**R Script**

Biomarker data was first log-transformed

All p-values were multiple testing adjusted: dataframe$adj\_p <- p.adjust(dataframe$p\_values, method = "BH")

**Figure 1A: Kaplan-Meijer 30-day mortality**

# Load necessary packages

Library(survival)

p <- ggsurvplot(survfit(Surv(days\_survived,day30\_mortality) ~ lymphocyte\_grouping, data = dataframe),

break.time.by = 10,

conf.int = F,

xlab = "Survival days",

ylab = "Survivor probability",

legend.labs = c("No lymphopenia", "Mild lymphopenia", "Severe lymphopenia"),

palette =c("#4DAF4A", "#984EA3", "#377EB8"),

pval = T,

risk.table = T,

risk.table.title = "",

fontsize = 3,

xlim = c(0,31))

p

**Figure 1B + C: non-linear mortality relationship**

## first check linearity

library(rms)

unadj\_lympho <- lrm(day30\_mortality ~ rcs(Lymphocyte\_count, 3), data= dataframe)

anova(unadj\_lympho)

# Load necessary packages

library(ggplot2)

library(scales)

# Define data preparation

d <- datadist(dataframe)

options(datadist = "d")

# Define unadjusted analysis

unadj\_lympho <- lrm(day30\_mortality ~ rcs(Lymphocyte\_count, 3), data = dataframe)

# Update datadist for adjustment

d$limits["Adjust to", "Lymphocyte\_count"] <- 1

unadj\_lympho <- update(unadj\_lympho)

x1 <- Predict(unadj\_lympho, Lymphocyte\_count, ref.zero = TRUE, fun = exp)

## Predict change from 1.0 to 0.5 for unadjusted analysis

Predict(unadj\_lympho, ref.zero = T, Lymphocyte\_count = 0.5, fun = exp)

# Create ggplot for unadjusted analysis

x3 <- ggplot(x1) +

xlab("Lymphocyte count") +

ylab("30-day Mortality Odds ratio") +

theme\_bw() +

theme(legend.position = "bottom", panel.grid.major = element\_blank(), panel.grid.minor = element\_blank()) +

geom\_vline(xintercept = 1, linetype = 'dotted', col = 'purple') +

geom\_vline(xintercept = 0.5, linetype = 'dotted', col = 'blue') +

scale\_x\_continuous(breaks = c(0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2.0, 2.25),

name = "Lymphocyte count", limits = c(0.25, 2.25)) +

scale\_y\_continuous(breaks = c(0.75, 1, 2.5, 5, 7.5, 10, 12.5)) +

coord\_trans(y = 'log')

x3

# Define adjusted analysis

adj\_lymf <- lrm(day30\_mortality ~ + age + inclusion\_hospital + wave2 + wave3 + wave4 + sex +

malignancy + chronic\_immunosupression + anti\_il6\_treatment + remdesivir\_treatment + imatinib\_treatment + cortico\_treatment + antibiotic\_treatment + rcs(Lymphocyte\_count, 3), data = dataframe)

# Summary of adjusted analysis

d$limits["Adjust to", "Lymphocyte\_count"] <- 1

adj\_lymf <- update(adj\_lymf)

x1 <- Predict(adj\_lymf, Lymphocyte\_count, ref.zero = TRUE, fun = exp)

## Predict change from 1.0 to 0.5 for adjusted analysis

Predict(adj\_lymf, ref.zero = TRUE, Lymphocyte\_count = 0.5, fun = exp)

