Survival analysis of patients with osteosarcoma

Léa ROGUE

2025-03-13

List of Figures

Contents

This script performs survival analysis of osteosarcoma samples (microarray, GSE21257).

Load librairies

```
library(dplyr)
library(survival)
library(forestplot)
library(survminer)
library(ggplot2)
library(ggsurvfit)
library(gridExtra)
library(tidyr)
library(stringr)
```

Functions

```
# Function to apply coxph model per columns
apply_coxph_model <- function(df) {
    # Empty df
    results <- data.frame()

    # Loop on each columns to apply coxph
    for (col in colnames(df)[3:ncol(df)]) {
        model <- coxph(Surv(OS.delay, OS.event) ~ df[[col]],
            data = df)

        # Extract results
        exp_coef <- summary(model)$coefficients[, "exp(coef)"]
        lower_ci <- summary(model)$conf.int[, "lower .95"]
        upper_ci <- summary(model)$conf.int[, "upper .95"]
        p_value <- summary(model)$coefficients[, "Pr(>|z|)"]
        results <- rbind(results, data.frame(Variable = col,
            HR = exp_coef, LowerCI = lower_ci, UpperCI = upper_ci,
            Pvalue = p_value))
    }
    return(results)
}

# Function to create Kaplan-Meier plot
plot_km <- function(mod, df) {
    km_plot <- ggsurvplot(mod, data = df, pval = TRUE, conf.int = FALSE,
        ggtheme = theme_bw(), palette = c("#E7B800", "#2E9FDF",
            "#FF6F61", "#4EBB92"))
    return(km_plot$plot)
}

# Function to generate forest plot from coxph analysis from
# the function apply_coxph_model
```

```
generate_forestplot <- function(data, Type) {
    if (Type == TRUE) {
        # Order by Signature and p-value
        data <- data[order(data$Signature, data$Pvalue), ]
        data$Pvalue <- sprintf("%.4f", data$Pvalue)
        data %>%
            forestplot(labeltext = c(Signature, Variable, Pvalue),
                mean = HR, lower = LowerCI, upper = UpperCI,
                grid = TRUE, zero = 1, col = fpColors(box = "black",
                  line = "black"), hrzl_lines = TRUE, title = "Forest Plot for Cox Model",
                txt_gp = fpTxtGp(label = gpar(fontsize = 8)),
                ci.vertices = TRUE, boxsize = 0.1) %>%
            fp_set_zebra_style("#EFEFEF") %>%
            fp_add_header(Signature = c("", "Cell type"), Variable = c("",
                "Genes"), Pvalue = c("", "p-value"))
    } else {
        # Order by p-value
        data <- data[order(data$Pvalue), ]
        data$Pvalue <- sprintf("%.4f", data$Pvalue)
        data %>%
            forestplot(labeltext = c(Variable, Pvalue), mean = HR,
                lower = LowerCI, upper = UpperCI, grid = TRUE,
                zero = 1, col = fpColors(box = "black", line = "black"),
                hrzl_lines = TRUE, title = "Forest Plot for Cox Model",
                txt_gp = fpTxtGp(label = gpar(fontsize = 8)),
                ci.vertices = TRUE, boxsize = 0.1) %>%
            fp_set_zebra_style("#EFEFEF") %>%
            fp_add_header(Variable = c("", "Genes"), Pvalue = c("",
                "p-value"))
    }
}
```

Load and format data

```
# Matrix with z-scores intensities and information on
# immune signature and CTA
df_expr_z_scores <- read.table("../results/expr_matrix_CTA_sign_imm_z_scores.tsv",
    sep = "\t", header = TRUE, check.names = FALSE)

# Matrix with average expression per cell types in z-scores
df_imm_z_scores <- read.table("../results/imm_sign_avg_z_scores.tsv",
    sep = "\t", header = TRUE, check.names = FALSE)
rownames(df_imm_z_scores) <- df_imm_z_scores$Signature
df_imm_z_scores <- as.data.frame(t(df_imm_z_scores[, -1]))
```

```
# Create a better matrix
metadata <- read.table("../data/metadata.tsv", sep = "\t", header = F)
df_metadata <- as.data.frame(t(metadata))
colnames(df_metadata) <- df_metadata[1, ]
df_metadata <- df_metadata[-1, ]
colnames(df_metadata) <- gsub(" ", "_", colnames(df_metadata))
```

```r
df_metadata$OS.event <- ifelse(grepl("Alive", df_metadata$status),
    0, 1)
df_metadata$OS.delay <- as.numeric(str_extract(df_metadata$status,
    "\\d+"))
df_metadata_surv <- df_metadata %>%
    select(Patient, OS.delay, OS.event)
```

