Chondrosarcoma expression analysis

Léa ROGUE

2025-02-04

List of Figures

Contents

This script focuses on analyzing the relative expression of Cancer-Testis Antigen (CTA) genes and immune cell expression within the E-MTAB-7264 dataset of chondrosarcoma tumors from 102 patients.

The analysis is expanded by integrating additional data on Major Histocompatibility Complex (MHC) genes. The analysis is then focus on conventional chondrosarcoma. Metadata are incorporated, and a Differentially Expressed Genes (DEG) analysis is performed to compare tumor categories. Furthermore, we investigate DEG differences between histological subtypes. The script combines heatmap visualizations to explore the relationship between CTAs that impact patient survival and the presence of immune cells. Lastly, a Weighted Correlation Network Analysis (WGCNA) is conducted to assess the correlation between CTA expression and immune cell abundance.

Load libraries

```r
library(dplyr)
library(tidyr)
library(ComplexHeatmap)
library(circlize)
library(colorRamp2)
library(RColorBrewer)
library(limma)
library(EnhancedVolcano)
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 102 files.

```r
# 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)
    }
}

# Function to calculate Z scores
```

```r
calculate_z_scores <- function(df_input, col) {
    # Exclude col PROBEID SYMBOL and CTA
    data_values <- df_input[, -c(col)]

    # Calculate Z-scores
    z_scores_row <- t(scale(t(data_values)))

    # Add columns
    df_z_scores <- cbind(df_input[, c(col)], z_scores_row)

    # Return df
    return(df_z_scores)
}

# Matrix with intensities and information on immune
# signature and CTA
df_CTA_immune_whole_clean_avg_102 <- read.table("../results/whole_gene_int_CTA_sign_imm_clean.tsv",
    sep = "\t", header = TRUE, check.names = FALSE, row.names = 1)

# Matrix with z-scores intensities and information on
# immune signature and CTA
df_CTA_immune_whole_clean_z_scores_102 <- read.table("../results/whole_gene_CTA_sign_imm_clean_avg_z_sc
    sep = "\t", header = TRUE, check.names = FALSE)

# Matrix with avg on genes per signature in zscores
df_avg_immune_sign_z_scores_102 <- read.table("../results/imm_sign_avg_z_scores.tsv",
    sep = "\t", header = TRUE, check.names = FALSE)

# Metadata
df_metadata <- read.table("../results/metadata.tsv", sep = "\t",
    header = TRUE, check.names = FALSE, dec = ",")

# Conventional chondrosarcoma metadata
df_metadata_conv <- df_metadata[df_metadata$Histology != "N/A" &
    df_metadata$Histology != "benign" & df_metadata$Histology !=
    "dedifferentiated", ]
patients_conv <- df_metadata_conv$Patient
df_metadata_surv_conv <- df_metadata_conv[, c("Patient", "OS.delay",
    "OS.event")]
df_metadata_surv_conv <- na.omit(df_metadata_surv_conv)

# Metadata survival all individuals
df_metadata_surv_all <- df_metadata[, c("Patient", "OS.delay",
    "OS.event")]
df_metadata_surv_all <- na.omit(df_metadata_surv_all)

# Matrix for all patients with survival metadata
df_CTA_immune_whole_clean_avg_82 <- df_CTA_immune_whole_clean_avg_102[,
    c("Signature", "CTA", colnames(df_CTA_immune_whole_clean_avg_102)[colnames(df_CTA_immune_whole_clea
        df_metadata_surv_all$Patient])]

# Z scores matrix for all patients with survival metadata
df_z_scores_82 <- calculate_z_scores(df_CTA_immune_whole_clean_avg_82,
```

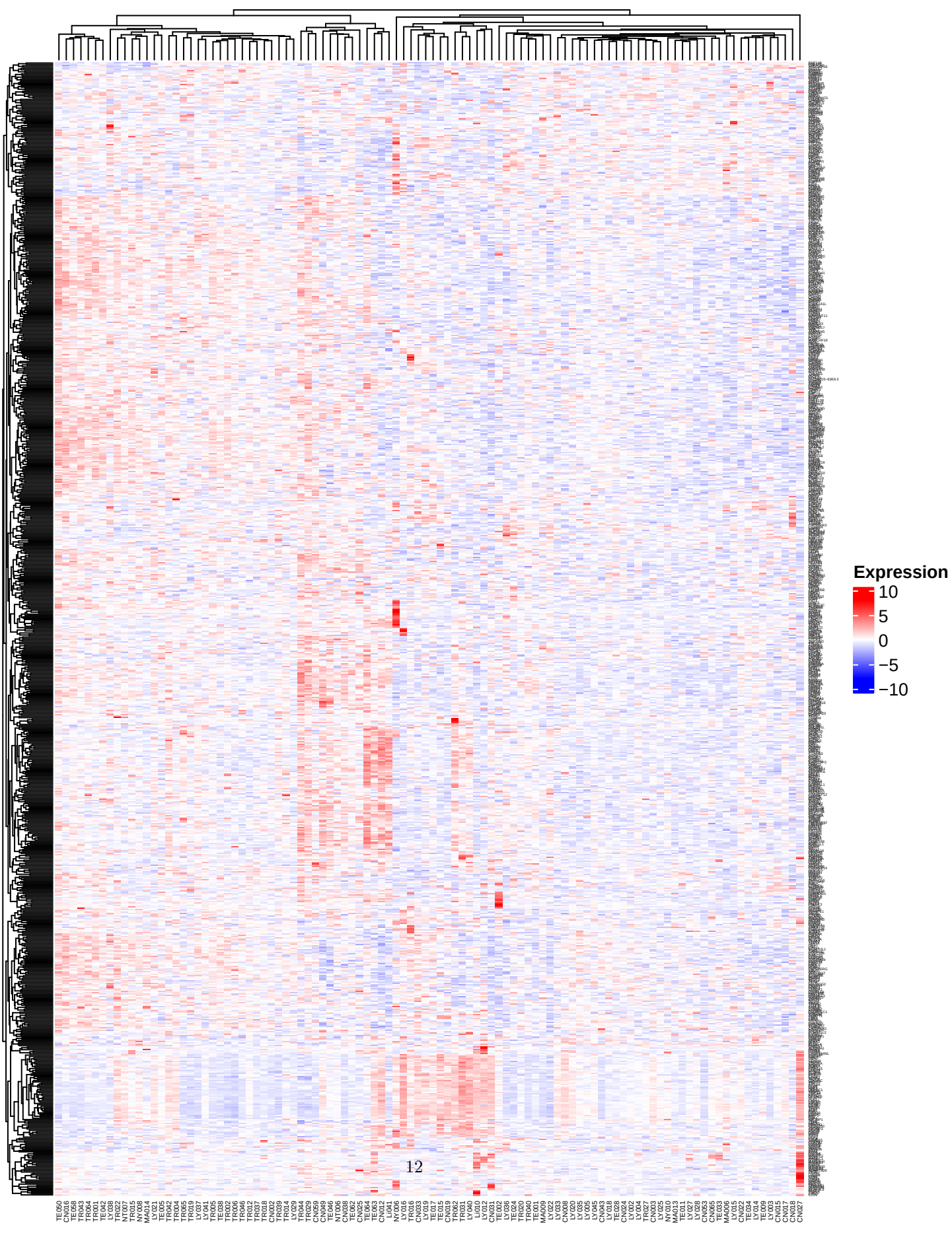
```
    c(1, 2))

# Matrix for conventional patients
df_CTA_immune_whole_clean_avg_73 <- df_CTA_immune_whole_clean_avg_102[,
    c("Signature", "CTA", colnames(df_CTA_immune_whole_clean_avg_102)[colnames(df_CTA_immune_whole_clean
        patients_conv])]
df_z_scores_73 <- calculate_z_scores(df_CTA_immune_whole_clean_avg_73,
    c(1, 2))

# Matrix for conventional patients with survival metadata
df_CTA_immune_whole_clean_avg_63 <- df_CTA_immune_whole_clean_avg_102[,
    c("Signature", "CTA", colnames(df_CTA_immune_whole_clean_avg_102)[colnames(df_CTA_immune_whole_clean
        df_metadata_surv_conv$Patient])]
df_z_scores_63 <- calculate_z_scores(df_CTA_immune_whole_clean_avg_63,
    c(1, 2))
```

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_CTA_immune_whole_clean_z_scores_102) <- df_CTA_immune_whole_clean_z_scores_102$SYMBOL
df <- df_CTA_immune_whole_clean_z_scores_102 %>%
    filter(CTA != "NA") %>%
    filter(!grepl("^(NA,)*NA$", CTA))

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

# Create the heatmap
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 all CTA",
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
```


Heatmap of all CTA
Expression
10
5
0
−5
−10

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

2) Relative expression of CTA associated with survival probability across all individuals

Here, I focus on the CTA genes that were found to be significant in a Cox proportional hazards model for all individuals (see script 6). These CTA genes influence the Hazard Ratio (HR), which reflects survival probabilities. An HR > 1 indicates an increased risk of death, meaning higher expression of the CTA gene corresponds to a higher probability of death, and vice versa.

```
# Read CTA list from files
l_CTA_all <- read.table("../data/CTA_signif_coxph_all_indiv.txt",
    header = FALSE)
l_CTA_all <- l_CTA_all$V1
data <- matrix_expr_cta[rownames(matrix_expr_cta) %in% l_CTA_all,
    ]

# Create the heatmap
heatmap_cta_signif_all <- 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, show_row_names = TRUE, column_title = "Heatmap of CTA with significant su
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif_all <- draw(heatmap_cta_signif_all)
```

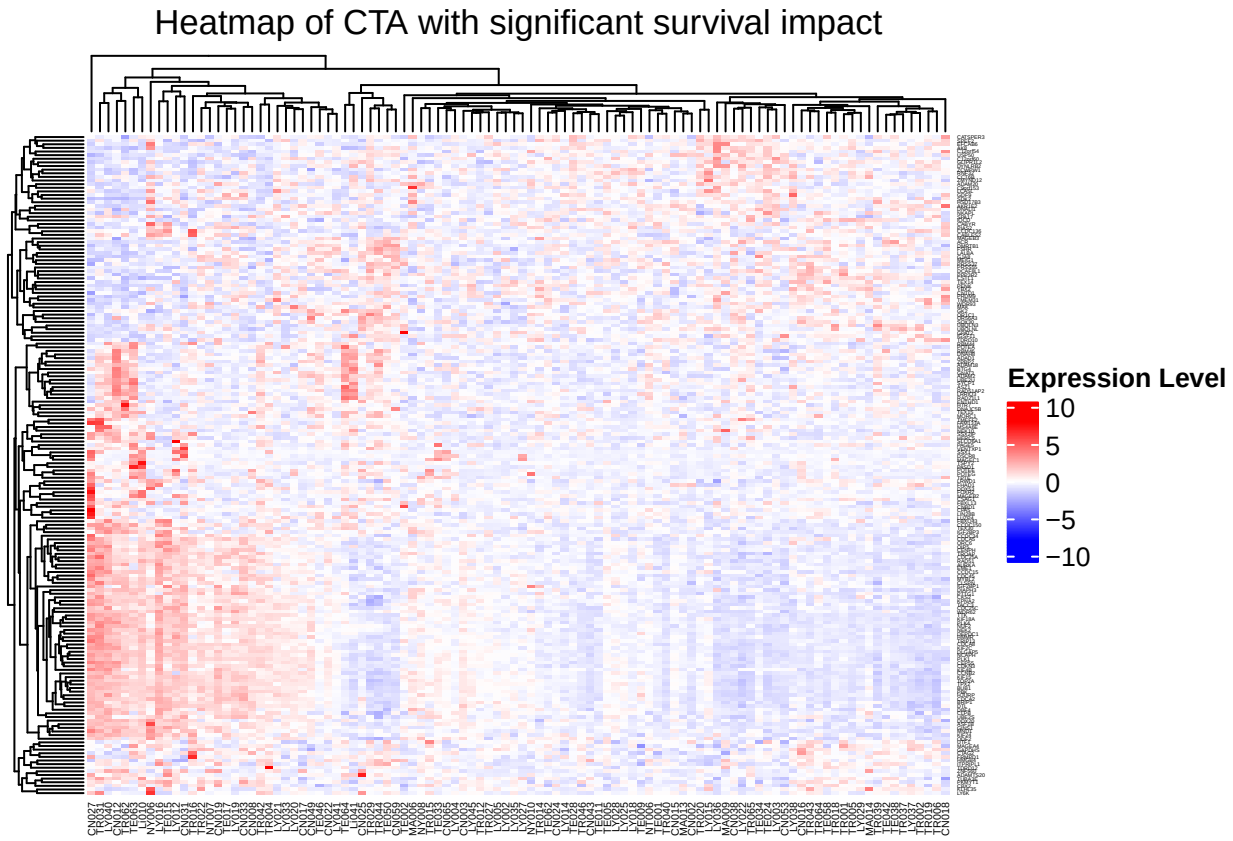


Figure 2: Heatmap of CTAs that impact survival (n = 102)

3) Relative expression of CTA associated with survival probability across all individuals (with survival metadata)

