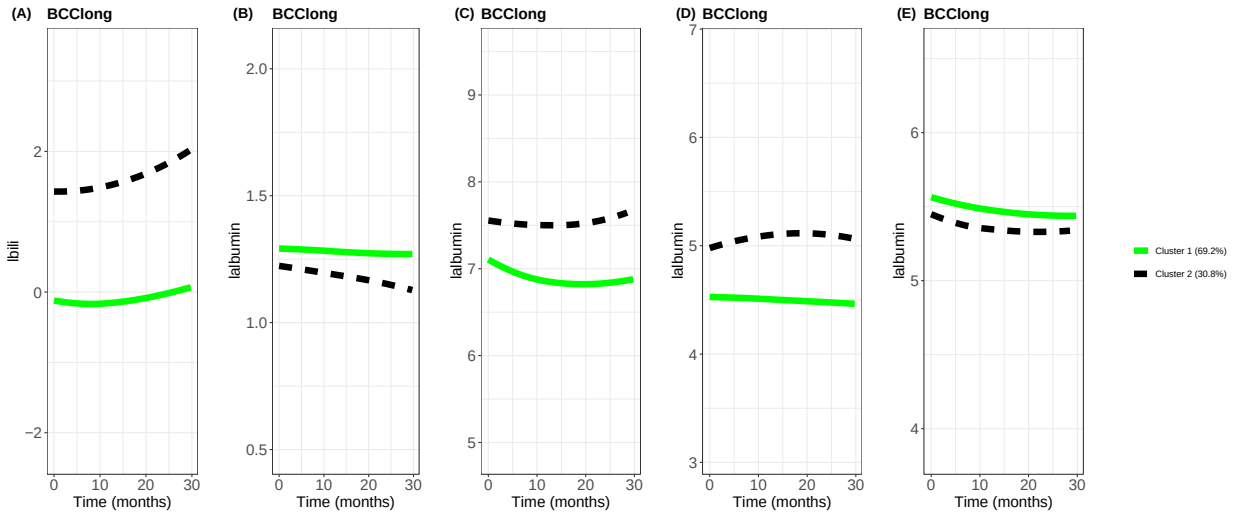
```
        ylim(c(min(dnew$lsgot,na.rm=TRUE),
               max(dnew$lsgot,na.rm=TRUE)))+
        scale_color_manual(values=c("green", "black"))+
        scale_fill_manual(values=c("green", "black"))
p5.BCC <- ggplot(data =dnew, aes(x = month, y = lplatelet,
                                 color=cluster.global.re,
                                 linetype=cluster.global.re,fill=cluster.global.re))+
  ggtitle("BCClong") +
        geom_smooth(aes(x = month, y = lplatelet,
                        color=cluster.global.re,
                        linetype=cluster.global.re,fill=cluster.global.re),
                    method = "loess", linewidth = 3,se = FALSE,span=2)+
        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")) +
        guides(fill=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
               color=guide_legend(title=NULL,nrow = 2,byrow=TRUE),
                linetype=guide_legend(title=NULL,nrow = 2,byrow=TRUE)) +
        xlab("Time (months)") + ylab("lalbumin")  +
        ylim(c(min(dnew$lplatelet,na.rm=TRUE),
               max(dnew$lplatelet,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 = month, y = lplatelet,
                                                color=cluster.global.re,
                                                linetype=cluster.global.re,fill=cluster.global.re))+
                           ggtitle("BCClong") +
                           geom_smooth(aes(x = month, y = lplatelet,
                                           color=cluster.global.re,
                                           linetype=cluster.global.re,fill=cluster.global.re),
                                       method = "loess", linewidth = 3,se = FALSE,span=2)+
                           theme_bw() +
                           theme(legend.position = c(0.5,0.5),
                                 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")) +
                           guides(fill=guide_legend(title=NULL,ncol = 1,byrow=TRUE),
                                  color=guide_legend(title=NULL,ncol = 1,byrow=TRUE),
                                  linetype=guide_legend(title=NULL,ncol = 1,byrow=TRUE)) +
                           xlab("Time (months)") + ylab("lalbumin")  +
                           ylim(c(min(dnew$lplatelet,na.rm=TRUE),
                                  max(dnew$lplatelet,na.rm=TRUE)))+
                           scale_color_manual(values=c("green", "black"))+
```

```
                            scale_fill_manual(values=c("green", "black"))
)
```

```
## Warning: Removed 15 rows containing non-finite values (`stat_smooth()`).
```

```
plot_grid(p1.BCC,NULL,p2.BCC,NULL,p3.BCC,NULL,
          p4.BCC,NULL,p5.BCC,NULL,legend.BCC,
          labels=c("(A)","", "(B)","","(C)","","(D)","","(E)","",""), nrow = 1,
          rel_widths = c(1,0.1,1,0.1,1,0.1,1,0.1,1,0.1,0.7))
```

