Survival analysis of patients with osteosarcoma

Léa ROGUE

2025-03-13

List of Figures

Contents

Intro

Load libraries

```
library(dplyr)
library(tidyr)
library(ComplexHeatmap)
library(circlize)
library(colorRamp2)
library(RColorBrewer)
library(limma)
library(EnhancedVolcano)
library(writexl)
library(WGCNA)
```

Load data and function

The data was generated using the previous script on a virtual machine (8 CPUs, 32 GB RAM) provided by IFB Biosphere to leverage additional computational resources, as the analysis involves processing 53 files.

```
# Function to plot pca
pca_plot <- function(pca, batch, legend) {
    # Extract coo
    pca_scores <- pca$x

    # % variance explained
    var_explained <- pca$sdev^2/sum(pca$sdev^2) * 100
    pc1_var <- round(var_explained[1], 2)
    pc2_var <- round(var_explained[2], 2)

    # Plot
    plot(pca_scores[, 1], pca_scores[, 2], xlab = paste("PC1 (",
        pc1_var, "%)", sep = ""), ylab = paste("PC2 (", pc2_var,
        "%)", sep = ""), main = "PCA", pch = 19, col = batch,
        cex = 0.8)
    if (legend == TRUE) {
        legend("topright", legend = levels(batch), col = 1:length(levels(batch)),
            pch = 19)
    }
}

# Matrix with intensities and information on immune
# signature and CTA
df_expr_int <- read.table("../results/expr_matrix_int_CTA_sign_imm_clean.tsv",
    sep = "\t", header = TRUE, check.names = FALSE)
rownames(df_expr_int) = df_expr_int$SYMBOL

# Matrix with z-scores intensities and information on
# immune signature and CTA
```

```r
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)

# Metadata
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))
```

I. Relative expression of CTAs

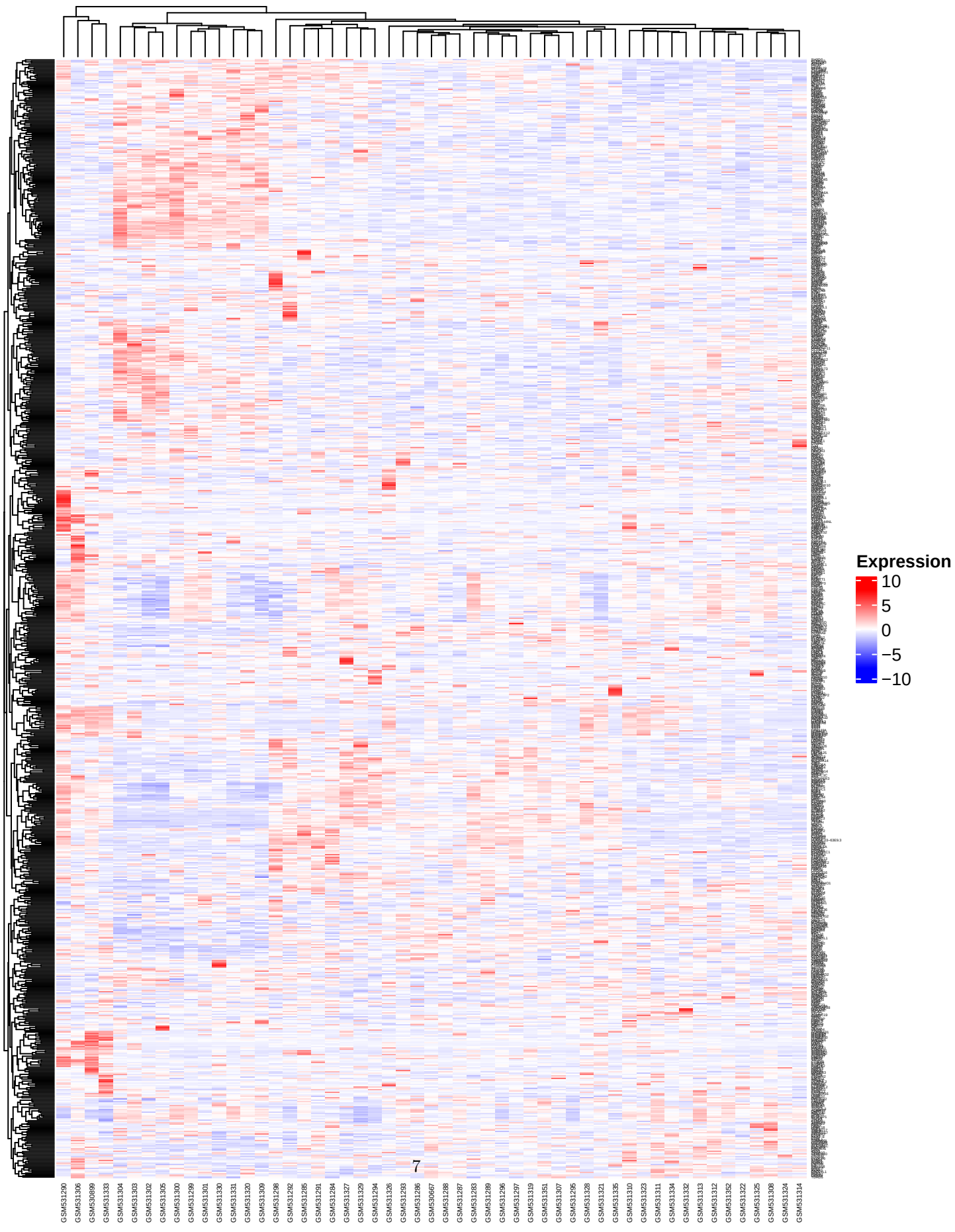
This analysis begins with the full matrix of z-scores, from which CTA genes are selected. The goal is to assess whether there are distinct patient groups based on CTA expression. ## 1) Global relative expression of CTA

```
# Prepare data to create heatmap Convert to a matrix
rownames(df_expr_z_scores) <- df_expr_z_scores$SYMBOL
df <- df_expr_z_scores %>%
    filter(CTA != "NA") %>%
    filter(!grepl("^(NA,)*NA$", CTA))

# Prepare data
expr_cta <- df %>%
    select(-SYMBOL, -CTA, -Signature)
matrix_expr_cta <- as.matrix(expr_cta)

# Create the heatmap
# pdf('../results/figures/heatmaps/heatmap_cta_all.pdf',
# height = 30)
set.seed(1)
colors <- colorRampPalette(c("blue", "white", "red"))(100)
Heatmap(matrix_expr_cta, cluster_rows = TRUE, cluster_columns = TRUE,
    cluster_column_slices = TRUE, clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete", show_column_dend = TRUE,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, show_row_names = TRUE, column_title = "Heatmap of CTA Genes",
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
```

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


Heatmap of CTA Genes
Expression
10
5
0
−5
−10
GSM531290
GSM531306
GSM530899
GSM531333
GSM531304
GSM531303
GSM531302
GSM531305
GSM531300
GSM531299
GSM531301
GSM531330
GSM531331
GSM531320
GSM531309
GSM531298
GSM531292
GSM531285
GSM531291
GSM531284
GSM531327
GSM531329
GSM531294
GSM531326
GSM531293
GSM531286
GSM530667
GSM531288
GSM531297
GSM531283
GSM531289
GSM531296
GSM531297
GSM531319
GSM531351
GSM531307
GSM531295
GSM531328
GSM531321
GSM531335
GSM531310
GSM531323
GSM531311
GSM531334
GSM531332
GSM531313
GSM531312
GSM531352
GSM531322
GSM531325
GSM531308
GSM531324
GSM531314

This heatmap reveals the variation in CTA gene expression across patients.

2) heatmap with CTA siginificant in chondro

```
# List of CTA for coxph analysis
l_CTA_conv <- read.table("../data/CTA_signif_coxph_conv_indiv.txt",
    header = FALSE)
l_CTA_conv <- l_CTA_conv$V1
data <- matrix_expr_cta[rownames(matrix_expr_cta) %in% l_CTA_conv,
    ]

# Create the heatmap
# pdf('../results/figures/heatmaps/heatmap_cta_coxph_signif_conv_indiv.pdf',
# height = 8)
heatmap_cta_signif_conv <- Heatmap(data, cluster_rows = TRUE,
    cluster_columns = TRUE, cluster_column_slices = TRUE, clustering_distance_columns = "euclidean",
    clustering_method_columns = "average", clustering_distance_rows = "euclidean",
    clustering_method_rows = "average", show_column_dend = TRUE,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, show_row_names = TRUE, column_title = "Heatmap of CTA Genes",
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif_conv <- draw(heatmap_cta_signif_conv)
```

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

3) Heatmap with CTA signif in osteo

```
# List of CTA for coxph analysis
df_CTA_osteo <- read.table("../results/results_coxph_osteo_cta_zscore_signif.tsv",
    header = T)
l_CTA_osteo <- df_CTA_osteo$Variable
data <- matrix_expr_cta[rownames(matrix_expr_cta) %in% l_CTA_osteo,
    ]

# Create the heatmap
# pdf('../results/figures/heatmaps/heatmap_cta_coxph_signif_osteo.pdf',
# height = 5)
heatmap_cta_signif <- Heatmap(data, cluster_rows = TRUE, cluster_columns = TRUE,
    cluster_column_slices = TRUE, clustering_distance_columns = "euclidean",
    clustering_method_columns = "average", clustering_distance_rows = "euclidean",
    clustering_method_rows = "average", show_column_dend = TRUE,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, show_row_names = TRUE, column_title = "Heatmap of CTA Genes",
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif <- draw(heatmap_cta_signif)
```