# Create ggplot for adjusted analysis

x3 <- ggplot(x1) +

xlab("Lymphocyte count") +

ylab("30-day Mortality Odds ratio") +

theme\_bw() +

theme(legend.position = "bottom", panel.grid.major = element\_blank(), panel.grid.minor = element\_blank()) +

geom\_vline(xintercept = 1, linetype = 'dotted', col = 'purple') +

geom\_vline(xintercept = 0.5, linetype = 'dotted', col = 'blue') +

scale\_x\_continuous(breaks = c(0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2.0, 2.25),

name = "Lymphocyte count", limits = c(0.25, 2.25)) +

coord\_trans(y = 'log') +

scale\_y\_continuous(breaks = c(0.75, 1, 2.5, 5, 7.5, 10))

x3

**Figure 2: PCA analysis per domain (endothelial as example)**

p\_load(vegan, devtools, ggibplot)

endo <- dataframe[c("endo biomarker1",   
 "endo biomarker2", ….. , “ lymphocyte\_grouping")]

##

Groups <-endo[,"lymphocyte\_grouping "]

e.pca <- prcomp(endo[ ,1:#number of markers], center = T, scale. = T)

colors <- c("#4DAF4A", "#984EA3", "#377EB8")

e.plot1<-ggbiplot(e.pca, ellipse=F, obs.scale = 1, var.scale = 1,var.axes=T, group=Groups,

circle = F, varname.size=2.0, alpha=0, varname.adjust=c(1))+

scale\_color\_manual(name="Groups",values=colors)+

stat\_ellipse(aes(colour=Groups), size = 1, type="norm", level = 0.10)+

coord\_fixed(xlim=c(-3,3), ylim=c(-3, 3))+

theme\_bw() +

theme(panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank())+

theme(text = element\_text(size = 8),

axis.title = element\_text(size = 8),

axis.text.x=element\_text(size = 8),

axis.text.y=element\_text(size = 8),

legend.position = "none")+

scale\_y\_continuous(breaks=seq(-3, 3, by = 1)) +

ggtitle("Endothelial and coagulation response")

e.plot1

## significance PC’s

pcdat <- as.data.frame(e.pca[["x"]])

pcdat$group <- Groups

summary(aov(pcdat$PC1 ~ pcdat$group))

summary(aov(pcdat$PC2 ~ pcdat$group))

**## Heatmap figure 3 with p-values**

**Construction of hedges g heatmap**

effect\_size1 <- low\_vs\_normal %>%

group\_by(domain, variable) %>%

cohens\_d(value ~ lymphocyte\_grouping, hedges.correction = T, var.equal = F) %>% as.data.frame() %>%

mutate(effsize\_2 = ifelse(effsize > -0.2 & effsize < 0.2, 0, effsize),

effsize\_cat = cut(effsize,

breaks = c(-Inf, -1.5, -0.8, -0.5, -0.2, 0.2, 0.5, 0.8, 1.5, Inf),

labels = c("Very large decrease (<-1.5)", "Large decrease (<-0.8)", "Moderate decrease (<-0.5)",

"Small decrease (<-0.2)", "Negligible difference", "Small increase (>0.2)",

"Moderate increase (>0.5)", "Large increase (>0.8)", "Very large increase (>1.5)")) %>% as.factor())

effect\_size1$comparison <- 1

effect\_size2 <- verylow\_vs\_normal %>%

group\_by(domain, variable) %>%

cohens\_d(value ~ lymphocyte\_grouping, hedges.correction = T, var.equal = F) %>% as.data.frame() %>%

mutate(effsize\_2 = ifelse(effsize > -0.2 & effsize < 0.2, 0, effsize),

effsize\_cat = cut(effsize,

breaks = c(-Inf, -1.5, -0.8, -0.5, -0.2, 0.2, 0.5, 0.8, 1.5, Inf),

labels = c("Very large decrease (<-1.5)", "Large decrease (<-0.8)", "Moderate decrease (<-0.5)",

"Small decrease (<-0.2)", "Negligible difference", "Small increase (>0.2)",

"Moderate increase (>0.5)", "Large increase (>0.8)", "Very large increase (>1.5)")) %>% as.factor())

effect\_size2$comparison <- 2

## combine

effect\_size <- rbind(effect\_size1, effect\_size2)