Here, I focus on the CTA genes that were found to be significant in a Cox proportional hazards model for all individuals (see script 6). These CTA genes influence the Hazard Ratio (HR), which reflects survival probabilities. An HR > 1 indicates an increased risk of death, meaning higher expression of the CTA gene corresponds to a higher probability of death, and vice versa.

```
data_all_82 <- df_z_scores_82[rownames(df_z_scores_82) %in% l_CTA_all,
    -c(1, 2)]

# Create the heatmap
heatmap_cta_signif_all_82 <- Heatmap(as.matrix(data_all_82),
    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 with significant survival impact",
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif_all_82 <- draw(heatmap_cta_signif_all_82)
```

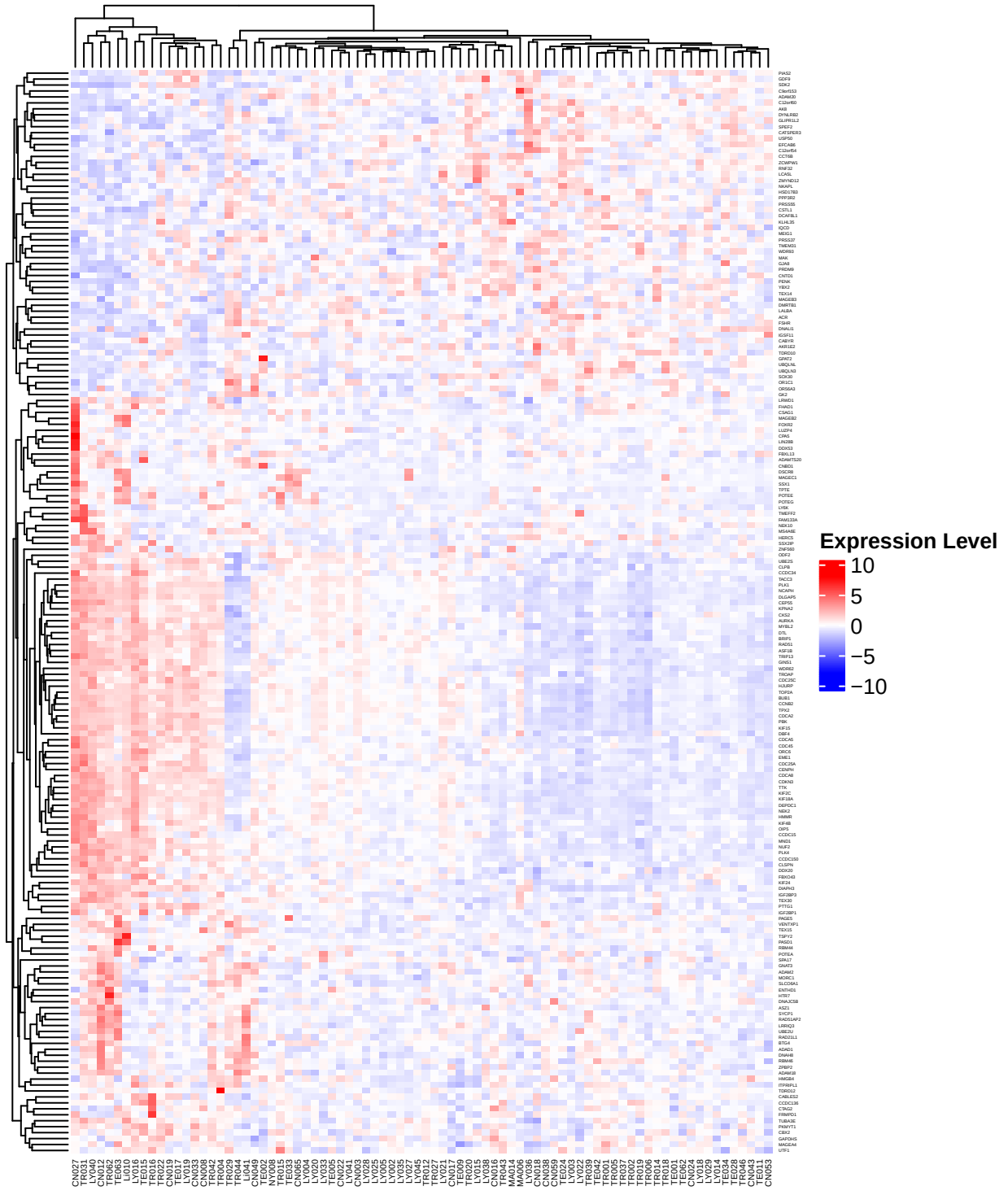


Figure 3: Heatmap of CTAs that impact survival (n = 82)

The heatmap reveals three distinct clusters of patients, which I further investigate in the survival analysis (see script 6) to determine if survival probabilities differ between these clusters.

```
# Store clusters for survival analysis in script 6 Take col
# indexes
indiv_clust <- column_order(heatmap_cta_signif_all_82)

# Create table with indiv names
df_indiv_clusters_hm <- data.frame(Cluster = c(rep(1, length(indiv_clust[1:18])),
    rep(2, length(indiv_clust[19:53])), rep(3, length(indiv_clust[54:82]))),
    Patient = c(colnames(data_all_82)[indiv_clust[1:18]], colnames(data_all_82)[indiv_clust[19:54]],
        colnames(data_all_82)[indiv_clust[55:82]]))

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

# Store cluster to analyze jitter plot in script 6 Take row
# indexes
cta_clust <- row_order(heatmap_cta_signif_all_82)

# Create table with cta
df_cta_clusters_hm <- data.frame(Cluster = c(rep(1, length(cta_clust[1:55])),
    rep(2, length(cta_clust[56:142])), rep(3, length(cta_clust[143:182]))),
    SYMBOL = c(rownames(data_all_82)[cta_clust[1:55]], rownames(data_all_82)[cta_clust[56:142]],
        rownames(data_all_82)[cta_clust[143:182]]))
# Save write.table(df_cta_clusters_hm, file =
# '../results/clusters_indiv/clusters_cta_signif_coxph_all_indiv.tsv',
# sep = '\t', quote = FALSE, row.names = FALSE)
```

a- Relative expression of selected CTA

```
l <- c("PAGE5", "VENTXP1", "TEX15", "TSPY2", "PASD1", "RBM44",
    "POTEA", "SPA17", "GNAT3", "ADAM2", "MORC1", "SLCO6A1", "ENTHD1",
    "HTR7", "DNAJC5B", "ASZ1", "SYCP1", "RAD51AP2", "LRRIQ3",
    "UBE2U", "RAD21L1", "BTG4", "ADAD1", "DNAH8", "RBM46", "ZPBP2",
    "ADAM18", "HMGB4", "ITPRIPL1", "TDRD12", "CABLES2", "CCDC136",
    "CTAG2", "FRMPD1", "TUBA3E", "PKMYT1", "CBX2", "GAPDHS",
    "MAGEA4", "UTF1")

data_selected_CTA_all <- df_z_scores_82[rownames(df_z_scores_82) %in%
    l_CTA_all, -c(1, 2)]
data_selected_CTA_all <- data_selected_CTA_all[!rownames(data_selected_CTA_all) %in%
    l, ]

# Create the heatmap
heatmap_cta_signif_conv_sorted_82 <- Heatmap(as.matrix(data_selected_CTA_all),
    cluster_rows = TRUE, cluster_columns = TRUE, cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean", clustering_method_columns = "complete",
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, show_row_names = TRUE, column_title = "Heatmap of selected CTA with signi
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
```

```
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif_conv_sorted_draw_82 <- draw(heatmap_cta_signif_conv_sorted_82)
```

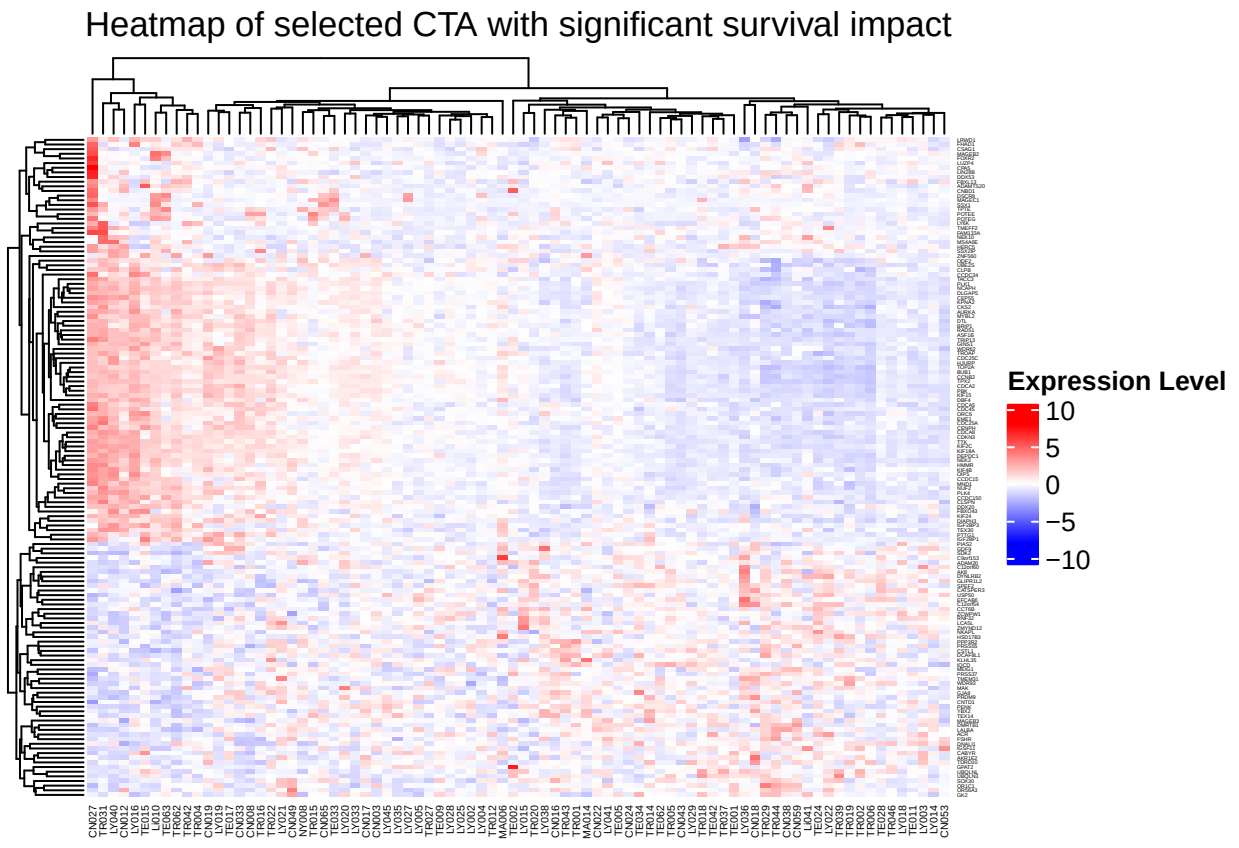


Figure 4: Heatmap of selected CTAs that impact survival (n = 82)

```r
# Store clusters Take col indexes
indiv_clust <- column_order(heatmap_cta_signif_conv_sorted_draw_82)

# Create table with indiv names
df_indiv_clusters_hm <- data.frame(Cluster = c(rep(1, length(indiv_clust[1:40])),
    rep(2, length(indiv_clust[41:82]))), Patient = c(colnames(data)[indiv_clust[1:40]],
    colnames(data)[indiv_clust[41:82]]))

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

b- Relative expression of non selected CTA

```r
data_non_selected_CTA_82 <- df_z_scores_82[rownames(df_z_scores_82) %in%
    l_CTA_all, -c(1, 2)]
data_non_selected_CTA_82 <- data_non_selected_CTA_82[rownames(data_non_selected_CTA_82) %in%
    l, ]