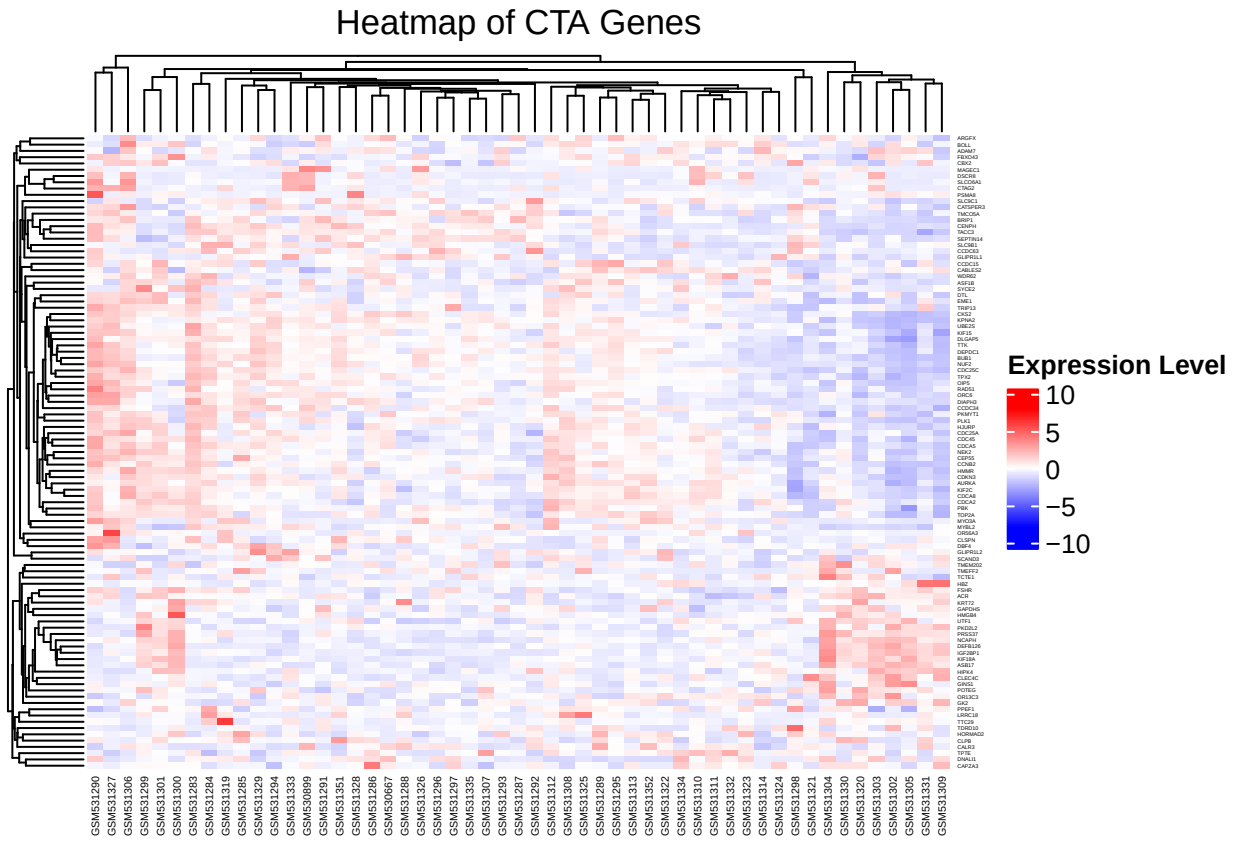


Figure 2: Heatmap of CTAs that impact survival (n = 53)

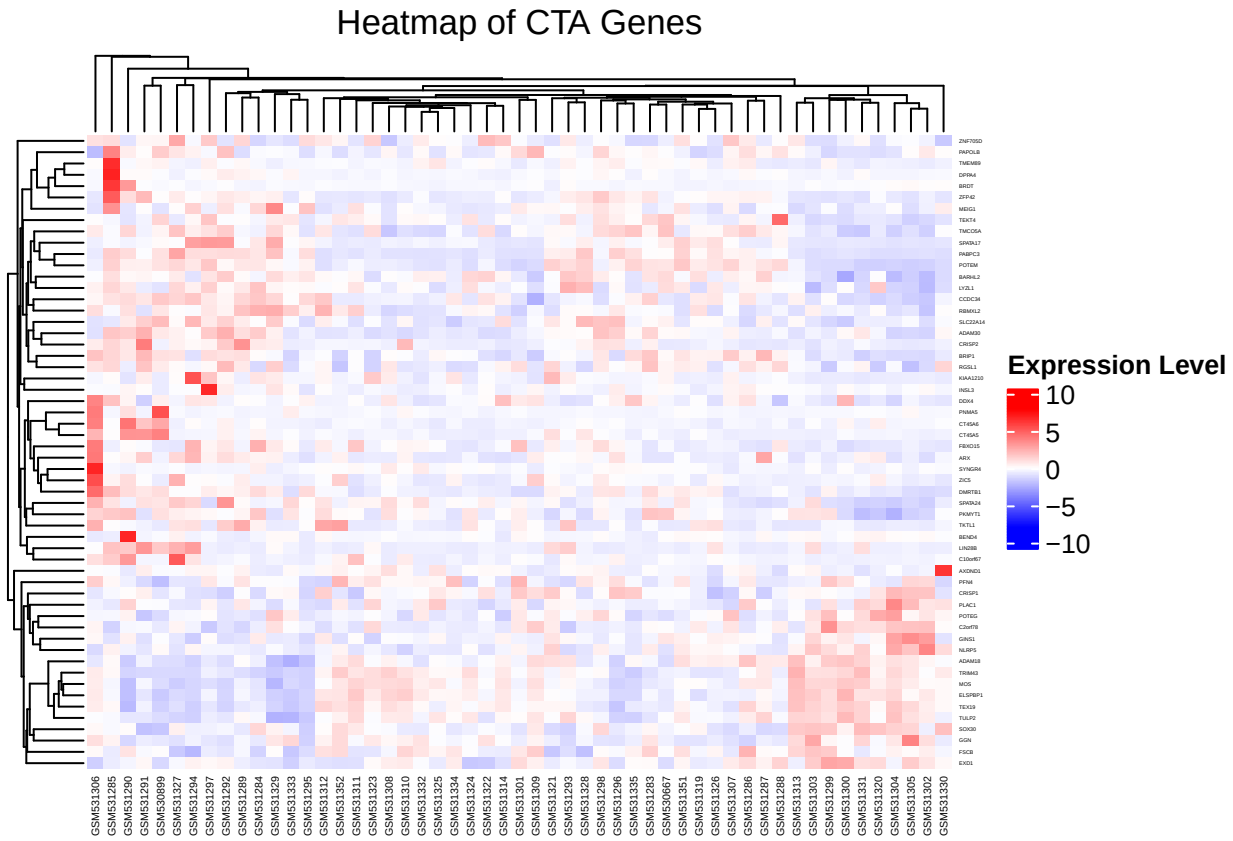


Figure 3: Heatmap of CTAs that impact survival (n = 53)

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

```r
# Take col indexes
indiv_clust <- column_order(heatmap_cta_signif)

# Create table with indiv names
df_indiv_clusters_hm <- data.frame(Cluster = c(rep(1, length(indiv_clust[1:14])),
    rep(2, length(indiv_clust[15:28])), rep(3, length(indiv_clust[29:43])),
    rep(4, length(indiv_clust[44:53]))), Patient = c(colnames(matrix_expr_cta)[indiv_clust[1:14]],
    colnames(matrix_expr_cta)[indiv_clust[15:28]], colnames(matrix_expr_cta)[indiv_clust[29:43]],
    colnames(matrix_expr_cta)[indiv_clust[44:53]]))

# Save write.table(df_indiv_clusters_hm, file =
# '../results/clusters_cta_signif_coxph_osteo_53.tsv', sep
# = '\t', quote = FALSE, row.names = FALSE)
```

```r
row_dend <- row_dend(heatmap_cta_signif)

# Cut in 2 clusters (HOT/COLD)
row_clusters <- cutree(as.hclust(row_dend), k = 2)

# Cluster annotation
df_cta_clusters_hm <- as.data.frame(row_clusters)

# Create the heatmap
Heatmap(data, cluster_rows = TRUE, cluster_columns = TRUE, row_split = row_clusters,
    cluster_column_slices = TRUE, clustering_distance_columns = "euclidean",
    clustering_method_columns = "average", clustering_distance_rows = "euclidean",
    clustering_method_rows = "average", show_column_dend = TRUE,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, show_row_names = TRUE, column_title = "Heatmap of CTA Genes",
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
```

```r
# dev.off() write.table(df_cta_clusters_hm, file =
# '../results/clusters_cta_heatmap_signif_coxph_osteo_53.tsv',
# sep = '\t', quote = FALSE)
```