## wide matrix

test <- effect\_size %>% select(domain, variable, effsize, comparison) %>%

pivot\_wider(names\_from = "comparison", values\_from = effsize)

## change colors

input <- as.matrix(test[ , c(-1, -2)])

input\_save <- input

rownames(input\_save) <- test$variable

input[input >-0.2 & input <0.2] <- 0

rownames(input) <- test$variable

##

p\_load(circlize)

min(input)

max(input)

col\_fun = colorRamp2(breaks=c(-0.6, 0, 0.3, 1.5), colors= c("#334B97","white", "red", "darkred"))

dis\_hm <- Heatmap(input, col = col\_fun,

column\_title\_gp = gpar(size=6, fontface="bold"), show\_column\_names = T, column\_names\_rot = 0,

show\_row\_names = T, show\_row\_dend = F, show\_column\_dend = F, row\_names\_side = "right",

row\_names\_gp = gpar(fontsize = 8), cluster\_rows = F, cluster\_columns = F,

border=T,

row\_order = order2,

row\_split = factor(test$domain, levels = c("Endothelial and coagulation activation", "Inflammation and organ damage","Chemokine", "Cytokines")),

cluster\_row\_slices = F,

heatmap\_legend\_param = list(at = c(-0.8, -0.5, -0.2, 0, 0.2, 0.8, 1.5),

labels = c("<-0.8", "<-0.5", "<-0.2", ">-0.2 & <0.2", ">0.2", ">0.8", ">1.5"),

title = "Hedges' g", color\_bar = "discrete", fontsize = 8),

row\_title\_rot = 0, row\_title\_gp = gpar(fontsize = 8, fontface="bold"),

column\_names\_gp = gpar(fontsize = 8, fontface="bold"),

rect\_gp = gpar(col = "black", lwd = 0.5),

border\_gp = gpar(col = "black", lwd = 1))

dis\_hm

**## Significance Figure 3**

**1. Check linearity biomarkers and lymphocyte counts using a loop and for non-linear markers calculate p-value with lymphocyte counts**

## linearity

p\_load(rms)

linearity <- function(x){

pvalue\_linear <- anova(ols(x ~ rcs(Lymphocyte\_counts, 3), data =dataframe))[2,5]

pvalue\_lympho <- anova(ols(x ~ rcs(Lymphocyte\_counts, 3), data =dataframe))[1,5]

vec <- c( pvalue\_linear, pvalue\_lympho)

names(vec) <- c("non-linearity", "pvalue\_lympho")

return(vec )

}

P<0.05 of the linear term indicating non-linearity

**2. unadjusted p-value lymphocyte for linear marker**

regression <- function(x){

reg <- anova(lm(x ~ Lymphocyte\_counts, data = dataframe))[1,5]

vec <- reg

names(vec) <- c("p\_value")

return(vec)

}

**3. Adjusted models non-linear markers + linear markers**

linearity <- function(x){

linear <- anova(ols(x ~ age + inclusion\_hospital + wave2 + wave3 + wave4 + sex +

malignancy + chronic\_immunosupression + anti\_IL6\_before\_sampling + imatinib\_before\_sample + dexa\_before\_sample + other\_cortico\_before\_sample + rcs(Lymphocyte\_counts, 3) , data =dataframe))[14,5]

pvalue\_lympho <- anova(ols(x ~ age + inclusion\_hospital + wave2 + wave3 + wave4 + sex +

malignancy + chronic\_immunosupression + anti\_IL6\_before\_sampling + imatinib\_before\_sample + dexa\_before\_sample + other\_cortico\_before\_sample + rcs(Lymphocyte\_counts, 3), data =dataframe))[13,5]

vec <- c(linear, pvalue\_lympho)

names(vec) <- c("nonlinear", "pvalue\_lympho")

return(vec )