# Create the heatmap
heatmap_cta_signif_conv_non_selected_82 <- Heatmap(as.matrix(data_non_selected_CTA_82),
    cluster_rows = TRUE, cluster_columns = TRUE, cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean", clustering_method_columns = "complete",
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, show_row_names = TRUE, column_title = "Heatmap of other selected CTA with
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 4),
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif_conv_non_selected_82 <- draw(heatmap_cta_signif_conv_non_selected_82)
```

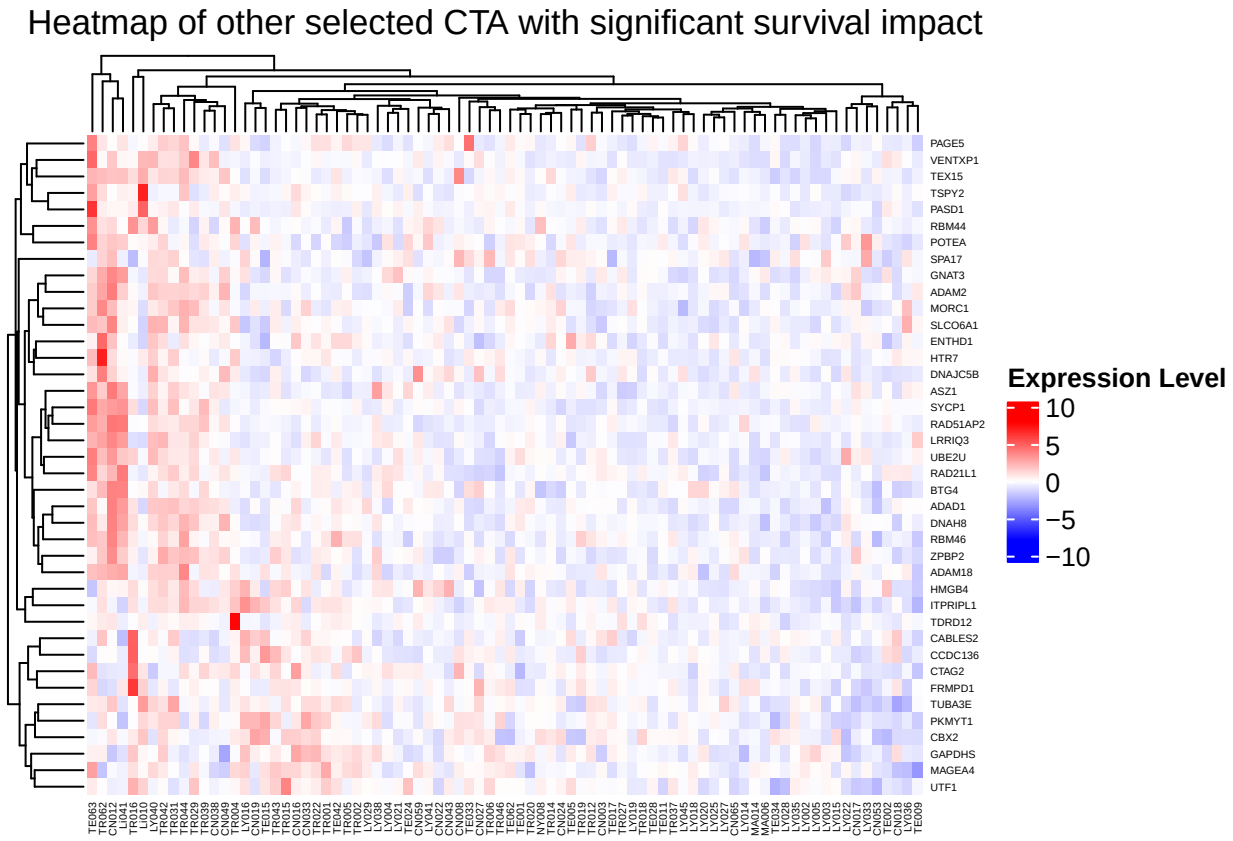


Figure 5: Heatmap of non selected CTAs that impact survival (n = 82)

```r
# Store clusters Take col indexes
indiv_clust <- column_order(heatmap_cta_signif_conv_non_selected_82)

# Create table with indiv names
df_indiv_clusters_hm <- data.frame(Cluster = c(rep(1, length(indiv_clust[1:15])),
    rep(2, length(indiv_clust[16:36])), rep(3, length(indiv_clust[37:73])),
    rep(4, length(indiv_clust[74:82]))), Patient = c(colnames(data_non_selected_CTA_82)[indiv_clust[1:15
    colnames(data_non_selected_CTA_82)[indiv_clust[16:36]], colnames(data_non_selected_CTA_82)[indiv_clu
    colnames(data_non_selected_CTA_82)[indiv_clust[74:82]]))

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

4) Relative expression of CTA associated with survival probability in conventional chondrosarcoma patients (with survival metadata)

In the literature, we can observe that the dedifferentiated end benign groups are an other category and can be very different than conventional chondrosarcoma (grade 1 to 3), so I select only the conventional patients. We also see later that some dedifferentiated patients are very infiltrated by immune cells so this can distort the results.

```r
# 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_cta_63 <- df_z_scores_63[rownames(df_z_scores_63) %in% l_CTA_conv,
    -c(1, 2)]

# Create the heatmap
heatmap_cta_signif_conv_63 <- Heatmap(as.matrix(data_cta_63),
    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 with significant survival impact",
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif_conv_63 <- draw(heatmap_cta_signif_conv_63)
```

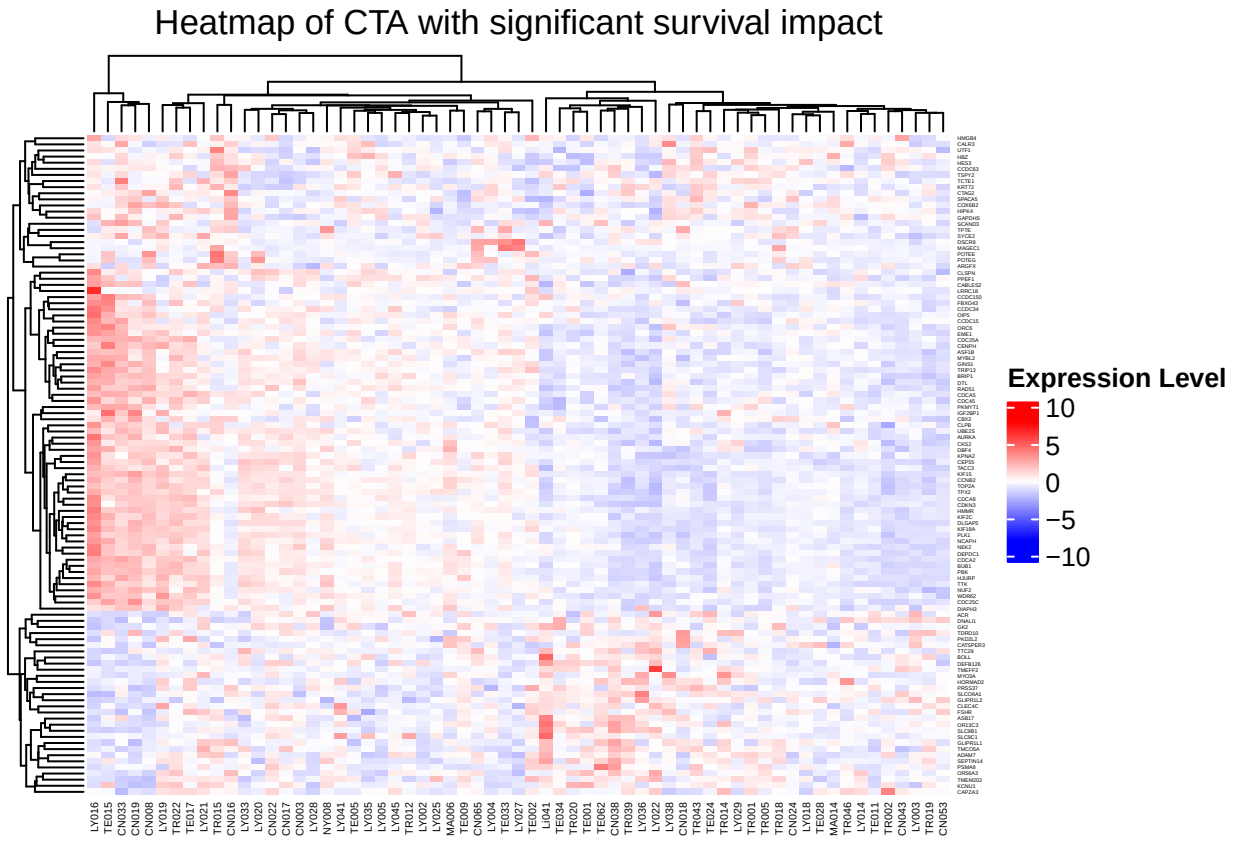


Figure 6: Heatmap of CTAs that impact survival (n = 63)

Similar to the previous heatmap, we observe three clusters of patients, which are then used for a survival analysis (see script 6).

```
# Store clusters Take col indexes
indiv_clust <- column_order(heatmap_cta_signif_conv_63)


# Create table with indiv names
df_indiv_clusters_hm <- data.frame(Cluster = c(rep(1, length(indiv_clust[1:5])),
    rep(2, length(indiv_clust[6:33])), rep(3, length(indiv_clust[34:63]))),
    Patient = c(colnames(data_cta_63)[indiv_clust[1:5]], colnames(data_cta_63)[indiv_clust[6:33]],
        colnames(data_cta_63)[indiv_clust[34:63]]))
# Save write.table(df_indiv_clusters_hm, file =
# '../results/clusters_indiv/clusters_conv_indiv_mhc_cta_signif_coxph.tsv',
# sep = '\t', quote = FALSE, row.names = FALSE)

# Store cluster to analyze jitter plot in script 6 Take row
# indexes
cta_clust <- row_order(heatmap_cta_signif_conv_63)

# Table with cta clusters
df_cta_clusters_hm <- data.frame(Cluster = c(rep(1, length(cta_clust[1:22])),
    rep(2, length(cta_clust[23:78])), rep(3, length(cta_clust[79:108]))),
    SYMBOL = c(rownames(data_cta_63)[cta_clust[1:22]], rownames(data_cta_63)[cta_clust[23:78]],
        rownames(data_cta_63)[cta_clust[79:108]]))

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

a- Relative expression of selected CTA

```
l <- c("HMGB4", "CALR3", "UTF1", "HBZ", "HES3", "CCDC63", "TSPY2",
    "TCTE1", "KRT72", "CTAG2", "SPACA5", "COX6B2", "HIPK4", "GAPDHS",
    "SCAND3", "TPTE", "SYCE2", "DSCR8", "MAGEC1", "POTEE", "POTEG",
    "ARGFX")

data_selected_CTA_conv_63 <- df_z_scores_63[rownames(df_z_scores_63) %in%
    l_CTA_conv, -c(1, 2)]
data_selected_CTA_conv_63 <- data_selected_CTA_conv_63[!rownames(data_selected_CTA_conv_63) %in%
    l, ]

# Create the heatmap
heatmap_cta_signif_conv_sorted_63 <- Heatmap(as.matrix(data_selected_CTA_conv_63),
    cluster_rows = TRUE, cluster_columns = TRUE, cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean", clustering_method_columns = "complete",
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, show_row_names = TRUE, column_title = "Heatmap of selected CTA with signi
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif_conv_sorted_draw_63 <- draw(heatmap_cta_signif_conv_sorted_63)
```

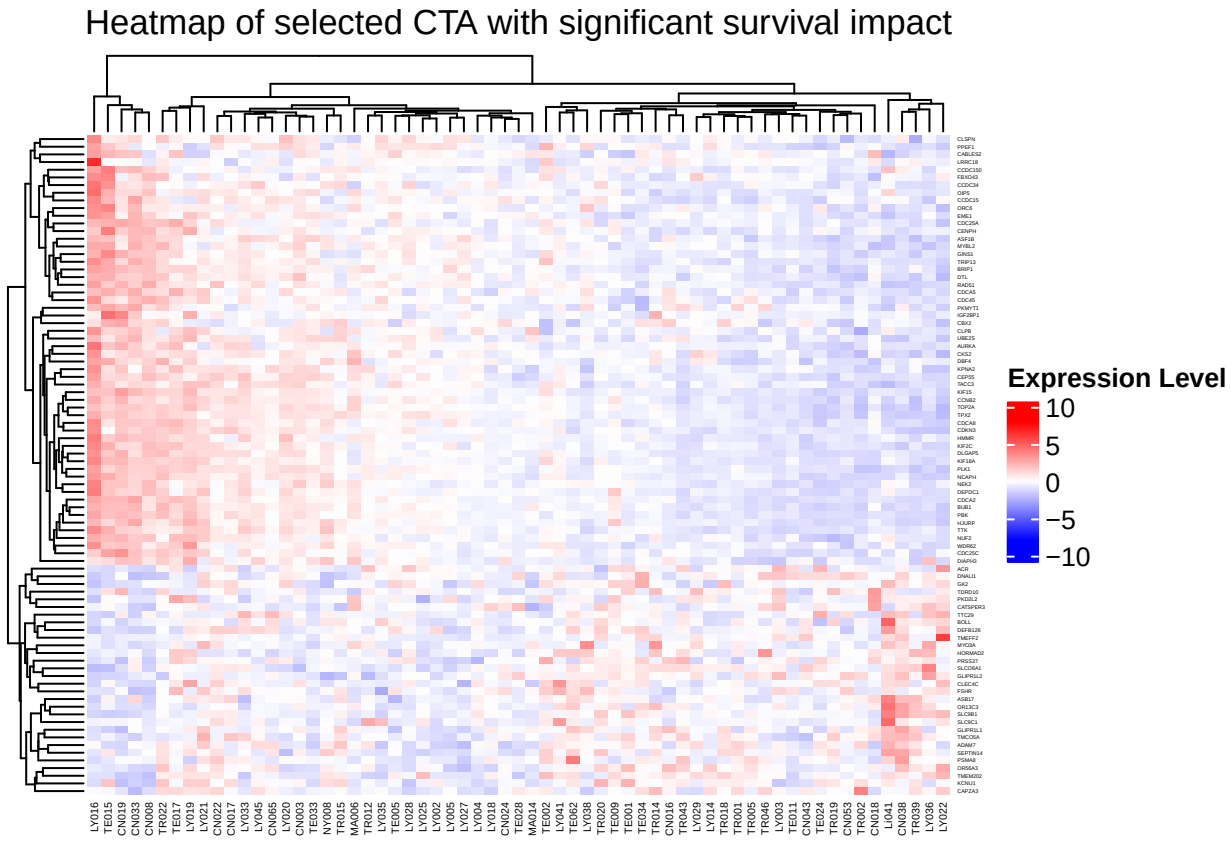


Figure 7: Heatmap of selected CTAs that impact survival (n = 63)

```r
# Store clusters Take col indexes
indiv_clust <- column_order(heatmap_cta_signif_conv_sorted_draw_63)