I. Immune cells survival analysis

This part concern the survival analysis with the expression of the immune cells. This is to observe the impact of each cell types on the survival probabilities of the patients.

a- CoxPH model

In this section, we use z-scores data with Cox model.

```
# Continuous data
df_cont_all <- df_imm_z_scores

# Replace space by _
colnames(df_cont_all) <- gsub(" ", "_", colnames(df_cont_all))

# Merge df
df_cont_all$Patient <- rownames(df_cont_all)
df_survival_cont <- merge(df_metadata_surv, df_cont_all, by = "Patient")
rownames(df_survival_cont) <- df_survival_cont$Patient
df_survival_cont <- df_survival_cont[, -1]


# Apply coxph on continuous
results_cont <- apply_coxph_model(df_survival_cont)
# write.table(results_cont,
# '../results/results_coxph_var_cont.tsv', sep = '\t',
# row.names = FALSE, quote = FALSE)
# pdf('../results/figures/forest_plots/all_indiv/forest_plot_all_patients_var_cont.pdf')
generate_forestplot(results_cont, Type = FALSE)
```

```
# dev.off()
```

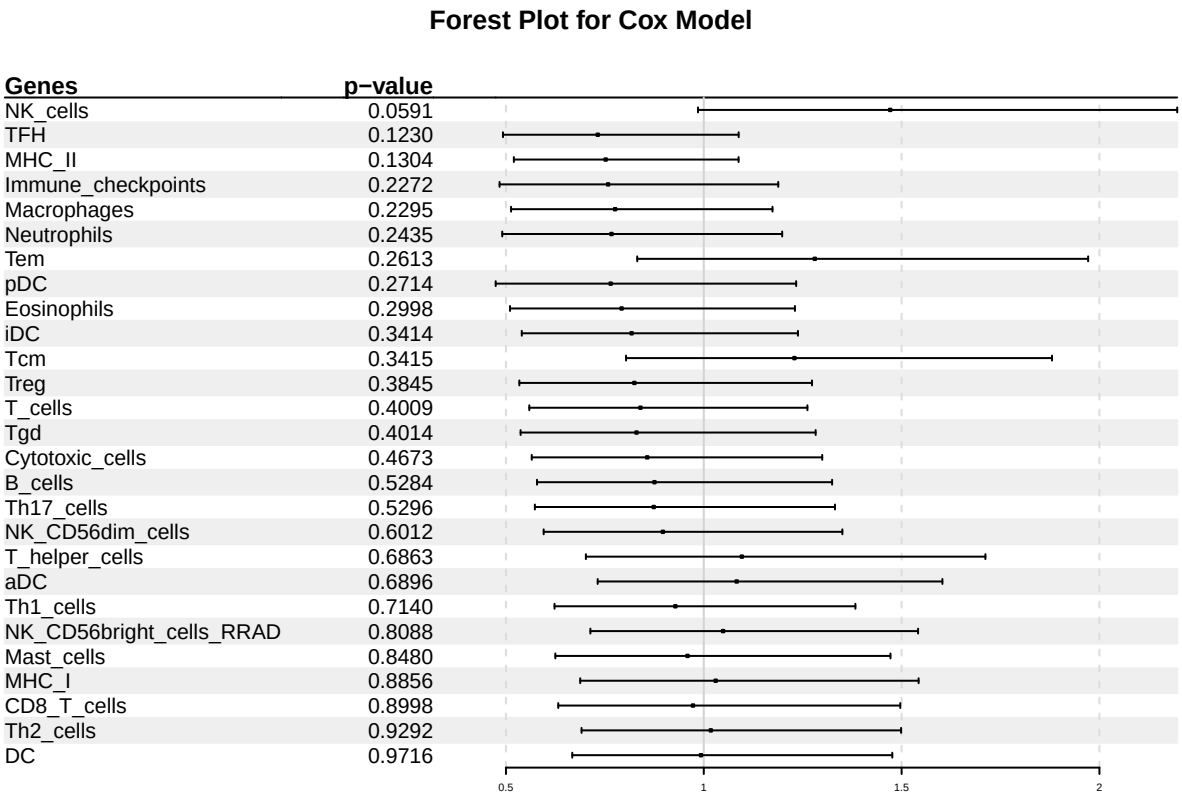

Figure 1: Forest plot for continuous immune cell types expression (n = 53)