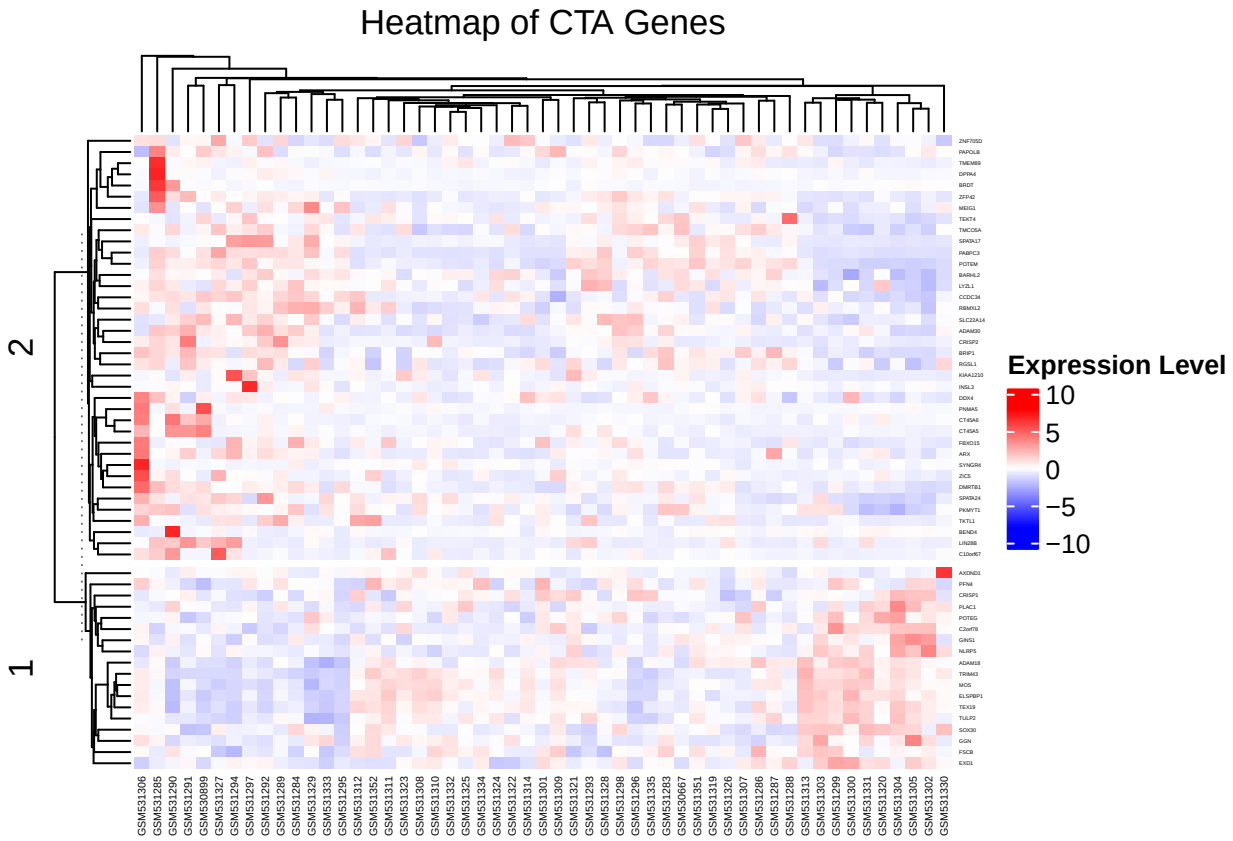


Figure 4: Heatmap of CTAs that impact survival with k = 2 (n = 53)

II. Relative immune cells expression

In this section, we aim to observe the relative expression of immune cell signatures in patients to characterize "hot" tumors, which are infiltrated by immune cells, versus "cold" tumors, which are poor in immune cells. Cold tumors are generally hard to treat and are associated with a worse prognosis. The matrix for creating these heatmaps is the average of genes per signature in z-scores for comparison. Hierarchical clustering is performed on this data.

1) Hierarchical clustering

```
# Heatmap
heatmap_data <- as.data.frame(df_imm_z_scores)
rownames(heatmap_data) <- heatmap_data$Signature
heatmap_data <- heatmap_data[, -1]  # Remove the Signature column
Heatmap(as.matrix(heatmap_data), cluster_rows = TRUE, cluster_columns = TRUE,
    cluster_column_slices = TRUE, clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete", show_column_dend = TRUE,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 7), heatmap_legend_param = list(title = "Expression Level"))
```

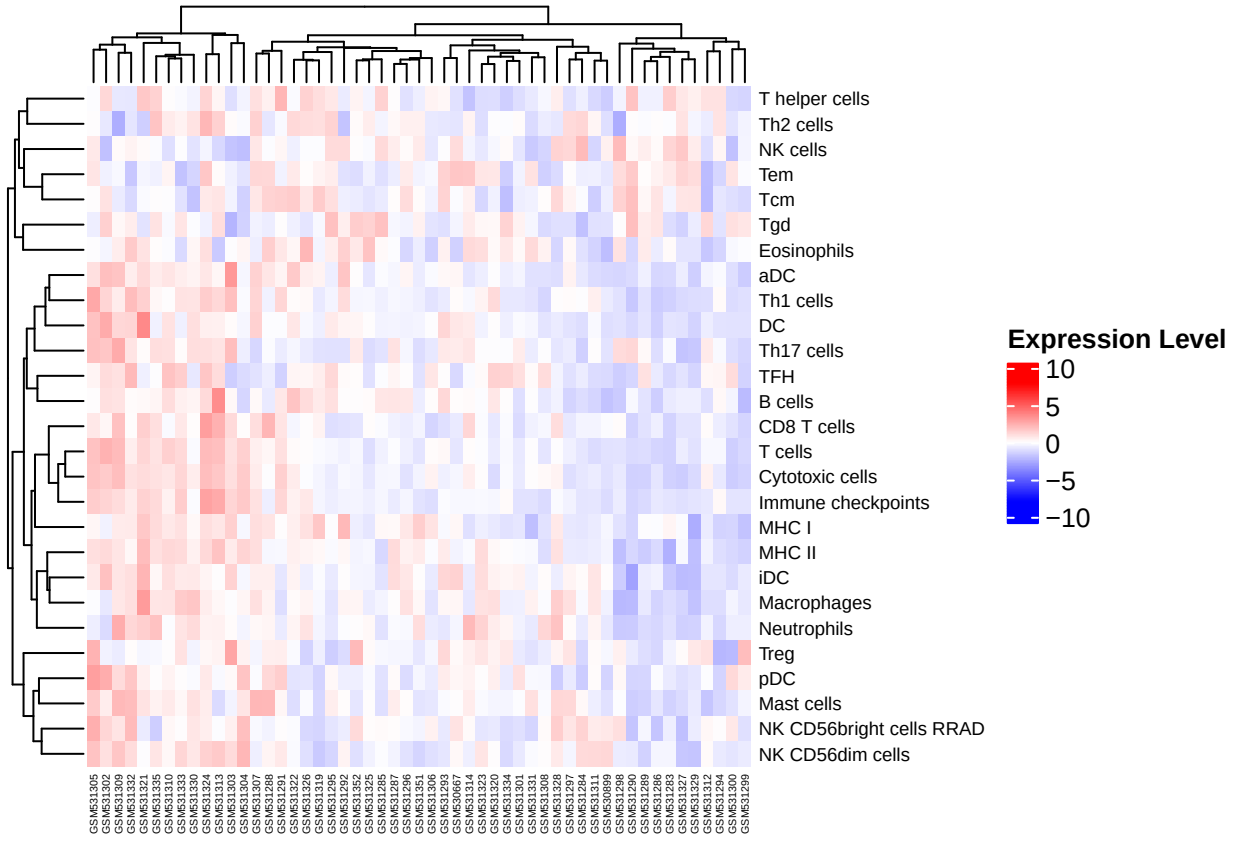


Figure 5: Heatmap and hierarchical clustering of relative immune cells expression (n = 53)

From the heatmap, we observe a separation between a "hot" side and a "cold" side, indicating that some

tumors are more infiltrated by immune cells than others. It's possible that some cells are more present than others, which could make the distinction between hot and cold tumors more apparent.

2) K-means clustering with k = 2 on patients

```r
# Elbow plot to have the number of clusters
X <- heatmap_data
# Number of clusters to test
wss <- numeric(15)

# Apply k-means
for (k in 1:15) {
    kmeans_result <- kmeans(X, centers = k, nstart = 25)
    wss[k] <- kmeans_result$tot.withinss
}

# Elbow Plot
plot(1:15, wss, type = "b", pch = 19, col = "blue", xlab = "Number of clusters (k)",
    ylab = "WSS (Within-cluster sum of squares)", main = "Elbow Plot")
```