}

regression <- function(x){

reg <- anova(lm(x ~ age + inclusion\_hospital + wave2 + wave3 + wave4 + sex +

malignancy + chronic\_immunosupression + anti\_IL6\_before\_sampling + imatinib\_before\_sample + dexa\_before\_sample + other\_cortico\_before\_sample + Lymphocyte\_counts, data = dataframe))[13,5]

vec <- reg

names(vec) <- c("p\_value")

return(vec)

}

**Direct comparison Figure 3 (not minus as data is already log-transformed)**

All\_data\_long$lymphocyte\_grouping <- factor(All\_data\_long$very\_low\_low, levels = c( "very\_low", "low", "Normal"), ordered = T)

tt <- function(x){

ttp <- t.test(x ~ low\_lympho\_subset$lymphocyte\_grouping)$p.value

logfc <- t.test(x ~ low\_lympho\_subset$lymphocyte\_grouping)$estimate[2] -

t.test(x ~ low\_lympho\_subset$lymphocyte\_grouping)$estimate[1]

vec <- c(ttp,logfc)

names(vec) <- c("p","logfc")

return(vec )

}

**Volcano plot**

p <- ggplot(t.tests, aes(x=log10fc, y= -log10(adj.p), label=label )) +

geom\_point(aes(col=col, size=0.2)) +

scale\_color\_manual(values = c("A" = "grey", "B" = "blue", "C" = "red")) +

scale\_shape\_manual(values=c(16,1,16,16))+

theme\_bw() +

theme(legend.position = "none",

plot.title = element\_text(face="bold", size = 15),

axis.title.y = element\_text(face="bold",size=15),

axis.title.x = element\_text(size=15),

axis.text.x = element\_text(#face="bold",

size=13, color = "black"),

axis.text.y = element\_text(#face="bold",

size=13, color = "black"),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

panel.border = element\_rect(colour = "black", fill=NA, size=1)) +

ylab(expression('-log'[10]\*'(BH adjusted P)')) +

geom\_hline(yintercept = 1.30103) +

coord\_cartesian(xlim = c(-0.5, 0.5), ylim=c(0, -log10(min(t.tests$adj.p))+1 )) +

geom\_text\_repel() +

xlab("LogFC") +

ylab(expression('-log'[10]\*'(BH adjusted P)')) +

geom\_hline(yintercept = 1.30103) +

ggtitle("Mild vs severe lymphopenia") #change title

p

**Figure 4: Mortality coordinates in lymphopenic patients**

## lymphocyte derived biomarkers

products <- c("TNF-alpha", "TNF-RI" , "Granzyme B", "CD40L", "CCL3" ,"CCL4", "CCL5", "IL2", "IL4", "IL10", "IL13", "IL17a", "IL6", "GM-CSF", "IFNg", "CXCL10")

## assess linearity mortality and lymphocyte counts in subset of lymphopenic patients => linear

library(rms)

lympho <- lrm(day30\_mortality ~ rcs(Lymphocyte\_count, 3), data= low\_lymf)

anova(lympho)

## assess nonlinearity biomarkers with lymphocyte counts in lymphopenic patients => all linear relationship

linearity <- function(x){

linear <- anova(ols(x ~ rcs(Lymphocyte\_counts, 3), data =low\_lymf))[2,5]

pvalue\_lympho <- anova(ols(x ~ rcs(Lymphocyte\_counts, 3), data =low\_lymf))[1,5]

vec <- c(linear, pvalue\_lympho)

names(vec) <- c("nonlinear", "pvalue\_lympho")

return(vec )

}

## first check interaction => IL-10 shows an association => removed

mortality\_inter <- function(x){

coeff <- summary(glm(mortality\_d30 ~ x \*Lymphocyte\_counts, data = low\_lymf, family = "binomial"))[["coefficients"]][2,1]

p\_value <- summary(glm(mortality\_d30 ~ x \* Lymphocyte\_counts, data = low\_lymf, family = "binomial"))[["coefficients"]][2,4]

p\_value\_interaction <- summary(glm(mortality\_d30 ~ x \*Lymphocyte\_counts, data = low\_lymf, family = "binomial"))[["coefficients"]][4,4]

vec <- c(coeff, p\_value, p\_value\_interaction)

names(vec) <- c("Coefficient\_mort", "p\_value\_mort", "p\_interaction")

return(vec)