II. CTA survival analysis

This section analyzes the impact of CTA on survival probabilites.

```
df_expr_cta <- subset(df_expr_z_scores, CTA == "CTA")
rownames(df_expr_cta) <- df_expr_cta$SYMBOL
df_expr_cta <- df_expr_cta[, -c(1:3)]
df_expr_cta <- as.data.frame(t(df_expr_cta))
df_expr_cta$Patient <- rownames(df_expr_cta)
df_expr_cta <- merge(df_metadata_surv, df_expr_cta, by = "Patient")
rownames(df_expr_cta) <- df_expr_cta$Patient
df_expr_cta <- df_expr_cta[, -1]

res_cta <- apply_coxph_model(df_expr_cta)

# Forest plot on significative CTA
res_cta_signif <- res_cta[res_cta$Pvalue < 0.05, ]
# write.table(res_cta_signif,
# '../results/results_coxph_osteo_cta_zscore_signif.tsv',
# sep = '\t', row.names = FALSE, quote = FALSE)
# pdf('../results/figures/forest_plots/forest_plot_cta_signif.pdf',
# width = 10, height = 15)
generate_forestplot(res_cta_signif, Type = FALSE)
```

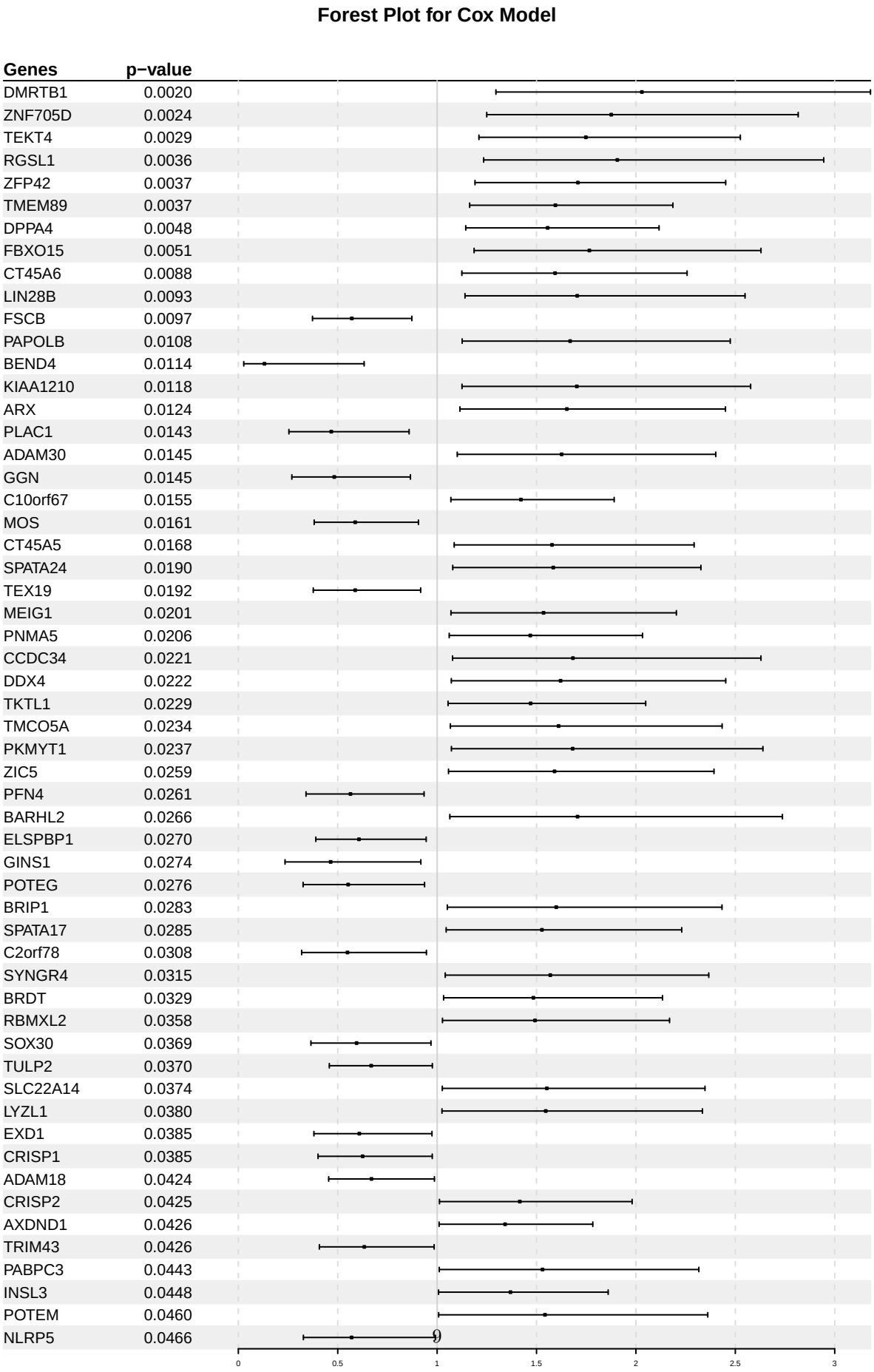

Figure 2: Forest plot for significant CTA

```
# dev.off()
```

This forest plot illustrates only the significant HR. We see here that there is more bad impact CTA than good impact.