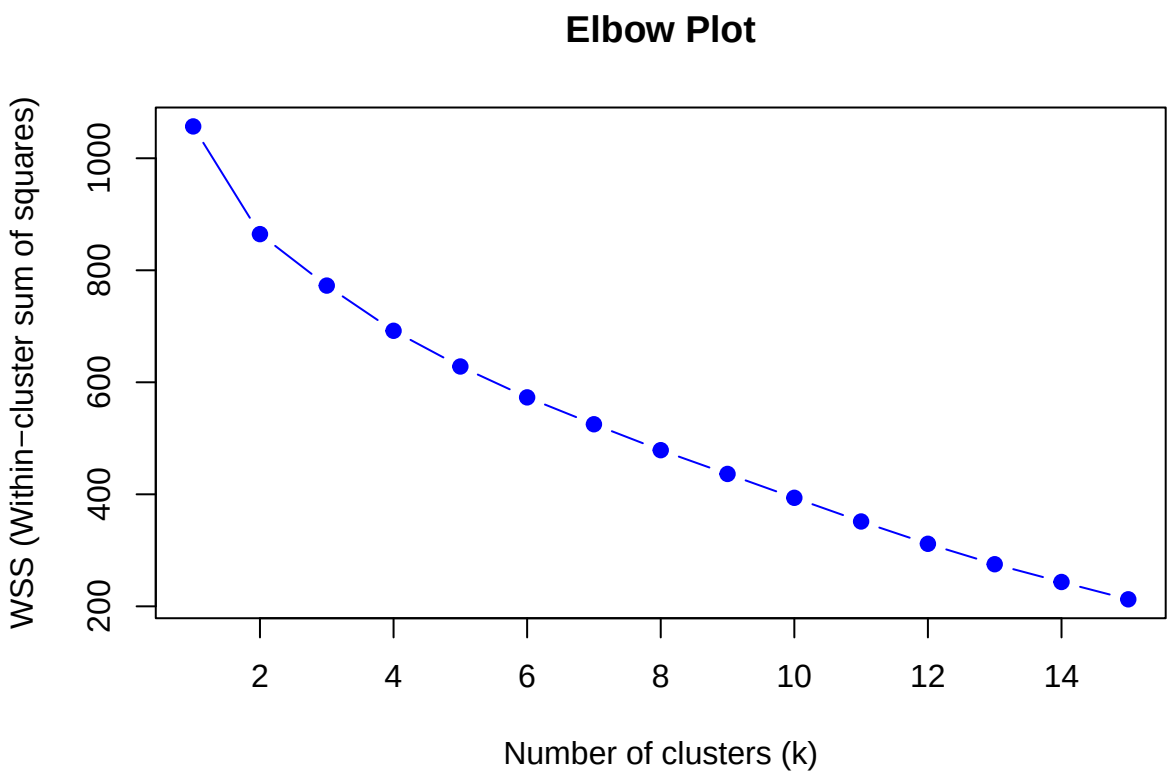


Figure 6: Elbow plot for k-means clustering

The elbow plot doesn't reveal a distinct elbow, so the number of clusters is chosen based on the scientific question. We proceed with k-means clustering with k = 2, as we are interested in distinguishing between the "cold" and "hot" clusters.

```r
# Saved pdf
#pdf("../results/figures/heatmaps/heatmap_kmeans_2.pdf", width = 8, height = 6)

# Set seed to reproducible results
set.seed(1)

# Create heatmap
heatmap <- Heatmap(
    as.matrix(heatmap_data),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete",
    show_column_dend = TRUE,
    column_km = 2,  # Nombre de clusters
    column_km_repeats = 100,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
    border = NA,
    show_column_names = TRUE,
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 7),
    heatmap_legend_param = list(title = "Expression Level")
)

# Print heatmap
set.seed(1)
heatmap = draw(heatmap)
```

```r
# Close the pdf
#dev.off()
```

3) Clustering k-means with k = 3

Given the presence of moderately infiltrated tumors, we perform k-means clustering with k = 4.

```r
#pdf("../results/figures/heatmaps/heatmap_kmeans_4.pdf", width = 8, height = 6)
set.seed(1)

# Generates heatmap
heatmap_km3 <- Heatmap(
    as.matrix(heatmap_data),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete",
    show_column_dend = TRUE,
    column_km = 3, # Number of clusters
    column_km_repeats = 20,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
```

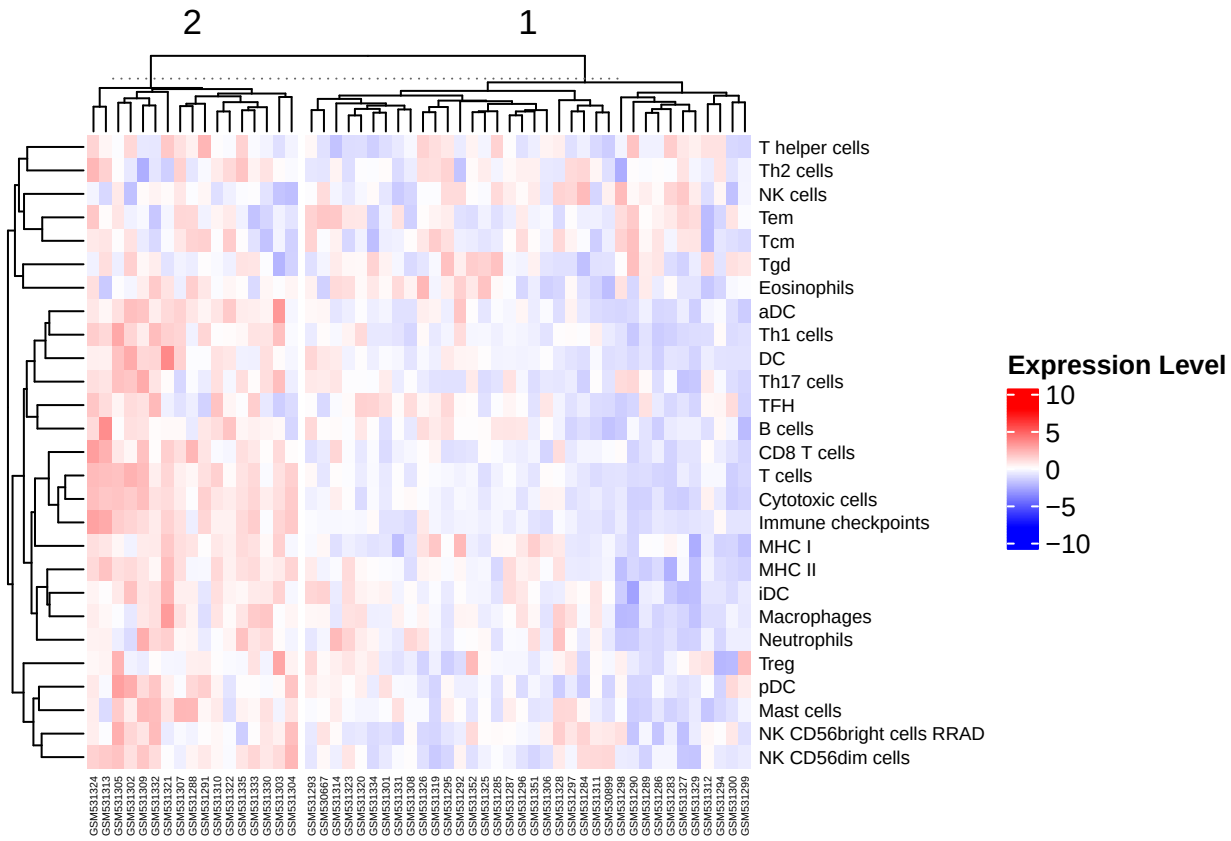


Figure 7: Heatmap with k-means clustering (k = 2) of immune cells expression (n = 53)

```
    border = NA,
    show_column_names = TRUE,
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 7),
    heatmap_legend_param = list(title = "Expression Level")
)
set.seed(1)
heatmap_km3 = draw(heatmap_km3)
```

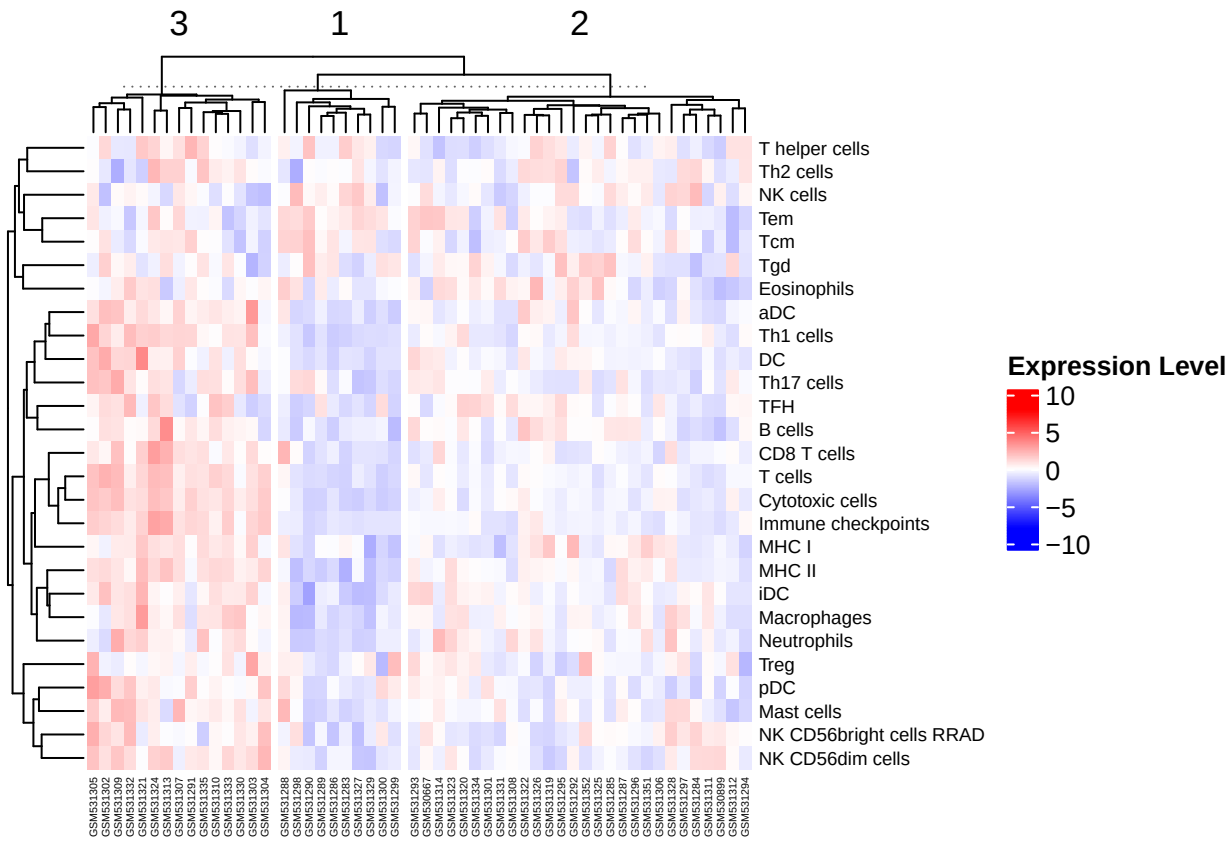


Figure 8: Heatmap with k-means clustering (k = 3) of immune cells expression (n = 53)