}

## get lymphocyte coefficient

coefficient <- function(x){

coeff <- summary(lm(x ~ Lymphocyte\_counts, data = low\_lymf))[["coefficients"]][2,1]

p\_value <- summary(lm(x ~ Lymphocyte\_counts, data = low\_lymf))[["coefficients"]][2,4]

vec <- c(coeff, p\_value)

names(vec) <- c("Coefficient\_lymphocyte", "p\_value\_lymphocyte")

return(vec)

}

## get percentage increase per 0.1 increase in lymphocytes

coeff.results$percentage\_lympho <- (exp(coeff.results$Coefficient\_lymphocyte) - 1) \* 100/10

## mortality

mortality <- function(x){

coeff <- summary(glm(mortality\_d30 ~ x, data = low\_lymf, family = "binomial"))[["coefficients"]][2,1]

p\_value <- summary(glm(mortality\_d30 ~ x, data = low\_lymf, family = "binomial"))[["coefficients"]][2,4]

vec <- c(coeff, p\_value)

names(vec) <- c("Coefficient\_mort", "p\_value\_mort")

return(vec)

}

## get increase in mortality with 25% increase in biomarker

mort.results$odds\_change <- (1.25^mort.results$Coefficient\_mort)

## combine for plot

importance <- merge(coeff.results, mort.results, by = "marker")

# Volcano plot with coordinates

importance$col <- "A"

importance$col <- ifelse(importance$bh\_adj\_lymphocyte<0.05 & importance$bh\_adj\_mort <0.05, "B", importance$col)

importance$col <- ifelse(importance$bh\_adj\_lymphocyte<0.05 & importance$bh\_adj\_mort >=0.05, "C", importance$col)

importance$col <- ifelse(importance$bh\_adj\_lymphocyte>=0.05 & importance$bh\_adj\_mort <0.05, "D", importance$col)

#label and color

importance$label <- importance$marker

colors <- c("#66C2A5","#FC8D62","#8DA0CB","#E78AC3","#A6D854","#FFD92F","#E5C494","#B3B3B3")

# ggplot

library(ggplot2)

library(ggrepel)

importance\_plot <- ggplot(importance, aes(x= odds\_change,y=percentage\_lympho, label=label)) +

geom\_point(aes(col=col, size=0.5)) +

scale\_color\_manual(values = c("A" = "grey", "B" = "orange", "C" = "purple", "D" = "#A6D854")) +

scale\_shape\_manual(values=c(16,1,16,16))+

theme\_bw() +

theme(legend.position = "none",

plot.title = element\_text(face="bold", size = 15),

axis.title.y = element\_text(face="bold",size=15),

axis.title.x = element\_text(size=15),

axis.text.x = element\_text(#face="bold",

size=13, color = "black"),

axis.text.y = element\_text(#face="bold",

size=13, color = "black"),

panel.grid.major = element\_blank(),

panel.grid.minor = element\_blank(),

panel.border = element\_rect(colour = "black", fill=NA, size=1)) +

geom\_text\_repel(max.overlaps = 9999) +

xlab("Increase in 30-day mortality odds\n per 25% increase in biomarker") +

ylab(expression("Percentage increase in biomarker\n per 0.1 increase in lymphocycte counts")) +

geom\_hline(yintercept=0, linetype='dotted', col = 'grey', size = 0.8) +

geom\_vline(xintercept=1, linetype='dotted', col = 'grey', size = 0.8) +

coord\_cartesian(xlim = c(0.75, 1.8)) +

scale\_x\_continuous(trans = "log", breaks = c(0.8, 1.00, 1.2, 1.4, 1.6, 1.8, 2.0)) +

scale\_y\_continuous(lim = c(-5, 7.5),

breaks = c(-5,-2.5, 0, 2.5, 5, 7.5),

minor\_breaks = NULL,

labels = c(-5, -2.5, 0, 2.5, 5, 7.5))

importance\_plot

**Figure 5A: clustering and scaling**

library(NbClust)

library(clValid)

library(clustree)

library(ggplot2)

library(cluster)

library(heatmaply)

library(RColorBrewer)

library(ComplexUpset)