# Create table with indiv names
df_indiv_clusters_hm <- data.frame(Cluster = c(rep(1, length(indiv_clust[1:33])),
    rep(2, length(indiv_clust[34:63]))), Patient = c(colnames(data_selected_CTA_conv_63)[indiv_clust[1:3
    colnames(data_selected_CTA_conv_63)[indiv_clust[34:63]]))

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

b- Relative expression of non selected CTA

```r
data_non_selected_CTA_63 <- df_z_scores_63[rownames(df_z_scores_63) %in%
    l_CTA_conv, -c(1, 2)]
data_non_selected_CTA_63 <- data_non_selected_CTA_63[rownames(data_non_selected_CTA_63) %in%
    l, ]

# Create the heatmap
heatmap_cta_signif_conv_non_selected_63 <- Heatmap(as.matrix(data_non_selected_CTA_63),
    cluster_rows = TRUE, cluster_columns = TRUE, cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean", clustering_method_columns = "complete",
    col = colorRamp2(seq(-8, 8, length.out = 100), colors), border = NA,
    show_column_names = TRUE, show_row_names = TRUE, column_title = "Heatmap of other selected CTA with
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"))
heatmap_cta_signif_conv_non_selected_63 <- draw(heatmap_cta_signif_conv_non_selected_63)
```

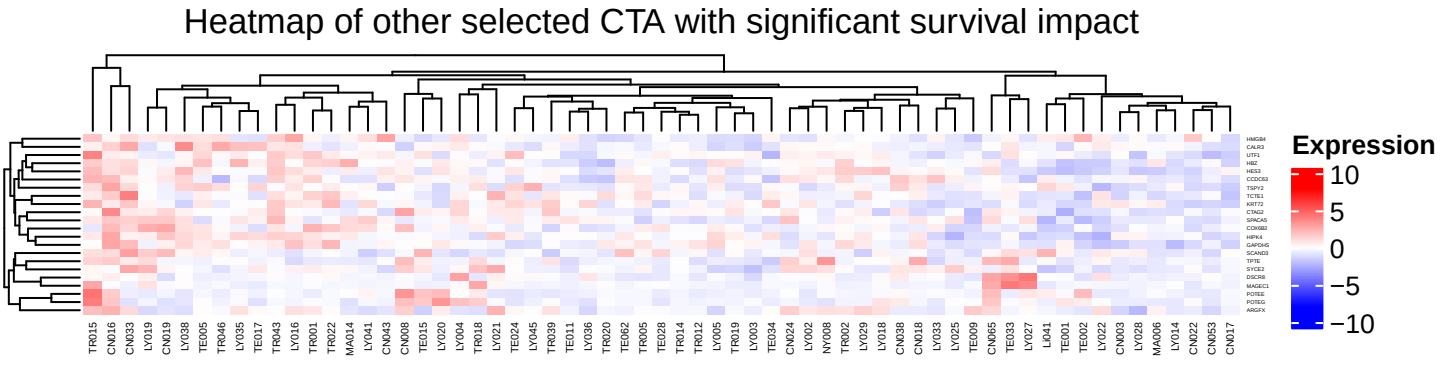


Figure 8: Heatmap of non selected CTAs that impact survival (n = 63)

```r
# Store clusters Take col indexes
indiv_clust <- column_order(heatmap_cta_signif_conv_non_selected_63)

# Create table with indiv names
df_indiv_clusters_hm <- data.frame(Cluster = c(rep(1, length(indiv_clust[1:17])),
    rep(2, length(indiv_clust[18:37])), rep(3, length(indiv_clust[38:49])),
    rep(4, length(indiv_clust[50:63]))), Patient = c(colnames(data_non_selected_CTA_63)[indiv_clust[1:17
    colnames(data_non_selected_CTA_63)[indiv_clust[18:37]], colnames(data_non_selected_CTA_63)[indiv_clu
    colnames(data_non_selected_CTA_63)[indiv_clust[50:63]]))

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

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_avg_immune_sign_z_scores_102)
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"))
```

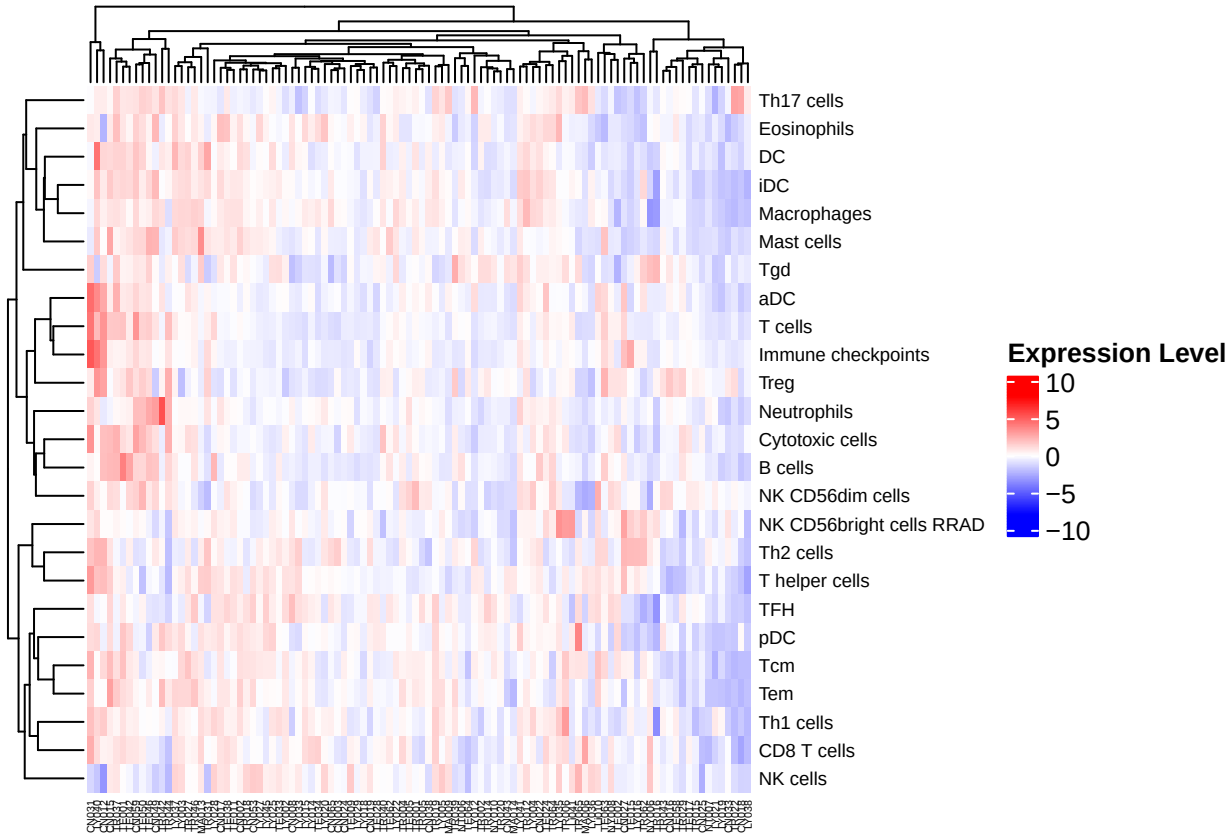


Figure 9: Heatmap and hierarchical clustering of relative immune cells expression (n = 102)

From the heatmap, we observe a clear 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")
```

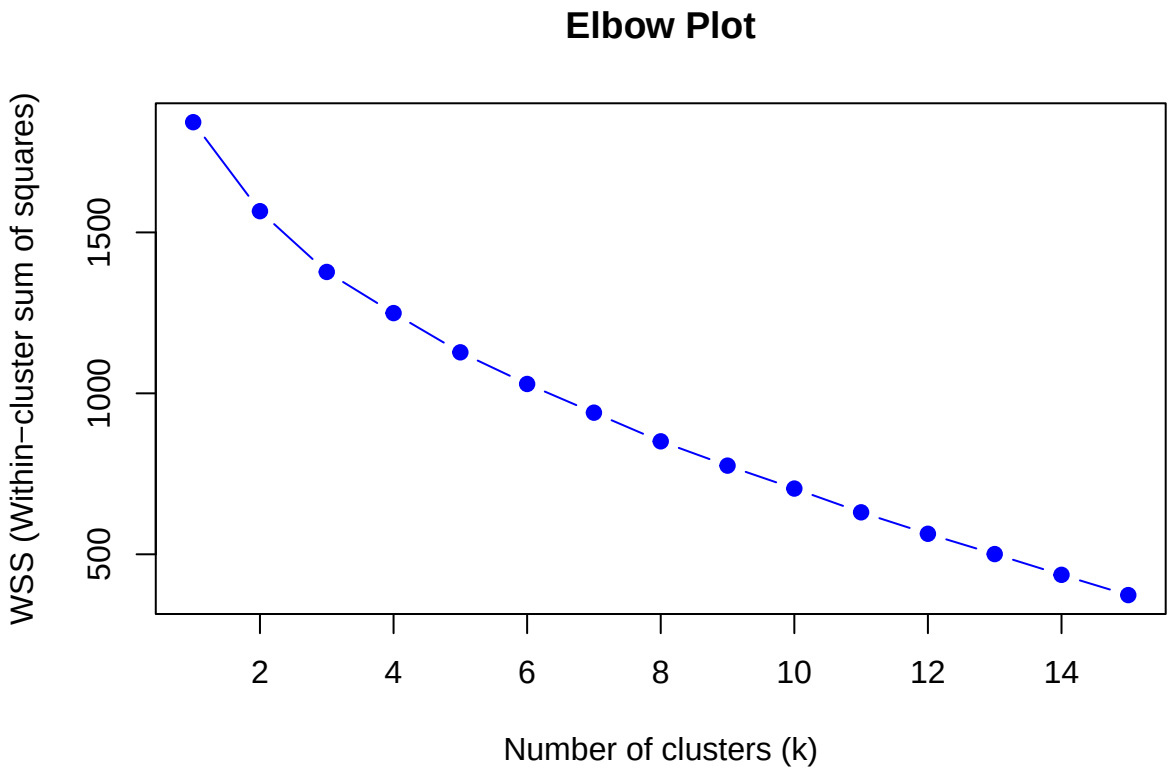


Figure 10: Elbow plot for k-means clustering

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

# Create heatmap
```

```r
heatmap <- Heatmap(
    as.matrix(heatmap_data),
    cluster_rows = FALSE,
    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)
```

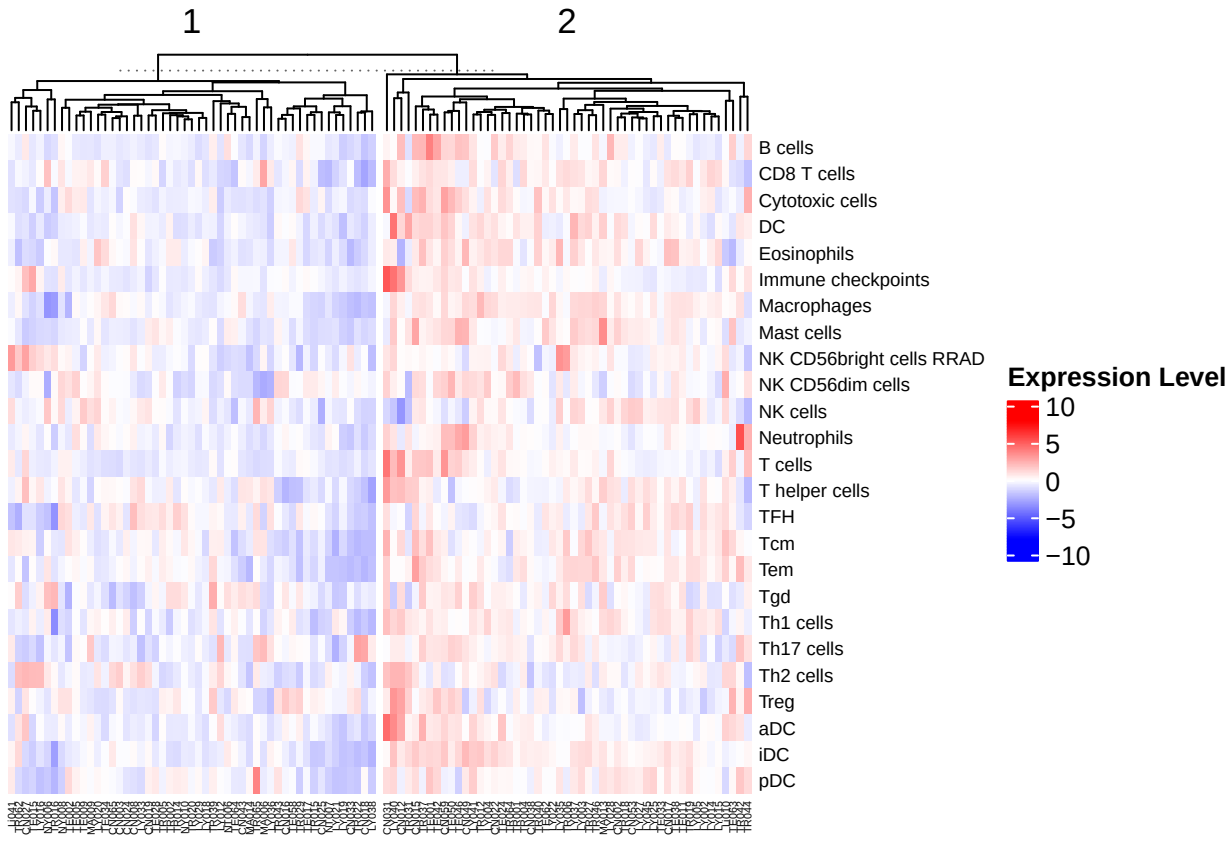


Figure 11: Heatmap with k-means clustering (k = 2) of immune cells expression (n = 102)

We observe that the two clusters correspond to cold and hot tumors, although the separation is not very clear, suggesting that there might be moderately infiltrated tumors. To further investigate, a Differentially Expressed Genes (DEG) analysis is also performed.