```
## Warning: Removed 5 rows containing non-finite values (`stat_smooth()`).
## Removed 15 rows containing non-finite values (`stat_smooth()`).
```




```
library(survminer)
```

```
## Warning: package 'ggpubr' was built under R version 4.2.2
```

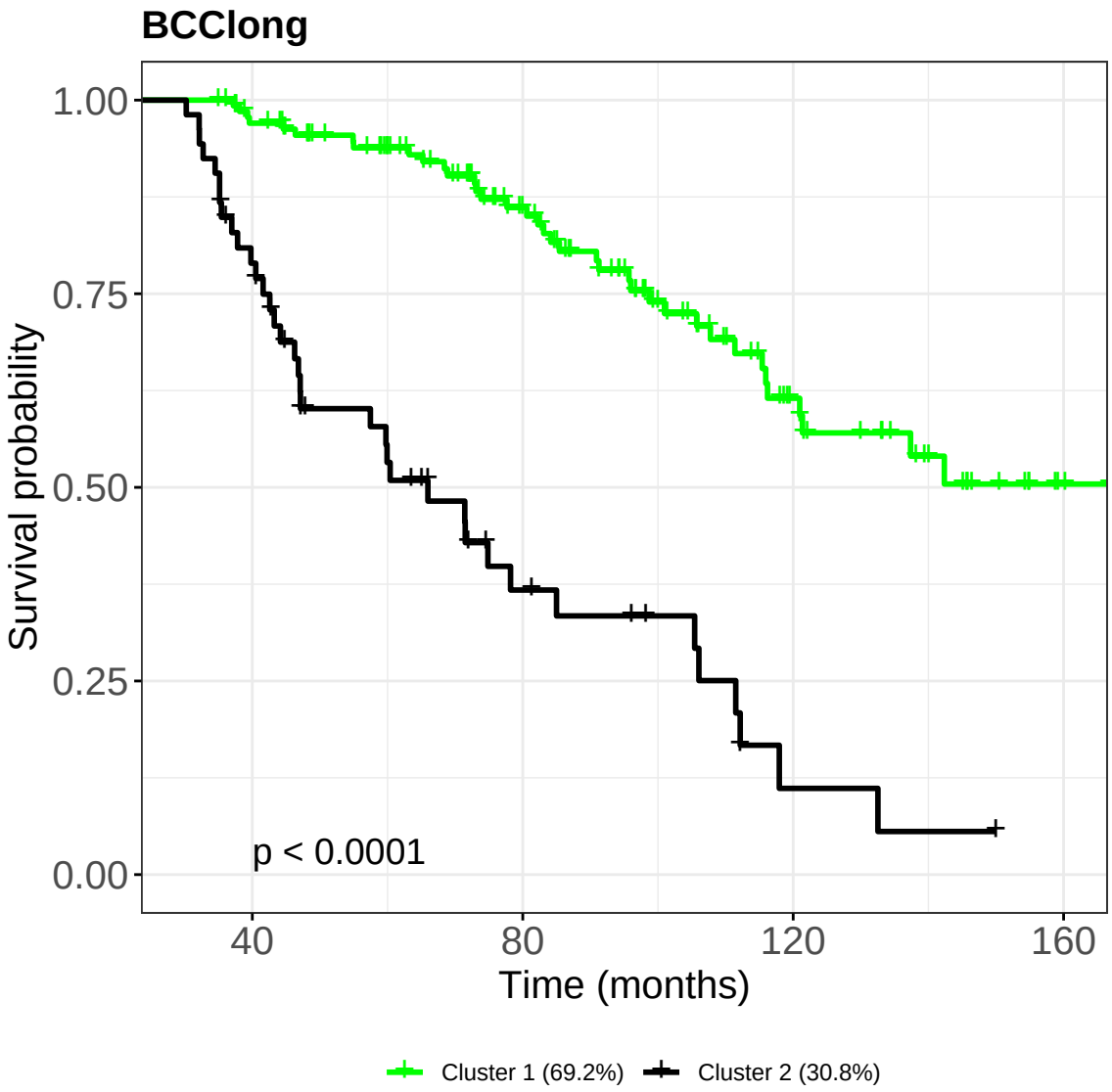
```
library(survival)
# use only data after 910 days (2.5 years)
dnew910.after <- dnew910[dnew910$day > 910,]; length(unique(dnew910.after$id))
```

```
## [1] 193
```

```
dnew910_uq <- merge(dnew910.after[!duplicated(dnew910.after$id, fromLast=TRUE),],
                    dnew_uq[,c("id","cluster.global.re","postprob")], by="id")
fit <- survfit(Surv(month, delta.death) ~  cluster.global.re,
               data = dnew910_uq, start.time=30.08)
# weighted cox model
res.cox <- coxph(Surv(month, delta.death) ~ cluster.global.re,
                 weights=postprob, data =  dnew910_uq )
pvalue <-  ifelse(summary(res.cox)$sctest[3] >= 0.0001,
                  summary(res.cox)$sctest[3],'<0.0001')

names(fit$strata) <-  paste("Cluster ",1:num.clust.BCC," (",per,"%)",sep="")
gp_survival.BCC <-   ggsurvplot(fit, data = dnew910_uq, title="BCClong",
                          risk.table = FALSE,
                 risk.table.y.text.col = FALSE,
                 pval = TRUE,
                 pval.coord = c(40, 0.03),
```

```
                          legend = "bottom", # conf.int = TRUE,
                          xlab = "Time (months)",
                 legend.title="Clusters",
                          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$plot <- gp_survival.BCC$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
```


BCClong
1.00
0.75
0.50
0.25
0.00
Survival probability
p < 0.0001
40
80
120
160
Time (months)
Cluster 1 (69.2%)
Cluster 2 (30.8%)