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

4) Clustering k-means with k = 4

Given the presence of moderately infiltrated tumors, we perform k-means clustering with k = 4.

```
#pdf("../results/figures/heatmaps/heatmap_kmeans_4.pdf", width = 8, height = 6)
set.seed(1)

# Generates heatmap
heatmap_km4 <- Heatmap(
    as.matrix(heatmap_data),
    cluster_rows = TRUE,
```

```
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete",
    show_column_dend = TRUE,
    column_km = 4, # Number of clusters
    column_km_repeats = 20,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
    border = NA,
    show_column_names = TRUE,
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 7),
    heatmap_legend_param = list(title = "Expression Level")
)
set.seed(1)
heatmap_km4 = draw(heatmap_km4)
```

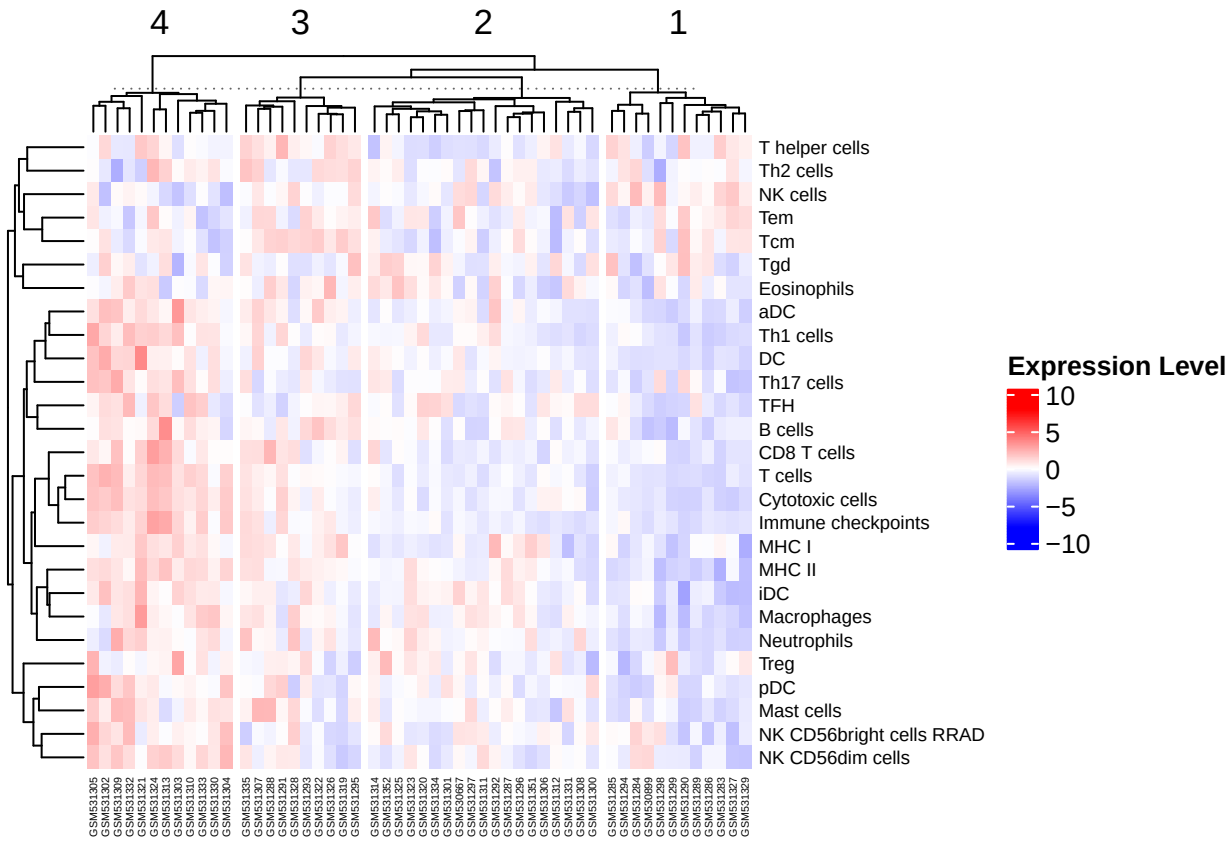


Figure 9: Heatmap with k-means clustering (k = 4) of immune cells expression (n = 53)

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

5) Adding metadata

```
# Select colors
gender_color <- c("F" = "#E74C3C",
                  "M" = "#3498DB")

histology_colors <- c("Anaplastic" = "#E74C3C",
                      "Chondroblastic" = "#3498DB",
                      "Fibroblastic" = "#2ECC71",
                      "Giant cell rich" = "#F1C40F",
                      "Osteoblastic" = "#9B59B6",
                      "Pleomorphic" = "#000000",
                      "Possibly chondromyxoid fibroma like" = "#D81B60",
                      "Sclerosing" = "#00FFFF",
                      "Telangiectatic" = "#39FF14")


# Utiliser mutate et case_when pour regrouper les catégories
df_metadata <- df_metadata %>%
  mutate(simplified_location = case_when(
    grepl("femur", tumor_location, ignore.case = TRUE) ~ "Femur",
    grepl("tibia", tumor_location, ignore.case = TRUE) ~ "Tibia",
    grepl("humerus", tumor_location, ignore.case = TRUE) ~ "Humerus",
    grepl("fibula", tumor_location, ignore.case = TRUE) ~ "Fibula",
    TRUE ~ "Unknown"  # Pour tout ce qui ne correspond pas à ces catégories
  ))

tumor_location_simplified_colors <- c("Femur" = "#E74C3C",
                                      "Fibula" = "#3498DB",
                                      "Humerus" = "#2ECC71",
                                      "Tibia" = "#F1C40F",
                                      "Unknown" = "#9B59B6")

huvos_grade_colors <- c("1" = "#E74C3C",
                        "2" = "#3498DB",
                        "3" = "#2ECC71",
                        "4" = "#F1C40F",
                        "Unknown" = "#9B59B6")

# Heatmap
#pdf("../results/figures/heatmaps/heatmap_complete_annotated.pdf", height = 10, width = 15)
set.seed(1)
heatmap_anno_all <- Heatmap(
    as.matrix(heatmap_data),
    cluster_rows = FALSE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    show_column_dend = FALSE,
    column_km = 3, # Number of clusters
    column_km_repeats = 20,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
    border = NA,
    show_column_names = TRUE,
```

```
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 7),
    heatmap_legend_param = list(title = "Expression Level"),
    top_annotation = columnAnnotation(
      Gender = df_metadata$gender,
      Histology = df_metadata$histological_subtype,
      Tumor_location = df_metadata$simplified_location,
      Huvos_grade = df_metadata$huvos_grade,
      col = list(Gender = gender_color,
                 Histology = histology_colors,
                 Tumor_location = tumor_location_simplified_colors,
                 Huvos_grade = huvos_grade_colors))
)
set.seed(1)
heatmap_anno_all <- draw(heatmap_anno_all)
```

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

```
# Heatmap
#pdf("../results/figures/heatmaps/heatmap_complete_annotated.pdf", height = 10, width = 15)
set.seed(1)
heatmap_anno_all <- Heatmap(
    as.matrix(heatmap_data),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    show_column_dend = FALSE,
    #column_km = 3, # Number of clusters
    #column_km_repeats = 20,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
    border = NA,
    show_column_names = TRUE,
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 7),
    heatmap_legend_param = list(title = "Expression Level"),
    top_annotation = columnAnnotation(
      Gender = df_metadata$gender,
      Histology = df_metadata$histological_subtype,
      Tumor_location = df_metadata$simplified_location,
      Huvos_grade = df_metadata$huvos_grade,
      col = list(Gender = gender_color,
                 Histology = histology_colors,
                 Tumor_location = tumor_location_simplified_colors,
                 Huvos_grade = huvos_grade_colors))
)
set.seed(1)
heatmap_anno_all <- draw(heatmap_anno_all)
```