Differentilly expressed genes analysis

```
# Extract clusters from the previous heatmap
column_clusters <- column_order(heatmap)

# Loop through each cluster to store patient IDs
cluster_list <- list()
for (i in 1:length(column_clusters)) {
    cluster_data <- data.frame(PatientID = colnames(as.matrix(heatmap_data))[column_clusters[[i]]],
        Cluster = i)
    cluster_list[[i]] <- cluster_data
}

# Combine all clusters into a single data frame
patient_clusters <- do.call(rbind, cluster_list)


# Annotate the cluster with COLD and HOT
group_cluster <- patient_clusters

# Transform values
group_cluster$Cluster <- ifelse(group_cluster$Cluster == 1, "COLD",
    "HOT")

# New row
new_row <- setNames(rep(NA, ncol(df_CTA_immune_whole_clean_avg_102)),
    names(df_CTA_immune_whole_clean_avg_102))

# Assign COLD and HOT
for (i in seq_len(nrow(group_cluster))) {
    patient <- group_cluster$PatientID[i]
    cluster_value <- group_cluster$Cluster[i]
    matching_cols <- grep(patient, names(df_CTA_immune_whole_clean_avg_102),
        value = TRUE)
    new_row[matching_cols] <- cluster_value
}

# Add the new column
df_whole_cold_hot_km2 <- rbind(new_row, df_CTA_immune_whole_clean_avg_102)

# DEG analysis with limma
df <- t(df_whole_cold_hot_km2)
groups <- df[-c(1:3), 1]

# Create factors
f <- factor(groups, levels = c("COLD", "HOT"))
design <- model.matrix(~0 + f)  # 0 to compare all pairwises
colnames(design) <- c("COLD", "HOT")
```

```r
# Fit the linear model
data_fit <- lmFit(df_CTA_immune_whole_clean_avg_102[-c(1, 2,
    3)], design)

# Define contrasts (HOT vs. COLD)
contrast_matrix <- makeContrasts(HOT - COLD, levels = design)
data_fit_contrast <- contrasts.fit(data_fit, contrast_matrix)

# Calculate the empirical Bayes statistics
data_fit_eb <- eBayes(data_fit_contrast)

# Extract the top genes
res <- topTable(data_fit_eb, adjust = "BH", sort.by = "P", number = Inf)
# write.table(res, file =
# '../results/DEG_tables/deg_k2_all_patients.tsv', sep =
# '\t', row.names = FALSE, quote = FALSE)

# Volcano plot
EnhancedVolcano(res, lab = rownames(res), pCutoff = 0.01, FCcutoff = 0.8,
    x = "logFC", y = "adj.P.Val", pointSize = 1.5, legendLabSize = 10,
    labSize = 3, title = "Volcano plot with all genes", subtitle = "Cluster 1 vs cluster 2 from heatmap
```

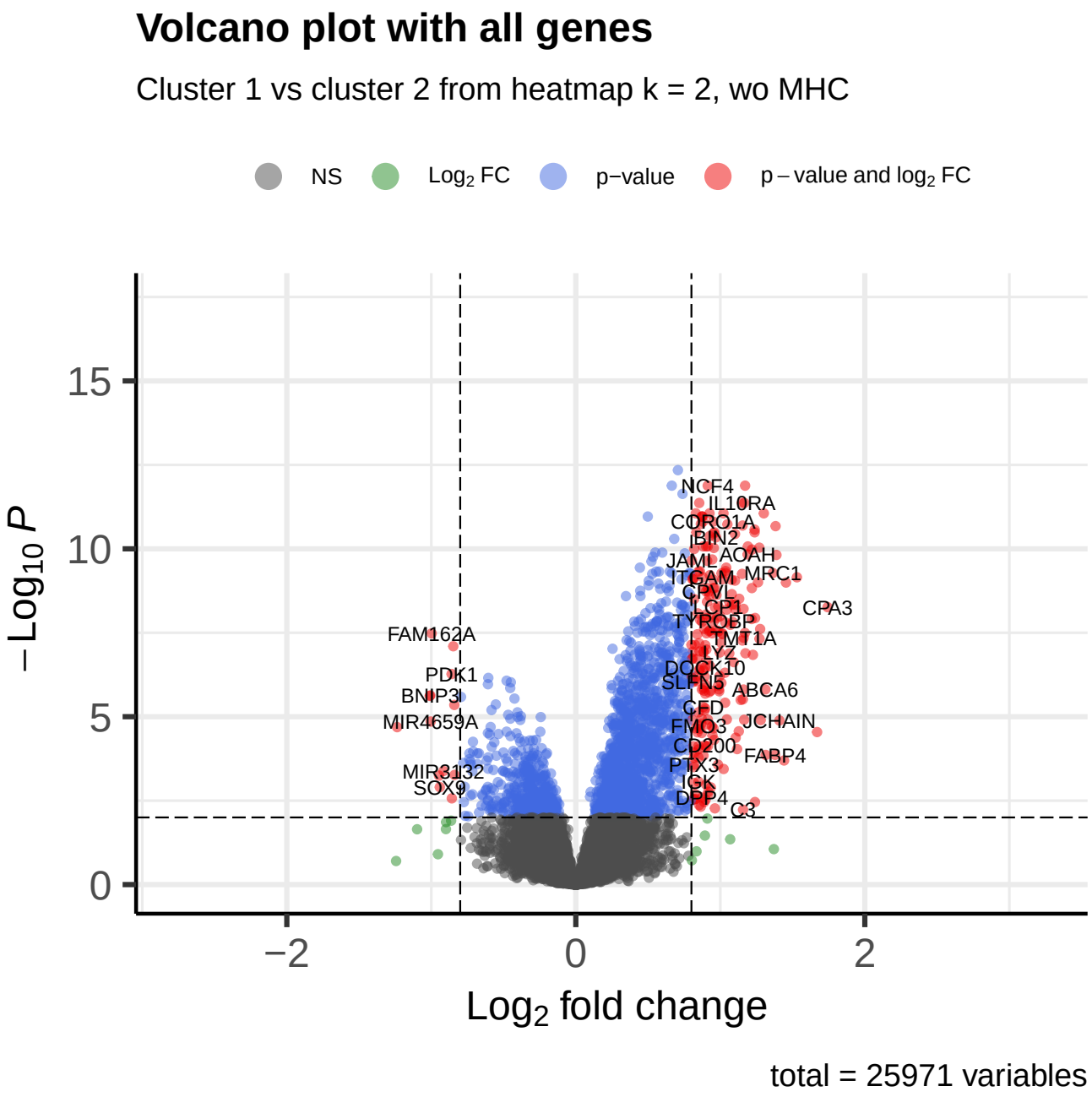


Figure 12: Volcano plot of DEG between C1 and C2 (from fig. 11)

The DEG analysis reveals that there are DEGs, but the log2 fold changes (log2FC) are small. To focus on more significant differences, we set a log2FC threshold of 0.8. There are 83 genes with a log2FC $>$ 1 or $<$ -1 and an adjusted p-value $<$ 0.01.

3) Clustering k-means with k = 4

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

```
set.seed(1)

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

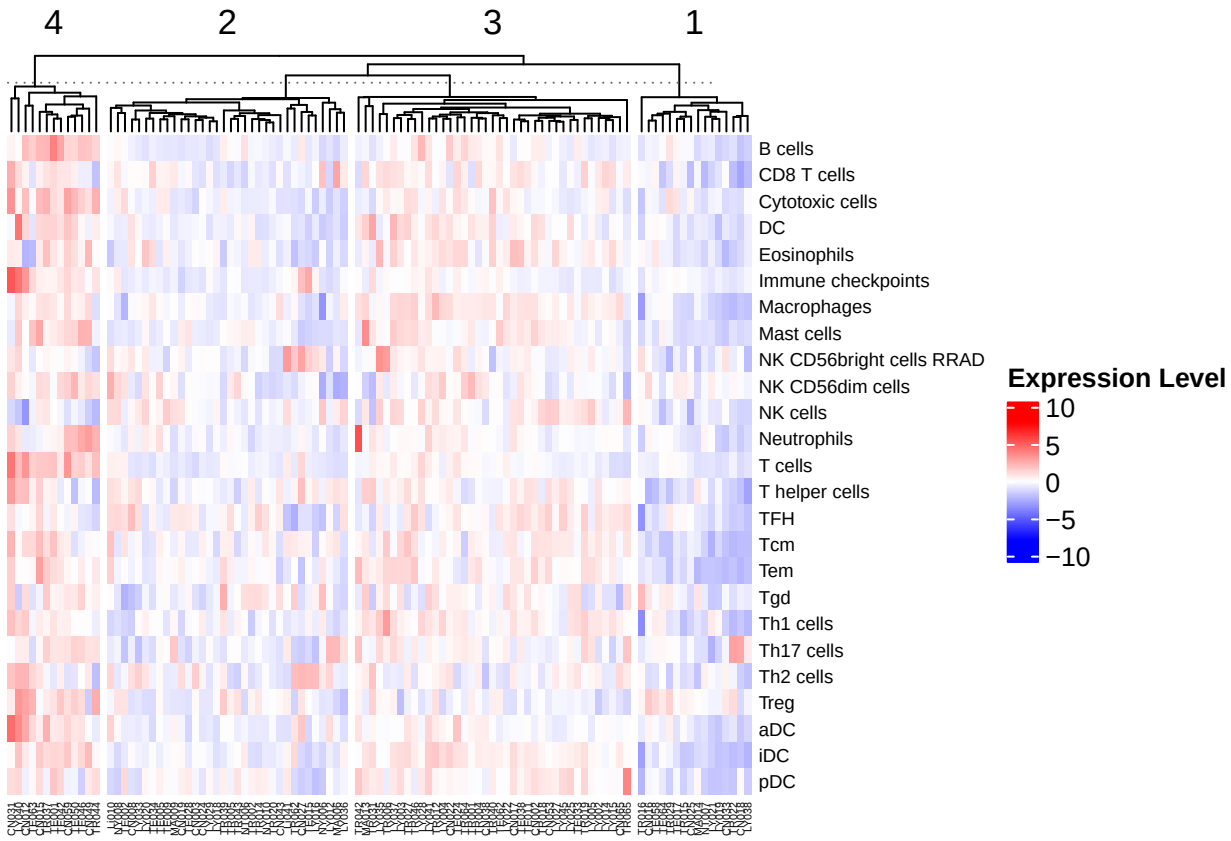


Figure 13: Heatmap with k-means clustering (k = 4) of immune cells expression (n = 102)

We observe that the two extreme clusters (COLD and HOT) are clearly separated, with a mix of intermediate clusters in between. To investigate further, we perform a DEG analysis on the extreme clusters.

a- Differentilly expressed genes analysis

```
# Select the clusters Extract clusters
column_clusters <- column_order(heatmap_km4)

# Loop through each cluster to store patient IDs
cluster_list <- list()
for (i in 1:length(column_clusters)) {
    cluster_data <- data.frame(PatientID = colnames(as.matrix(heatmap_data))[column_clusters[[i]]],
        Cluster = i)
    cluster_list[[i]] <- cluster_data
}

# Combine all clusters into a single data frame
patient_clusters <- do.call(rbind, cluster_list)
extreme_patient_clusters <- patient_clusters[patient_clusters$Cluster ==
    1 | patient_clusters$Cluster == 4, ]

# Annotate clusters
group_cluster <- extreme_patient_clusters
group_cluster$Cluster <- ifelse(group_cluster$Cluster == 1, "HOT",
    "COLD")
df_extr <- df_CTA_immune_whole_clean_avg_102[, group_cluster$PatientID]
rownames(df_extr) <- rownames(df_extr)

# Stock COLD and HOT
new_row <- setNames(rep(NA, ncol(df_extr)), names(df_extr))
for (i in seq_len(nrow(group_cluster))) {
    patient <- group_cluster$PatientID[i]
    cluster_value <- group_cluster$Cluster[i]

    matching_cols <- grep(patient, names(df_extr), value = TRUE)

    new_row[matching_cols] <- cluster_value
}
df_whole_cold_hot_km4_extr <- rbind(new_row, df_extr)

# DEG analysis
df <- t(df_whole_cold_hot_km4_extr)
groups <- df[, 1]

f <- factor(groups, levels = c("COLD", "HOT"))
design <- model.matrix(~0 + f)  # 0 to compare all pairwises
colnames(design) <- c("COLD", "HOT")

# Fit the linear model
data_fit <- lmFit(df_extr, design)

# Define contrasts (HOT vs. COLD)
```

```r
contrast_matrix <- makeContrasts(HOT - COLD, levels = design)
data_fit_contrast = contrasts.fit(data_fit, contrast_matrix)