# Scaling

scaling\_means <- colMeans(df\_clust\_biomarkers)

scaling\_sd <- apply(df\_clust\_biomarkers, 2, sd)

df\_clust\_biomarkers <- scale(df\_clust\_biomarkers)

row.names(df\_clust\_biomarkers) <- df\_clust$identifier

df\_clust\_biomarkers <- as.data.frame(df\_clust\_biomarkers)

## find best clustering method

intern <- clValid(df\_clust\_biomarkers, nClust = 2:10,

clMethods = c("hierarchical","kmeans","pam"), validation = "internal")

summary(intern) ## => hierarchical

## Determine optimal number using different max clusters

p\_load(NbClust, clValid, clustree)

clust <- NbClust(data=as.matrix(df\_clust\_biomarkers), method="ward.D2", distance = "euclidean", index = "all", max.nc = 5)

clust <- NbClust(data=as.matrix(df\_clust\_biomarkers), method="ward.D2", distance = "euclidean", index = "all", max.nc = 10)

clust <- NbClust(data=as.matrix(df\_clust\_biomarkers), method="ward.D2", distance = "euclidean", index = "all", max.nc = 15)

clust <- NbClust(data=as.matrix(df\_clust\_biomarkers), method="ward.D2", distance = "euclidean", index = "all", max.nc = Inf)

## get results

clust$Cluster\_result<-NA

clust$Cluster\_result<-as.factor(clust$Best.partition)

df\_clust$cluster <- clust$Cluster\_result

# Compute dissimilarity matrix with euclidean distances

dev.off()

d <- dist(df\_clust\_biomarkers, method = "euclidean")

res.hc <- hclust(d, method = "ward.D2" )

grp <- cutree(res.hc, k = 3)

plot(res.hc, cex = 0.2)

rect.hclust(res.hc, k = 3, border = 2:5)

## Make heatmap

row.names(df\_clust\_biomarkers) == df\_clust$identifier

row\_ha <- rowAnnotation(Cluster = df\_clust$cluster,

Mortality = df\_clust$mort\_ICU,

col = list(Cluster = c("1" = "aquamarine3", "2" = "darkgoldenrod1", "3" = "darkmagenta"),

Mortality = c("1" = "Red", "0" = "black"))

hist\_major <- Heatmap(as.matrix(df\_clust\_biomarkers),

split = df\_clust$cluster,

column\_names\_gp = gpar(fontsize = 10),

name = "Scaled\nvalue",

show\_column\_dend = FALSE,

show\_row\_names = FALSE,

use\_raster= TRUE,

raster\_resize\_mat = max,

border\_gp = gpar(col = "black", lty = 2)) + row\_ha

hist\_major

**Figure 5B and figure 5C: PCA plots and hedges g comparing clusters**

R Code in analogy to the previously described PCA analysis and hedges g heatmap

**Figure 6: see python code**

## Example of testing proportions

## test distribution mortality clusters in both cohorts

discovery\_cohort\_mortality\_counts <- c(29, 5, 6)

validation\_cohort\_mortality\_counts <- c(12, 1, 2)

# Create a contingency table with the class counts in each cohort

table <- matrix(c(discovery\_cohort\_mortality\_counts, validation\_cohort\_mortality\_counts), nrow = 3)

colnames(table) <- c("Discovery Cohort", "Validation Cohort")

rownames(table) <- c("Cluster 1", "Cluster 2", "Cluster 3")

# Perform the Fisher's exact test

fisher.test(table)