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

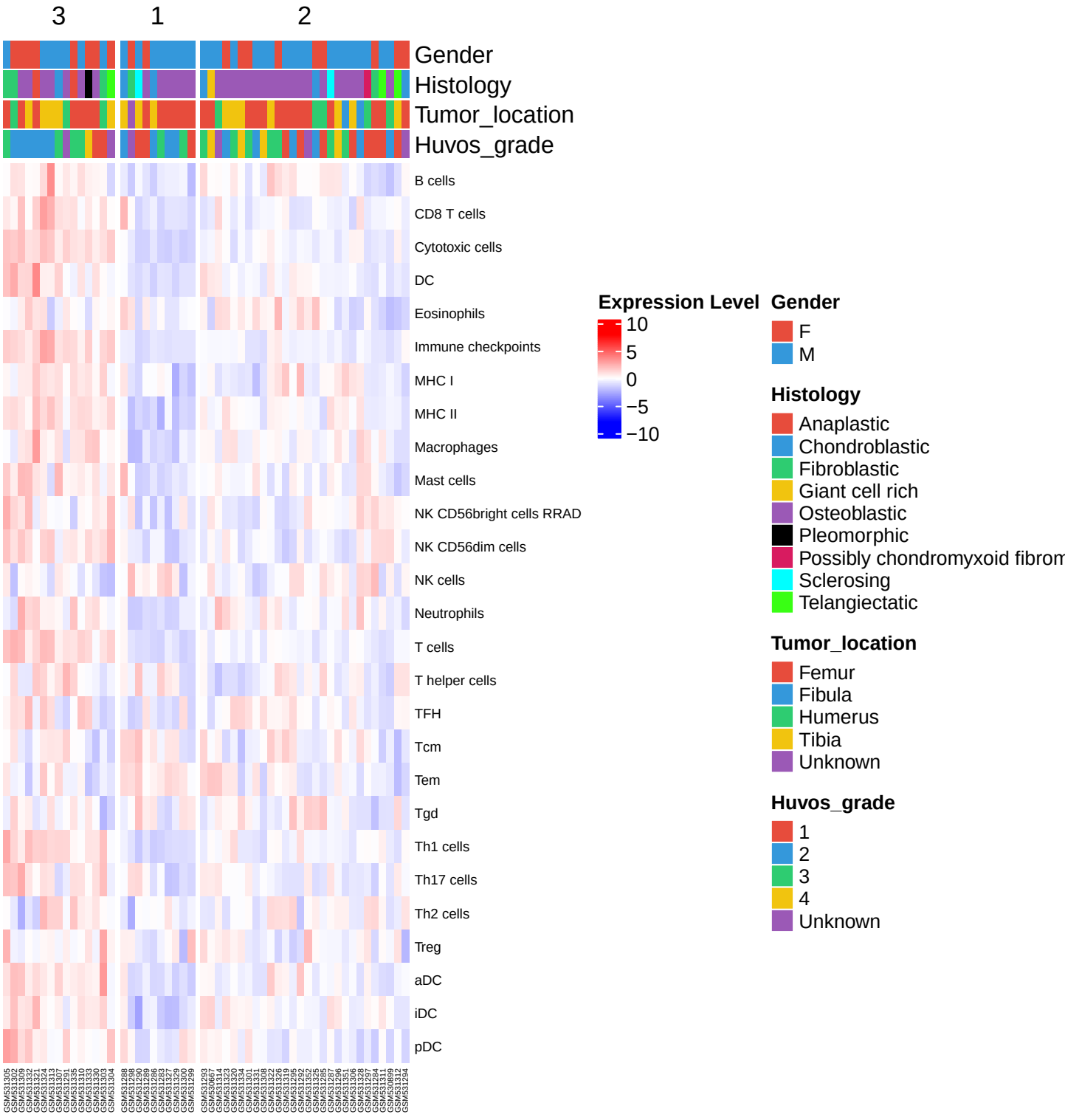


Figure 10: Heatmap with metadata (n = 53)

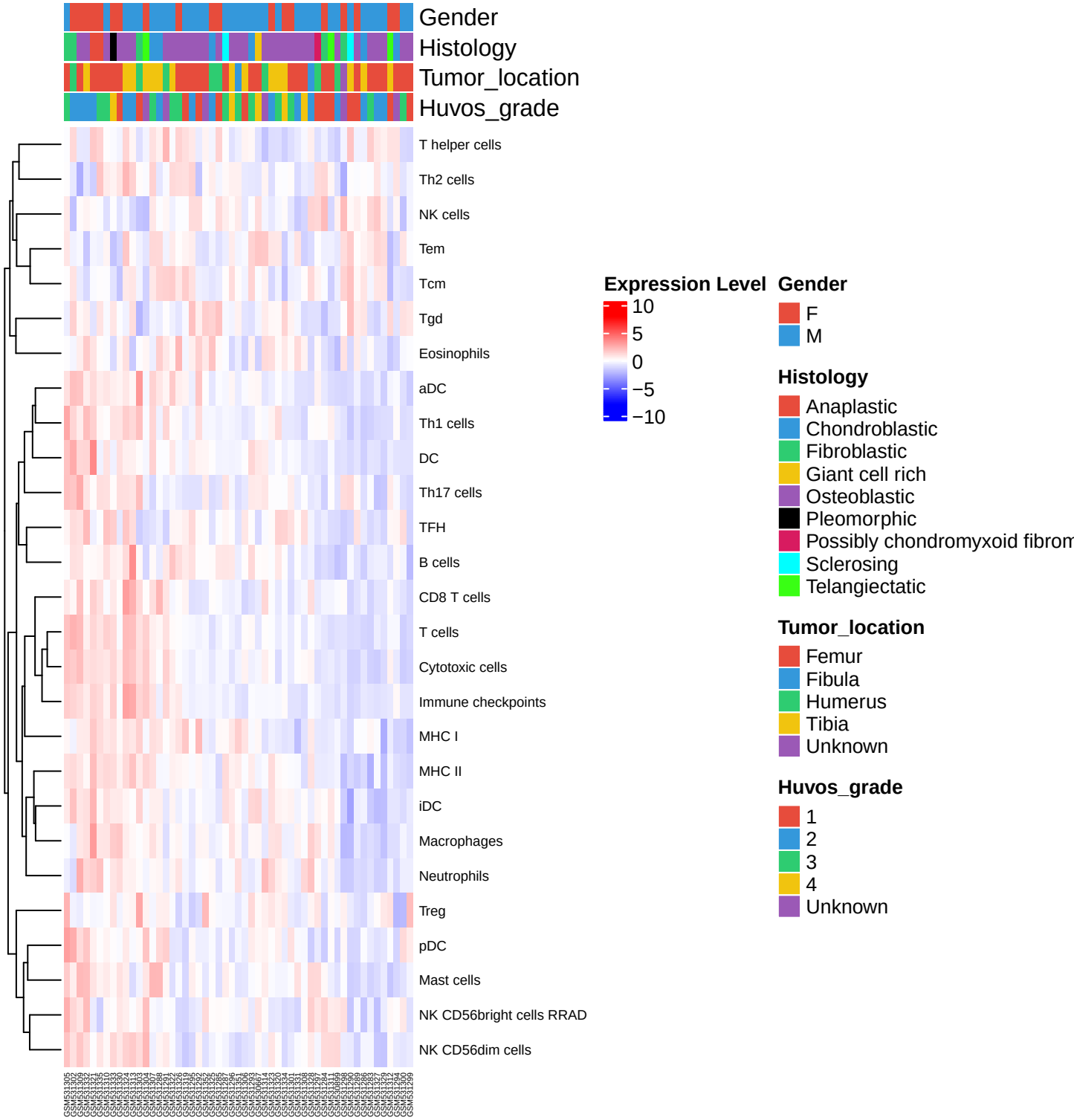


Figure 11: Heatmap with metadata (n = 53)

III. Exploring the relationship between the expression of CTAs and immune cell infiltration

Thanks to the survival analysis and this expression analysis, we can search if a link exists between CTA expression and immune cell infiltration in tumors.

1) Visual exploration

This section generates heatmaps by crossing CTA expression and immune cell types expression

a- Hierarchical clustering

```
# CTA heatmap
ht_cta <- Heatmap(t(data),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete",
    show_column_dend = TRUE,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
    border = NA,
    show_column_names = TRUE,
    show_row_names = TRUE,
    column_title = "Heatmap of CTA Genes",
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 4),
    heatmap_legend_param = list(title = "Expression Level")
)

# Kmeans heatmap
ht_k2_conv <- Heatmap(
    as.matrix(t(heatmap_data)),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete",
    show_column_dend = TRUE,
    row_km = 2, # Number of clusters
    row_km_repeats = 20,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
    border = NA,
    show_column_names = TRUE,
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 4),
    heatmap_legend_param = list(title = "Expression Level")
)
ht_cta + ht_k2_conv
```