III. Survival analysis for clusters from CTA heatmaps

This section use clustering results from script 1 on expression analysis.

1) Significative coxph CTA genes clustering

This clustering is from heatmap 3.

```
# Read clusters
l_clust <- read.table("../results/clusters_cta_signif_coxph_osteo_53.tsv",
    header = TRUE, sep = "\t")

# Prepare table
df_cluster_cta_all <- merge(l_clust, df_metadata_surv, by = "Patient")
rownames(df_cluster_cta_all) <- df_cluster_cta_all$Patient
p <- plot_km(survfit(Surv(OS.delay, OS.event) ~ Cluster, data = df_cluster_cta_all),
    df_cluster_cta_all)
p
```

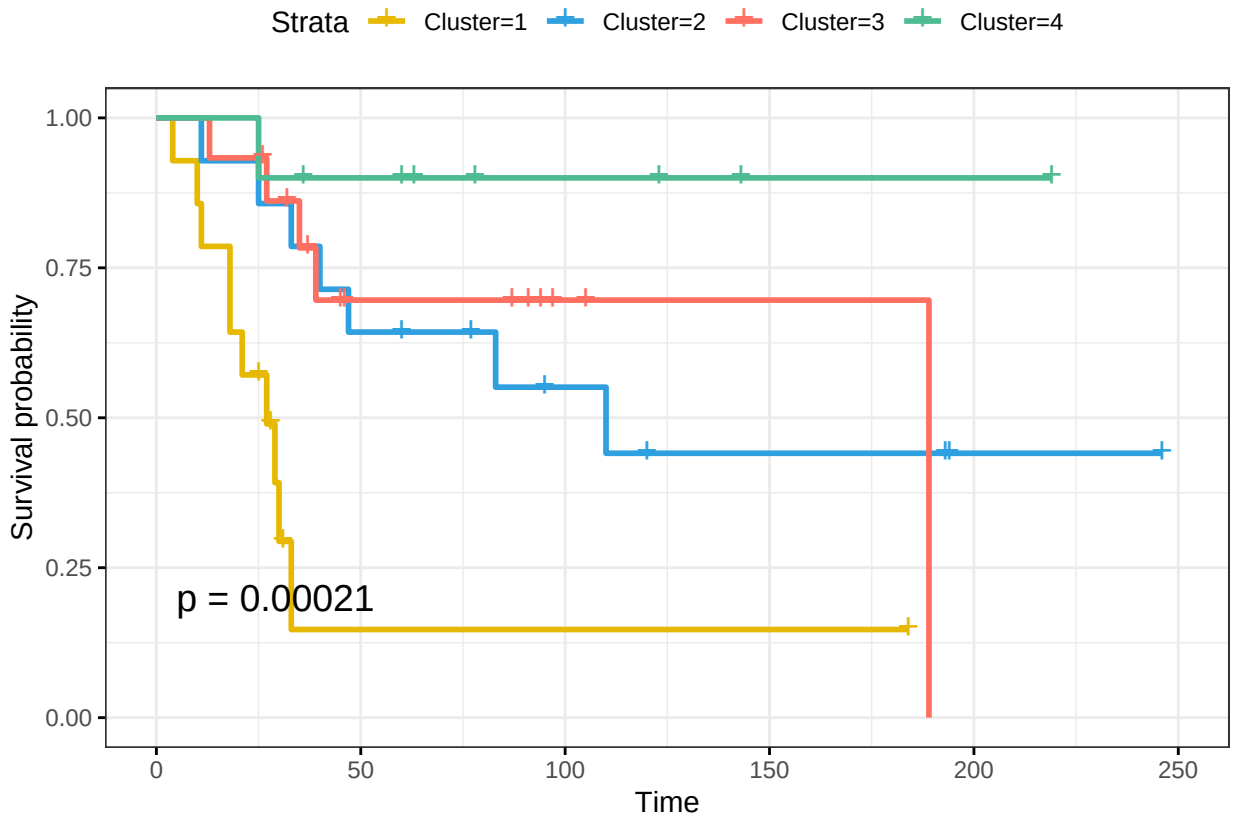

Figure 3: Kaplan-Meier plot with 4 clusters

This confirm the KM plot with all patients. We clearly see that some CTA have bad impact on survival.