# Calculate the empirical Bayes statistics
data_fit_eb <- eBayes(data_fit_contrast)

# Extract the top genes
res <- topTable(data_fit_eb, adjust = "BH", sort.by = "P", number = Inf)
res_sign <- topTable(data_fit_eb, adjust = "BH", sort.by = "P",
    number = Inf, p.value = 0.05, lfc = 1)

# Select CTA
deg <- rownames(res)
cta <- df_CTA_immune_whole_clean_avg_102 %>%
    filter(CTA != "NA") %>%
    filter(!grepl("^(NA,)*NA$", CTA))
cta <- rownames(cta)
deg_cta <- intersect(deg, cta)
res_cta <- res[deg_cta, ]
# write.table(res, file =
# '../results/DEG_tables/deg_k4_all_patients.tsv', sep =
# '\t', row.names = FALSE, quote = FALSE)

# Volcano plot with all the genes
EnhancedVolcano(res, lab = rownames(res), pCutoff = 0.01, FCcutoff = 0.8,
    x = "logFC", y = "adj.P.Val", pointSize = 1.5, legendLabSize = 10,
    labSize = 3, title = "Volcano plot with all genes", subtitle = "Cluster HOT vs cluster COLD from hea
```

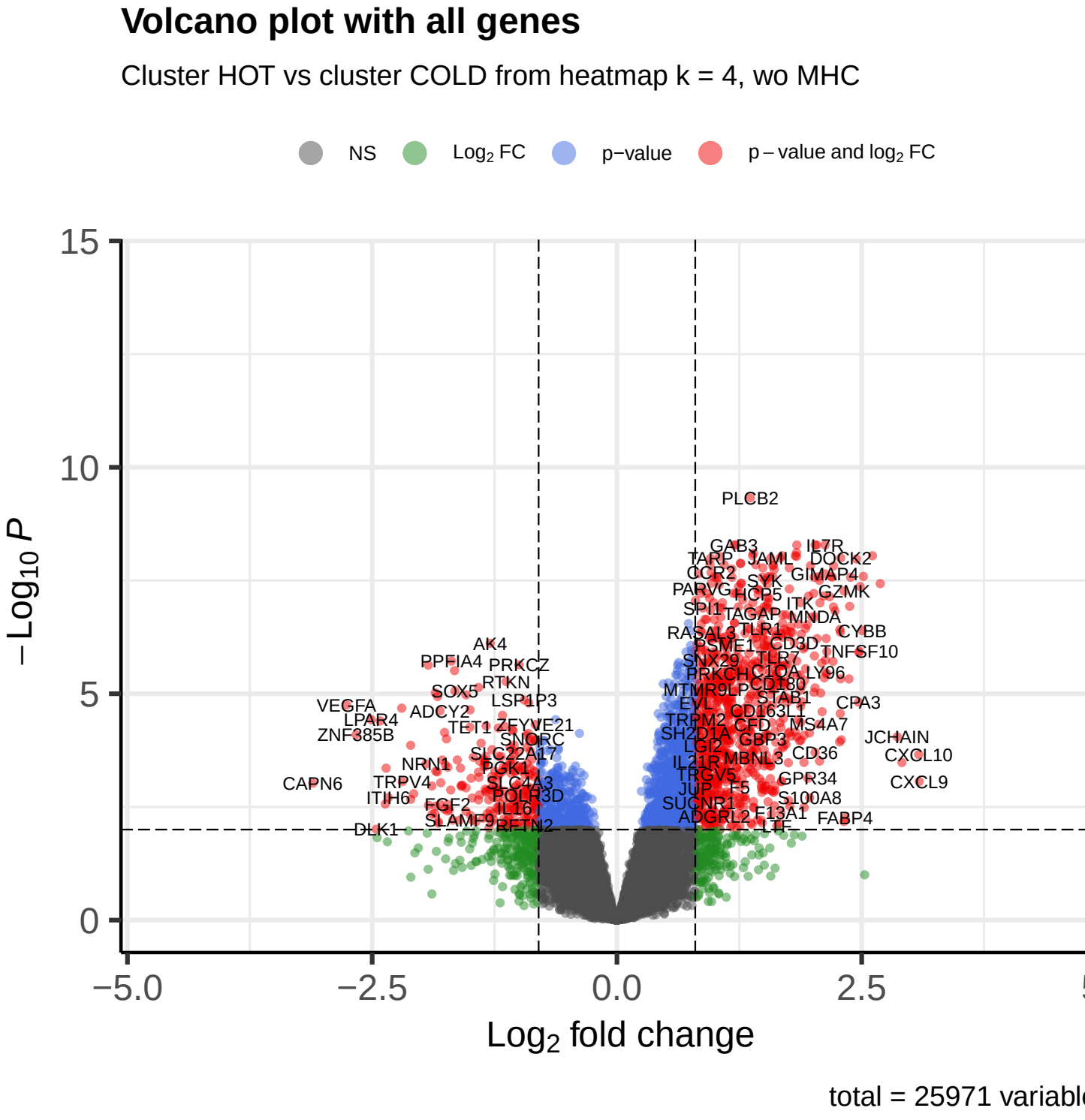


Figure 14: Volcano plot of DEG between C1 and C4 (from fig. 13)

```r
# Volcano plot for CTA genes
EnhancedVolcano(res_cta, lab = rownames(res_cta), pCutoff = 0.05,
    FCcutoff = 0.8, x = "logFC", y = "adj.P.Val", pointSize = 1.5,
    legendLabSize = 10, labSize = 3, title = "Volcano plot with CTA genes",
    subtitle = "Cluster HOT vs cluster COLD from heatmap k = 4, wo MHC")
```

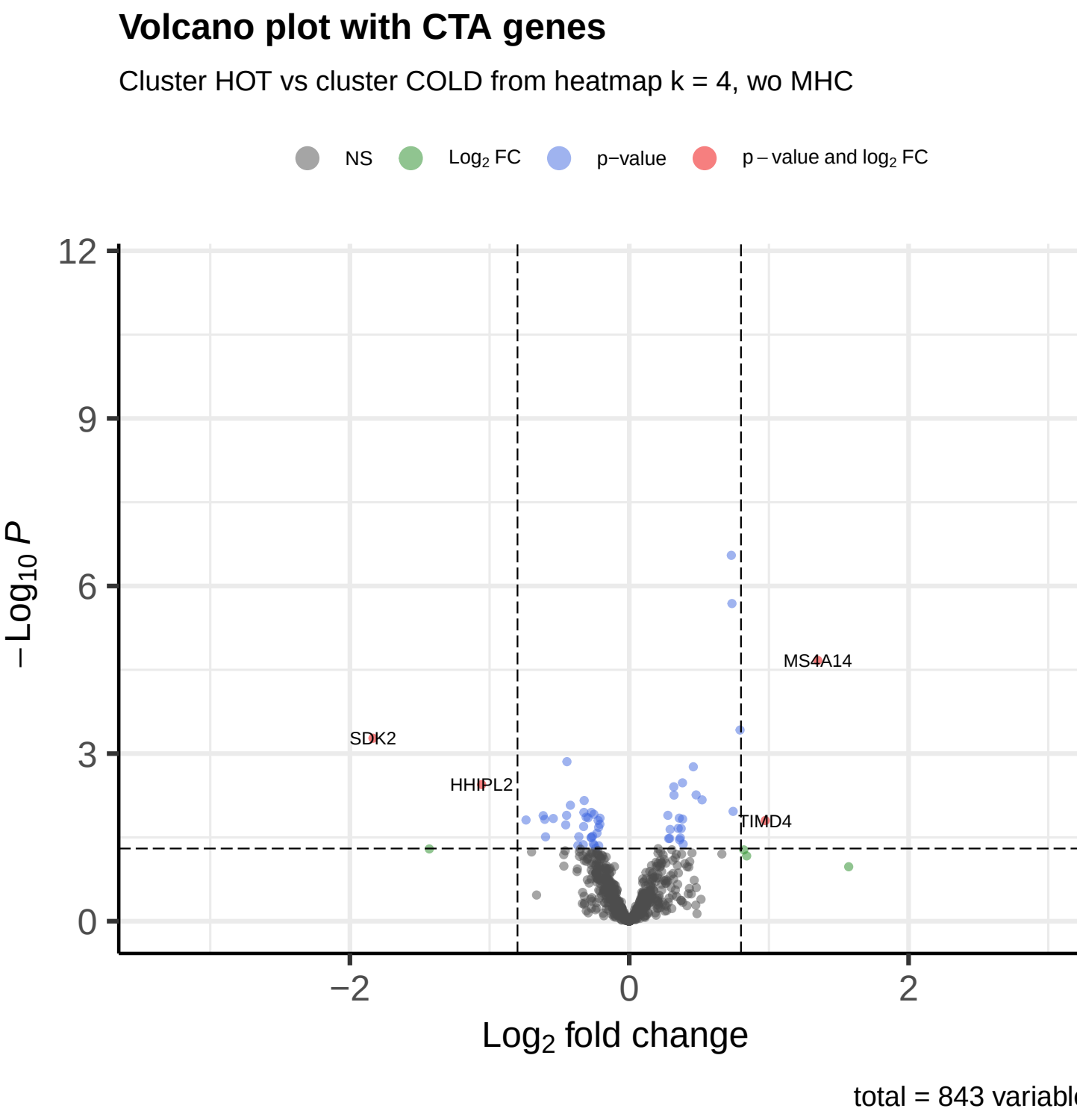


Figure 15: Volcano plot ofdifferentially expressed CTA between C1 and C4 (from fig. 13)

The first volcano plot considers all genes, with thresholds of 0.01 for adjusted p-value and 0.8 for log2FC. There are 74 DEGs with log2FC $>$ 2 or $<$ -2 and an adjusted p-value $<$ 0.01. The second plot show 4 differentially expressed CTA genes.

b- Verification of the intensities differences

To check whether the differences in expression are real, we visualize the intensity distributions for certain genes.

```
df_sdk2 <- t(df_whole_cold_hot_km4_extr[c("1", "SDK2"), ])

data_vector <- as.vector(as.numeric(df_sdk2[, 2]))

# Create boxplot
boxplot(data_vector ~ df_sdk2[, 1], main = "Boxplot : HOT vs COLD for SDK2",
    xlab = "Condition", ylab = "Values", col = c("blue", "red"),
    border = "black", names = c("COLD", "HOT"))
```

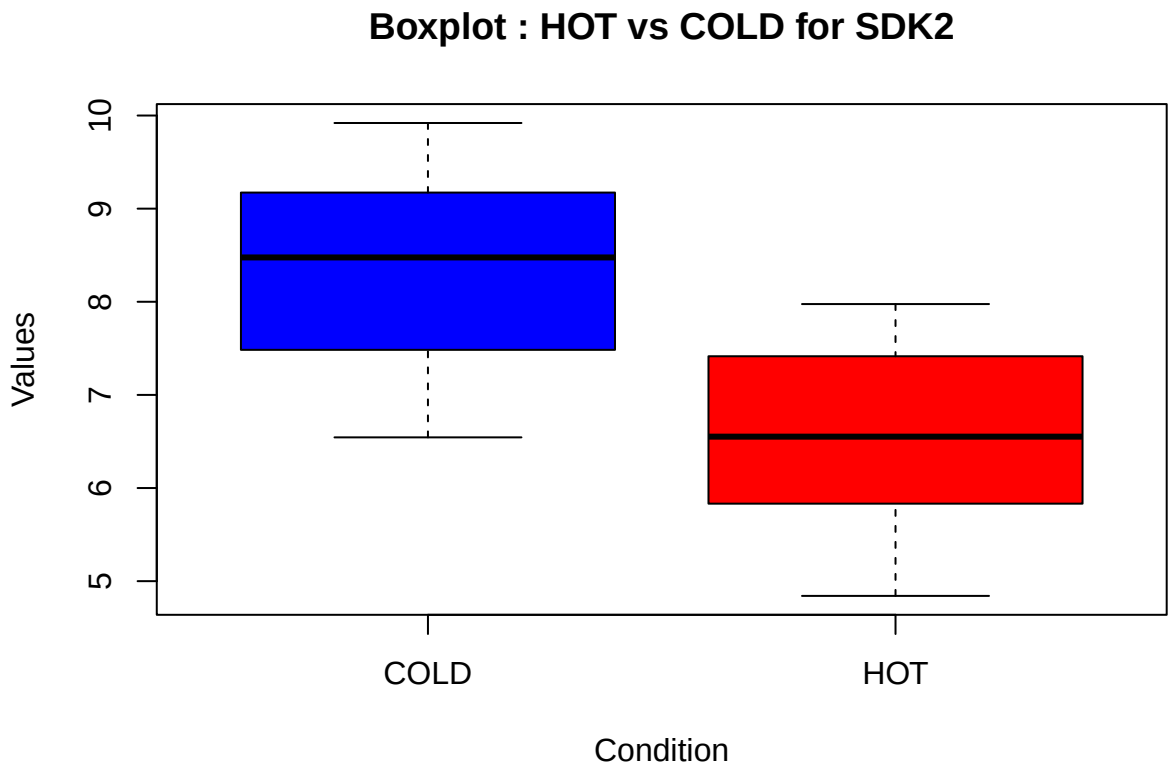


Figure 16: Intensities boxplot for SDK2 in C1 and C4 (from fig. 13)

i) SDK2

SDK2 appears to be more expressed in COLD tumors.

ii) Histogram of intensities for each genes To further visualize the intensity distribution across all genes:

```
hist(res$AveExpr, main = "Histogram of the average intensities for all genes",
    xlab = "Average expression")
```

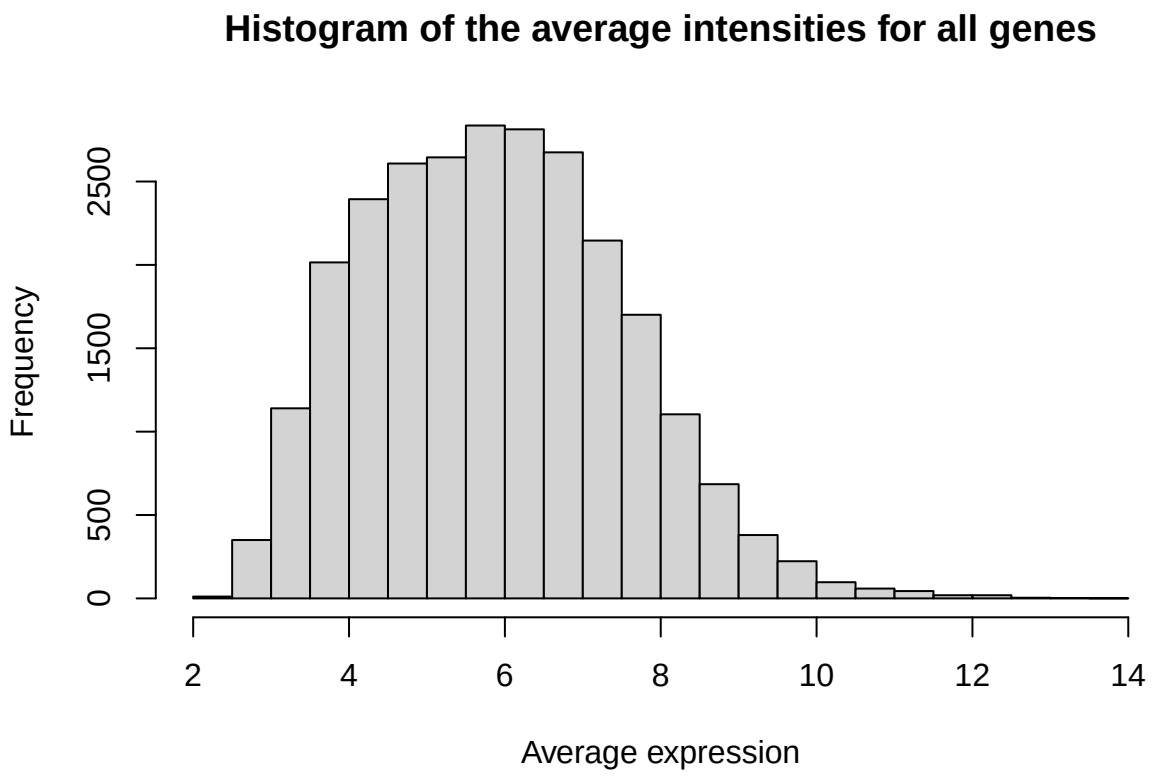


Figure 17: Histogram of the average intensities for all genes

For extreme patients, the average expression of SDK2 is 7.5. These value don't fall at the extremes of the histogram, suggesting that clustering and patient classification might be influenced by other factors, such as histology or the presence of immunosuppressive cells.

III. MHC genes integration

In this section, we integrate MHC (Major Histocompatibility Complex) genes due to their significant impact on immune-oncology within the tumor microenvironment. MHC genes play a crucial role in the immune response by presenting antigens to T-cells, thus influencing tumor immunogenicity.

```
# Read data
df_MHC <- read.table("../data/MHC_genes.txt", sep = "\t", header = TRUE,
    check.names = FALSE)

# Take MHC genes
df_expr_MHC <- merge(df_MHC, df_CTA_immune_whole_clean_z_scores_102,
    by.x = "SYMBOL")
```

1) MHC genes expression

First, we load and merge the MHC gene data with the existing dataset of immune cell signatures.

```
# Heatmap
heatmap_data_mhc <- as.matrix(df_expr_MHC[, -c(1, 2, 3, 4)])
rownames(heatmap_data_mhc) <- df_expr_MHC[, 1]
mhc_type <- df_expr_MHC$Type
row_name_colors <- ifelse(mhc_type == "MHC I", "green", "black")

# Heatmap
Heatmap(as.matrix(heatmap_data_mhc), 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, col = row_name_colors),
    heatmap_legend_param = list(title = "Expression Level"))
```

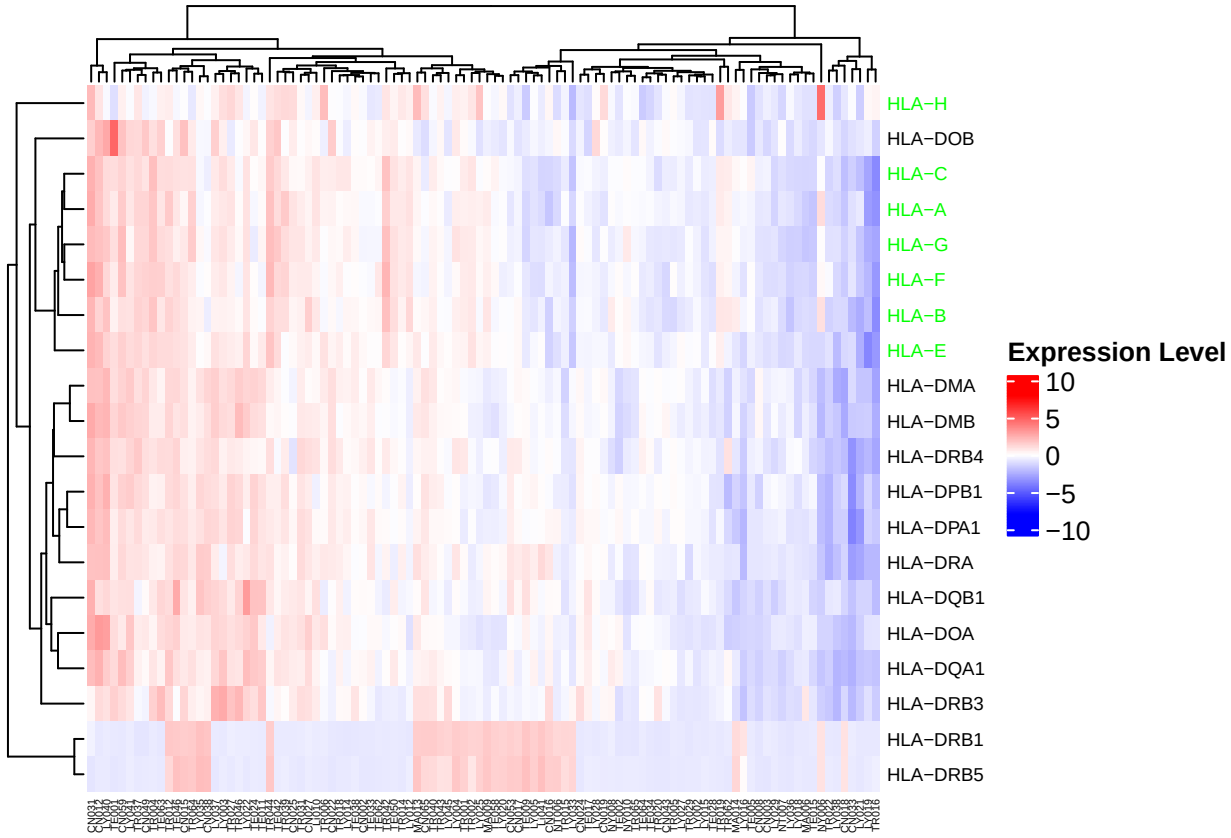


Figure 18: Heatmap of MHC genes expression (n = 102)

From the heatmap, we observe that HLA-DRB does not exhibit the same expression pattern as the other MHC genes.

2) MHC genes and immune cells signatures

Next, we integrate MHC genes with immune cell signatures to explore their role in the immune landscape of the tumor.

```
df_MHC_imm_sign <- read.table("../results/imm_sign_mhc_genes_avg_z_scores.tsv", sep = "\t", header = TRU

# Heatmap
heatmap_data_mhc_all <- as.data.frame(df_MHC_imm_sign)
rownames(heatmap_data_mhc_all) <- heatmap_data_mhc_all$Signature
heatmap_data_mhc_all <- heatmap_data_mhc_all[ , -1]  # Remove the Signature column

# Generates heatmap
set.seed(1)
heatmap_km4_mhc_all <- Heatmap(
    as.matrix(heatmap_data_mhc_all),
    cluster_rows = FALSE,
    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_mhc_all <- draw(heatmap_km4_mhc_all)
```

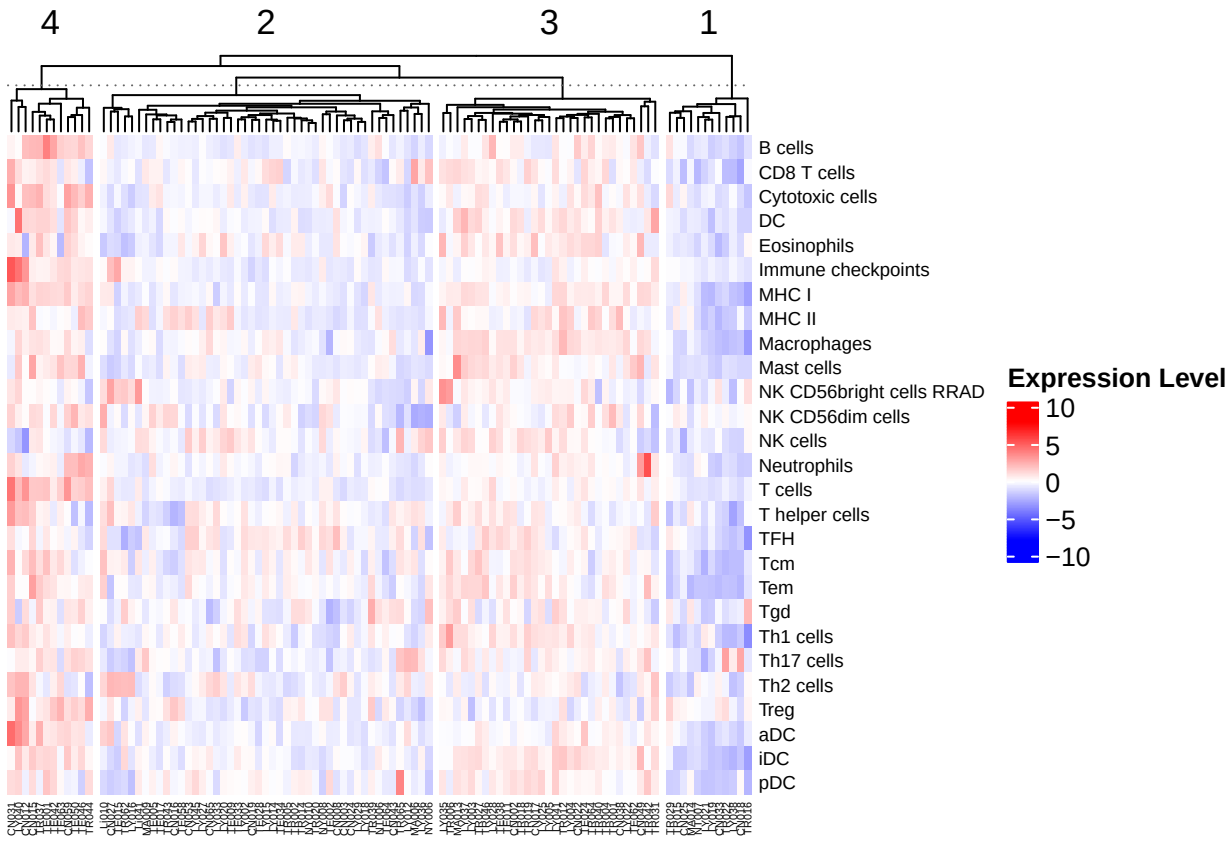


Figure 19: Heatmap with k-means clustering (k = 4) of immune cells expression (+ MHC) (n = 102)

The heatmap here indicates similar clustering patterns as before, which suggests that adding MHC genes does not drastically change the overall structure of the immune cell signatures. However, we observe distinct expression profiles for MHC types, particularly MHC I genes.

DEG of all patients with MHC clustering with clustering k-means (k = 4)

We now perform a differential expression analysis for the extreme clusters (HOT vs COLD) identified using the MHC clustering.