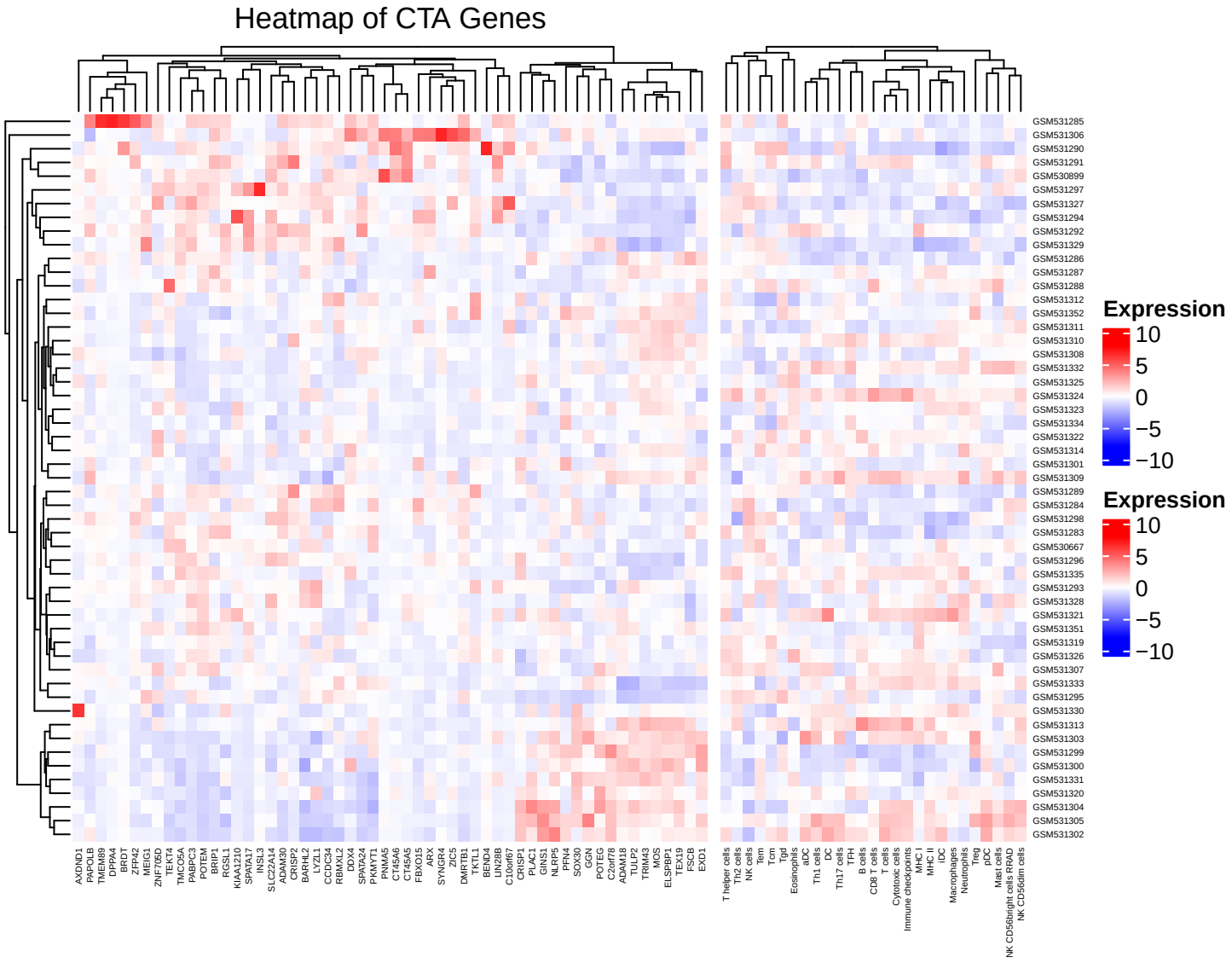


Figure 12: Heatmap of CTA that impact survival analysis and immune cells expression (n = 53)

This heatmap show that the cluster with an over expression of some CTA are from many patients that have the lowest immune cell types expression. So, visually, we can see a link between the 2 analysis.

2) Correlation

```
rownames(df_imm_z_scores) <- df_imm_z_scores$Signature
df_imm <- df_imm_z_scores[, -1]
cor_matrix <- cor(t(df_imm), t(matrix_expr_cta), method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, 63)

# Significant p-val
pval_signif <- cor_pval < 0.05

# Heatmap
# pdf('../results/figures/correlation_matrix_all_CTA_osteo.pdf',
# height = 6, width = 20)
colors <- colorRampPalette(c("#75E05A", "white", "#EA4343"))(100)
heatmap_all_cta_corr_osteo <- Heatmap(cor_matrix, cluster_rows = TRUE,
    cluster_columns = TRUE, cluster_column_slices = TRUE, clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete", show_column_dend = TRUE,
    column_names_gp = gpar(fontsize = 2), row_names_gp = gpar(fontsize = 5),
    col = colorRamp2(seq(-1, 1, length.out = 100), colors), heatmap_legend_param = list(title = "Pearso
    cell_fun = function(j, i, x, y, width, height, fill) {
        if (pval_signif[i, j]) {
            grid.text("*", x, y, gp = gpar(fontsize = 8, col = "black"))
        }
    })
```

```
## Warning: You defined `cell_fun` for a heatmap with more than 100 rows or
## columns, which might be very slow to draw. Consider to use the
## vectorized version `layer_fun`.
```

```
heatmap_all_cta_corr_osteo <- draw(heatmap_all_cta_corr_osteo)
```

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

```
# Store clusters HOT and COLD
col_dend <- column_dend(heatmap_all_cta_corr_osteo)

# Cut in 2 clusters (HOT/COLD)
col_clusters <- cutree(as.hclust(col_dend), k = 2)

# Cluster annotation
df_cta_clusters_hm <- as.data.frame(col_clusters)
df_cta_clusters_hm$Cluster <- ifelse(col_clusters == 1, "HOT",
    "COLD")


# pdf('../results/figures/correlation_matrix_all_CTA_osteo_2_clust.pdf',
# height = 6, width = 20)
```

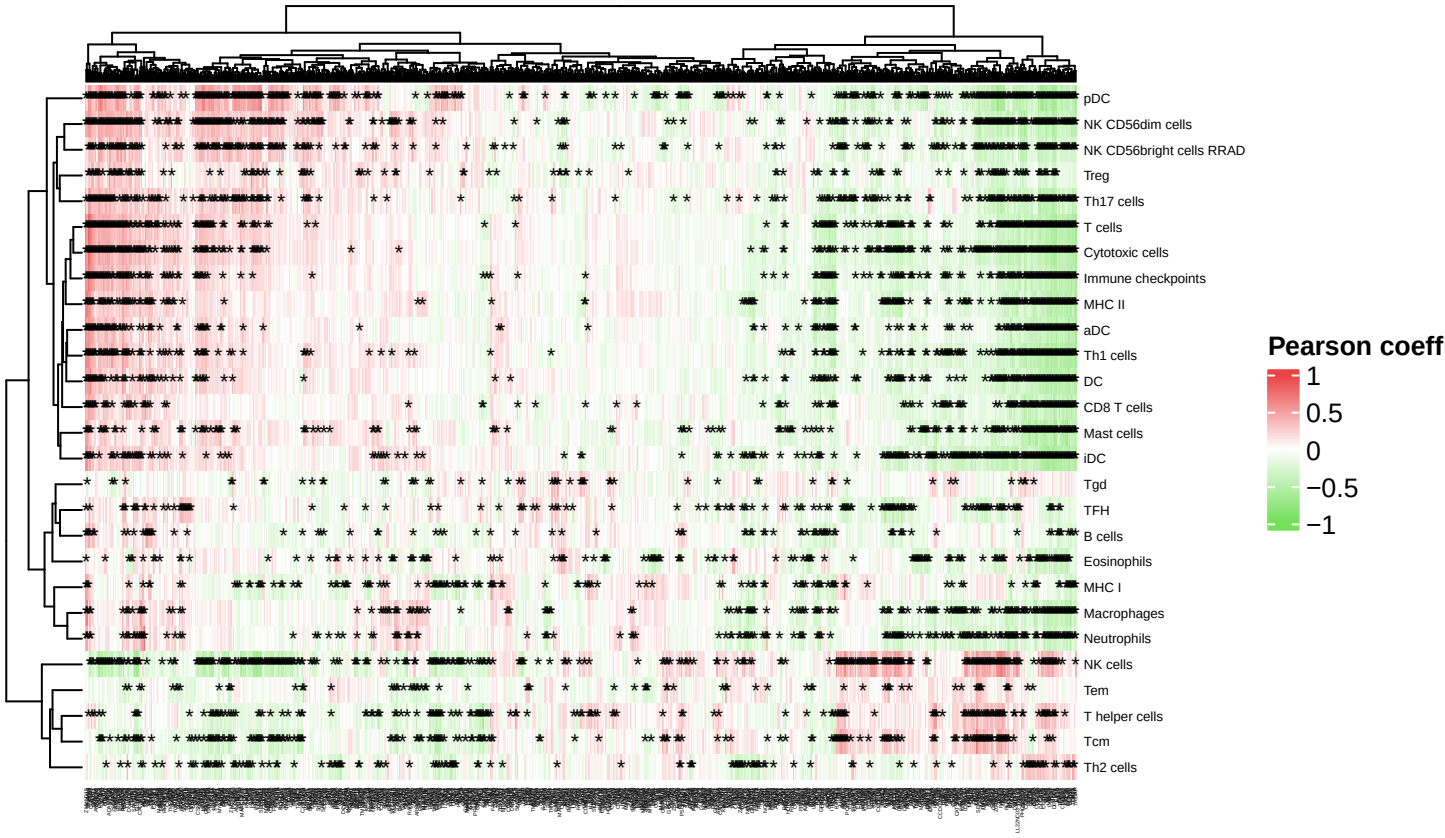


Figure 13: Correlation heatmap (n = 53)

```
heatmap_corr_cta_k2 <- Heatmap(cor_matrix, cluster_rows = TRUE,
    cluster_columns = TRUE, column_split = col_clusters, cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean", clustering_method_columns = "complete",
    show_column_dend = TRUE, column_names_gp = gpar(fontsize = 2),
    row_names_gp = gpar(fontsize = 5), col = colorRamp2(seq(-1,
        1, length.out = 100), colors), heatmap_legend_param = list(title = "Pearson coefficient"),
    cell_fun = function(j, i, x, y, width, height, fill) {
        if (pval_signif[i, j]) {
            grid.text("*", x, y, gp = gpar(fontsize = 8, col = "black"))
        }
    })
```

```
## Warning: You defined `cell_fun` for a heatmap with more than 100 rows or
## columns, which might be very slow to draw. Consider to use the
## vectorized version `layer_fun`.
```

```
heatmap_corr_cta_k2 <- draw(heatmap_corr_cta_k2)
```