Merge C2 and C3 Here, we want to merge the 2 clusters to see the differences than with 3.

```r
# Prepare table
df_cluster_cta_all_merge <- merge(l_clust, df_metadata_surv,
    by = "Patient")
df_cluster_cta_all_merge$Cluster <- ifelse(df_cluster_cta_all_merge$Cluster ==
    2, "2+3", df_cluster_cta_all_merge$Cluster)
df_cluster_cta_all_merge$Cluster <- ifelse(df_cluster_cta_all_merge$Cluster ==
    3, "2+3", df_cluster_cta_all_merge$Cluster)
rownames(df_cluster_cta_all_merge) <- df_cluster_cta_all_merge$Patient
p <- plot_km(survfit(Surv(OS.delay, OS.event) ~ Cluster, data = df_cluster_cta_all_merge),
    df_cluster_cta_all_merge)
p
```

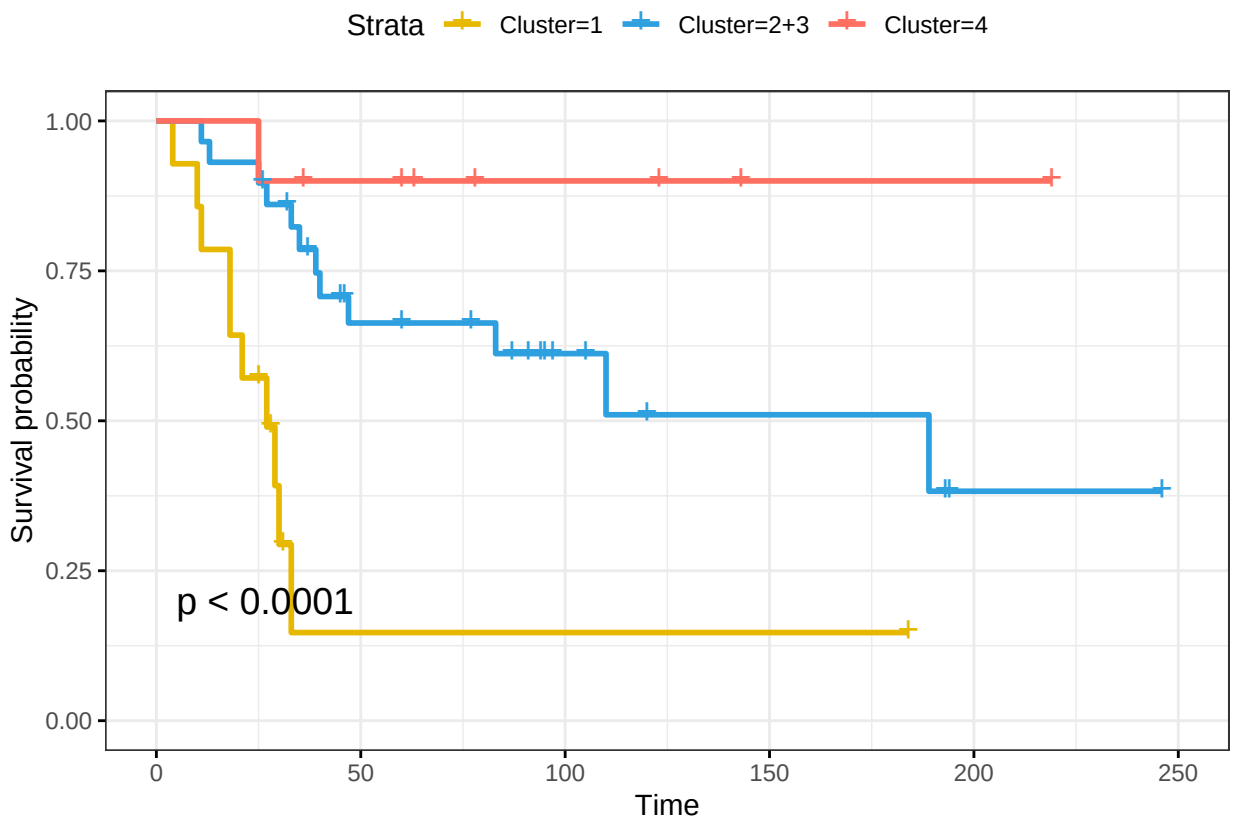

Figure 4: Kaplan-Meier plot with 3 clusters (2+3)

We see that it is significant.