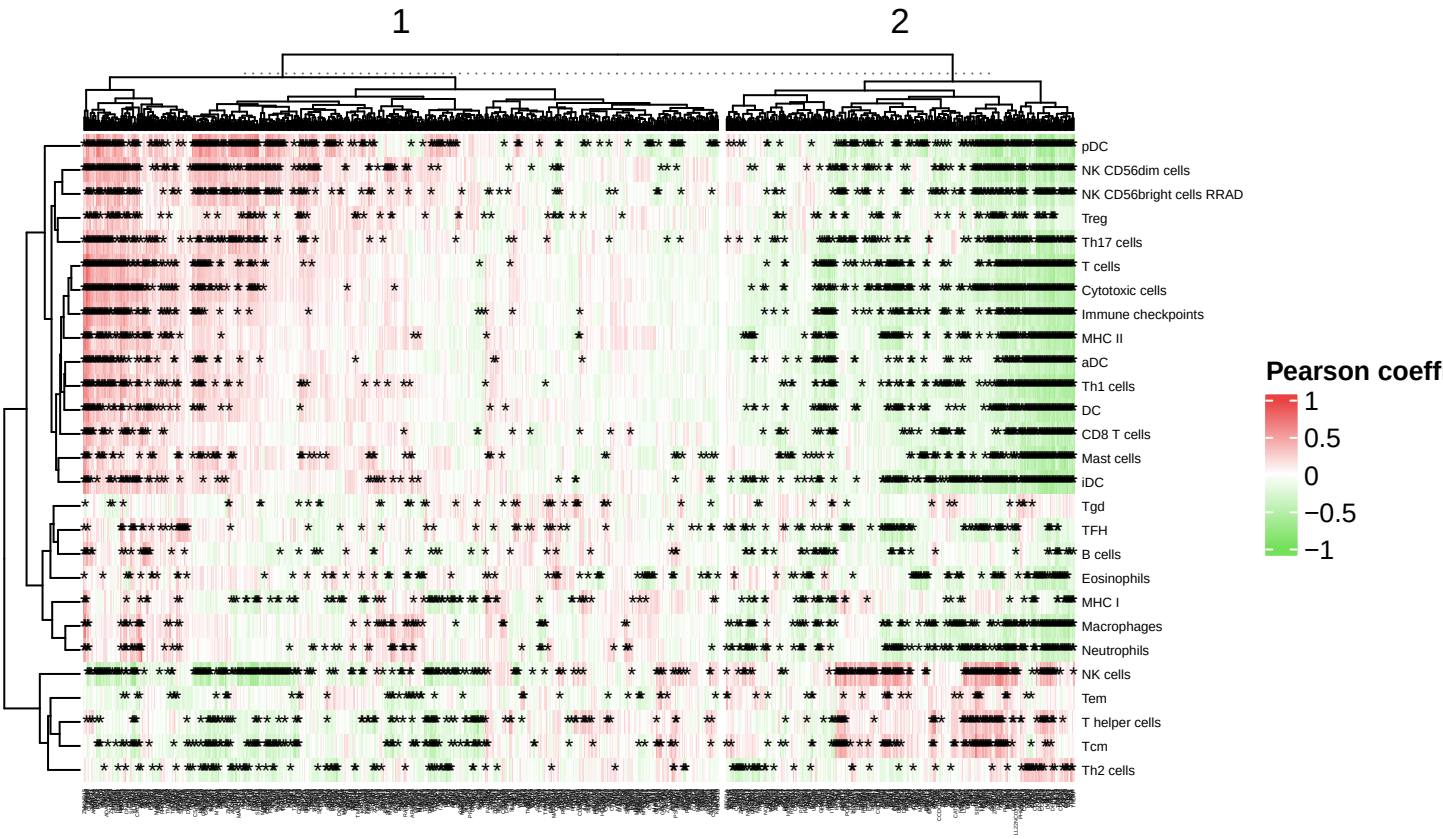


Figure 14: Correlation heatmap k = 2 (n = 53)

```
# dev.off() Save write.table(df_cta_clusters_hm, file =
# '../results/clusters_cta_pearson_53_k2.tsv', sep = '\t',
# quote = FALSE)
```

```r
# Store clusters HOT and COLD
col_dend <- column_dend(heatmap_all_cta_corr_osteo)

# Cut in 2 clusters (HOT/COLD)
col_clusters <- cutree(as.hclust(col_dend), k = 3)

# Cluster annotation
df_cta_clusters_hm <- as.data.frame(col_clusters)
df_cta_clusters_hm$Cluster <- ifelse(col_clusters == 1, "HOT",
    "COLD")

# pdf('../results/figures/correlation_matrix_all_CTA_osteo_3_clust.pdf',
# height = 6, width = 20)
Heatmap(cor_matrix, cluster_rows = TRUE, cluster_columns = TRUE,
    column_split = col_clusters, cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean", clustering_method_columns = "complete",
    show_column_dend = TRUE, column_names_gp = gpar(fontsize = 2),
    row_names_gp = gpar(fontsize = 5), col = colorRamp2(seq(-1,
        1, length.out = 100), colors), heatmap_legend_param = list(title = "Pearson coefficient"),
    cell_fun = function(j, i, x, y, width, height, fill) {
        if (pval_signif[i, j]) {
            grid.text("*", x, y, gp = gpar(fontsize = 8, col = "black"))
        }
    })
```

```
## Warning: You defined `cell_fun` for a heatmap with more than 100 rows or
## columns, which might be very slow to draw. Consider to use the
## vectorized version `layer_fun`.
```

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

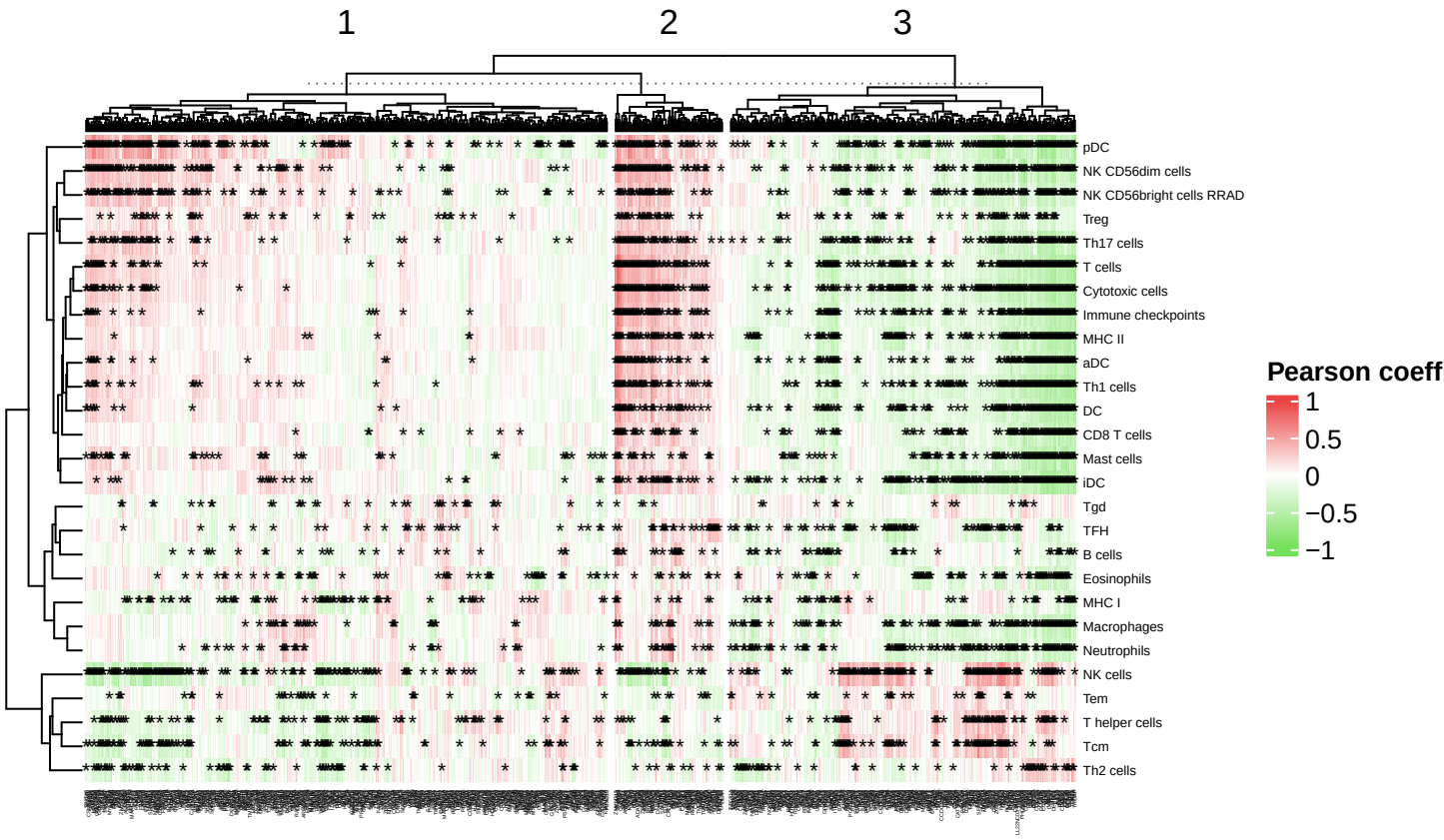


Figure 15: Correlation heatmap k = 3 (n = 53)