```
# Select and extract clusters
column_clusters <- column_order(heatmap_km4_mhc_all)

# Loop through each cluster to store patient IDs
cluster_list <- list()
for (i in 1:length(column_clusters)) {
    cluster_data <- data.frame(PatientID = colnames(as.matrix(heatmap_data_mhc_all))[column_clusters[[i]
        Cluster = i)
    cluster_list[[i]] <- cluster_data
}

# Combine all clusters into a single data frame
patient_clusters <- do.call(rbind, cluster_list)
extreme_patient_clusters_mhc <- patient_clusters[patient_clusters$Cluster ==
    1 | patient_clusters$Cluster == 4, ]

# Annotate clusters
group_cluster <- extreme_patient_clusters_mhc
group_cluster$Cluster <- ifelse(group_cluster$Cluster == 1, "HOT",
    "COLD")
df_extr <- df_CTA_immune_whole_clean_avg_102[, group_cluster$PatientID]

# Stock COLD and HOT
new_row <- setNames(rep(NA, ncol(df_extr)), names(df_extr))
for (i in seq_len(nrow(group_cluster))) {
    patient <- group_cluster$PatientID[i]
    cluster_value <- group_cluster$Cluster[i]
    matching_cols <- grep(patient, names(df_extr), value = TRUE)
    new_row[matching_cols] <- cluster_value
}
df_whole_cold_hot_km4_extr <- rbind(new_row, df_extr)

# DEG analysis
df <- t(df_whole_cold_hot_km4_extr)
groups <- df[, 1]

f <- factor(groups, levels = c("COLD", "HOT"))
design <- model.matrix(~0 + f)  # 0 to compare all pairwises
colnames(design) <- c("COLD", "HOT")

# Fit the linear model and take only columns with numeric
# values
data_fit <- lmFit(df_extr, design)
```

```r
# Define contrasts (HOT vs. COLD)
contrast_matrix <- makeContrasts(HOT - COLD, levels = design)
data_fit_contrast = contrasts.fit(data_fit, contrast_matrix)

# Calculate the empirical Bayes statistics
data_fit_eb <- eBayes(data_fit_contrast)

# Extract the top genes
res <- topTable(data_fit_eb, adjust = "BH", sort.by = "P", number = Inf)

# Select CTA
deg <- rownames(res)
cta <- df_CTA_immune_whole_clean_avg_102 %>%
    filter(CTA != "NA") %>%
    filter(!grepl("^(NA,)*NA$", CTA))
cta <- rownames(cta)
deg_cta <- intersect(deg, cta)
res_cta <- res[deg_cta, ]
# write.table(res, file =
# '../results/DEG_tables/deg_k4_all_patients.tsv', sep =
# '\t', row.names = FALSE, quote = FALSE)

# Volcano plots
EnhancedVolcano(res, lab = rownames(res), pCutoff = 0.01, FCcutoff = 0.8,
    x = "logFC", y = "adj.P.Val", pointSize = 1.5, legendLabSize = 10,
    labSize = 3, title = "Volcano plot with all genes", subtitle = "Cluster HOT vs cluster COLD from hea
```

```r
# Save DEG
res_whole <- res[res$logFC > 2 | res$logFC < -2, ]
res_whole <- res_whole[res_whole$P.Value < 0.01, ]
# write.table(res_whole[order(res_whole$logFC),], file =
# '../results/DEG_tables/deg_k4_mhc_14_indiv.tsv', quote =
# FALSE, sep = '\t')
```

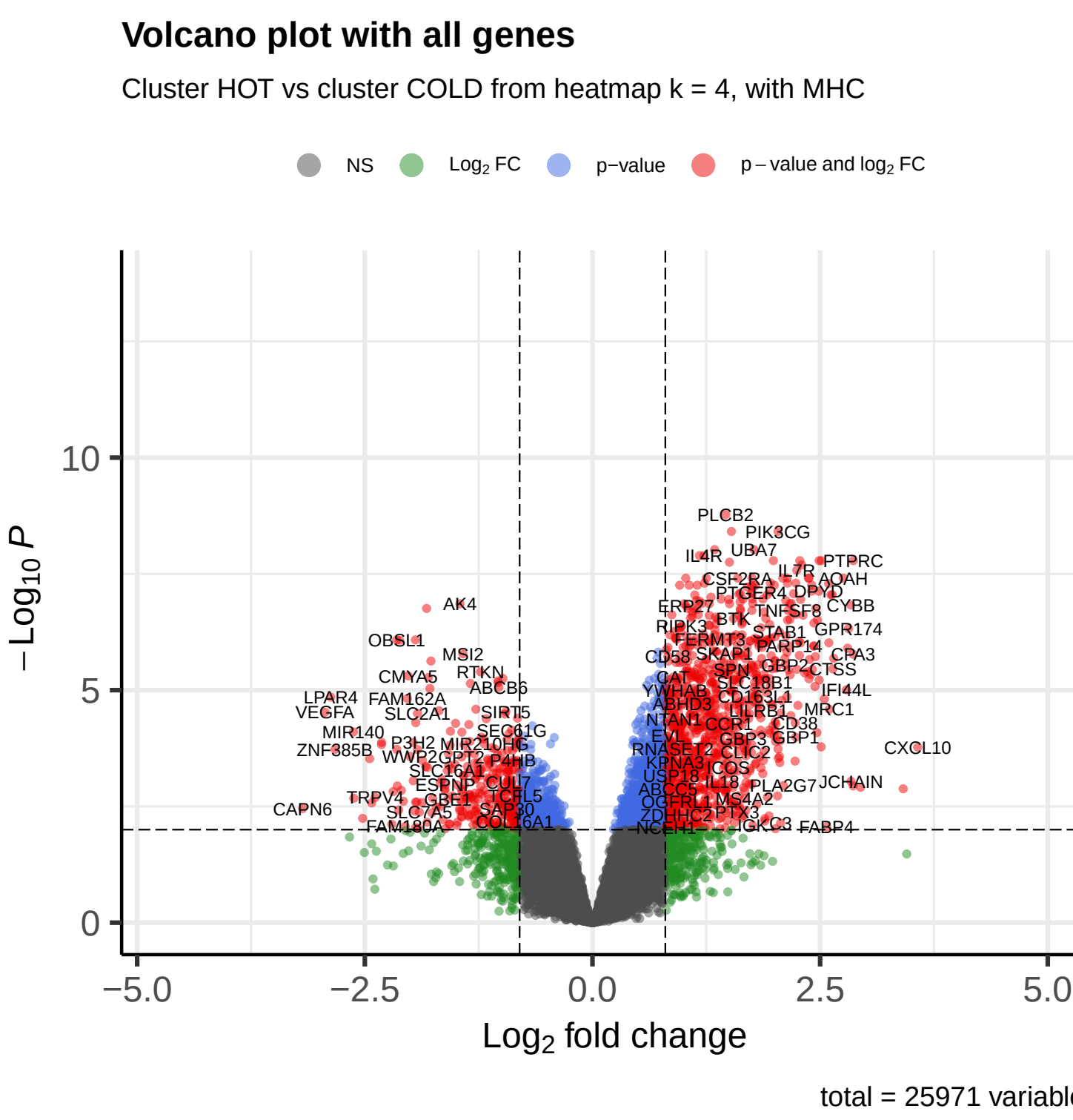


Figure 20: Volcano plot of DEG between C1 and C4 (from fig. 19)

```
EnhancedVolcano(res_cta, lab = rownames(res_cta), pCutoff = 0.05,
    FCcutoff = 0.8, x = "logFC", y = "adj.P.Val", pointSize = 1.5,
    legendLabSize = 10, labSize = 3, title = "Volcano plot with CTA genes",
    subtitle = "Cluster HOT vs cluster COLD from heatmap k = 4, with MHC")
```

```
res_cta[(res_cta$logFC > 1 | res_cta$logFC < -1) & res_cta$P.Value <
    0.05, ]
```

```
##            logFC  AveExpr         t      P.Value    adj.P.Val          B
## MS4A14  1.597826 5.704877  8.217498 1.399517e-08 1.873549e-06  9.8193642
## SDK2   -1.968038 7.504860 -4.980618 3.884112e-05 8.911156e-04  2.0298642
## HHIPL2 -1.139488 6.984319 -4.023488 4.640459e-04 6.040971e-03 -0.3865657
## DMP1    1.977792 5.360136  2.903641 7.586320e-03 4.804796e-02 -3.0505378
## PENK   -1.545385 7.071317 -2.769695 1.040544e-02 5.957170e-02 -3.3446145
```

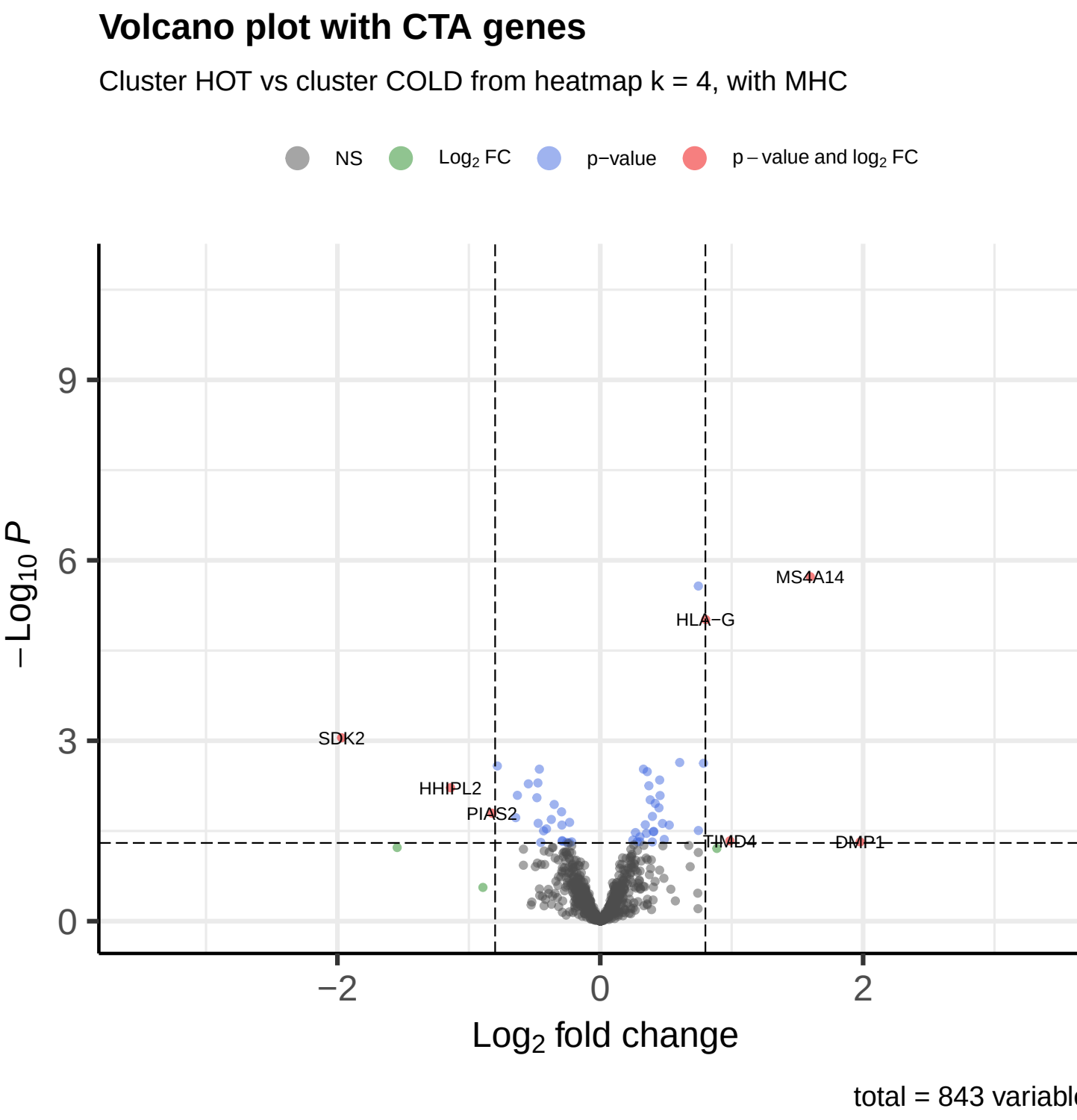


Figure 21: Volcano plot of differentially expressed CTA between C1 and C4 (from fig. 19)

The differential expression analysis shows a similar set of differentially expressed CTA genes as before, suggesting that the inclusion of MHC genes does not substantially alter the CTA gene expression profile between the HOT and COLD clusters. However, there are still notable differences in the immune landscape of the two groups.

IV. Relative immune cells expression without dedifferentiated and benign patients

In this section, we focus on the immune cells expression data, excluding dedifferentiated and benign patients, and perform clustering to investigate the infiltrated tumors or not.

```
# Read complete data with mhc annotation
df_complete <- read.table("../results/matrix_complete_intensities.tsv",
    sep = "\t", header = TRUE, check.names = FALSE)

df_complete_73 <- df_complete[, c("Signature", "CTA", colnames(df_complete)[colnames(df_complete) %in%
    patients_conv])]

# Average the expression between same immune cells types
# Take rows with immune cells signature from normalized
# data
df_avg_immune_sign_73 <- df_complete_73 %>%
    filter(Signature != "NA")

# Group by signature and calculate mean of expression
# values
df_avg_immune_sign_final_73 <- as.data.frame(df_avg_immune_sign_73 %>%
    select(-c(CTA)) %>%
    group_by(Signature) %>%
    summarise(across(where(is.numeric), \(x) mean(x,
        na.rm = TRUE))))

rownames(df_avg_immune_sign_final_73) <- df_avg_immune_sign_final_73$Signature
df_avg_immune_sign_final_73 <- df_avg_immune_sign_final_73[,
    -1]

df_imm_z_scores_73 <- t(scale(t(df_avg_immune_sign_final_73)))
```

1) Clustering with kmeans, k = 2 on columns

```
set.seed(1)
heatmap_km2_conv <- Heatmap(
    as.matrix(df_imm_z_scores_73),
    cluster_rows = FALSE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete",
    show_column_dend = TRUE,
    column_km = 2, # 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")
)
heatmap_km2_conv <- draw(heatmap_km2_conv)
```

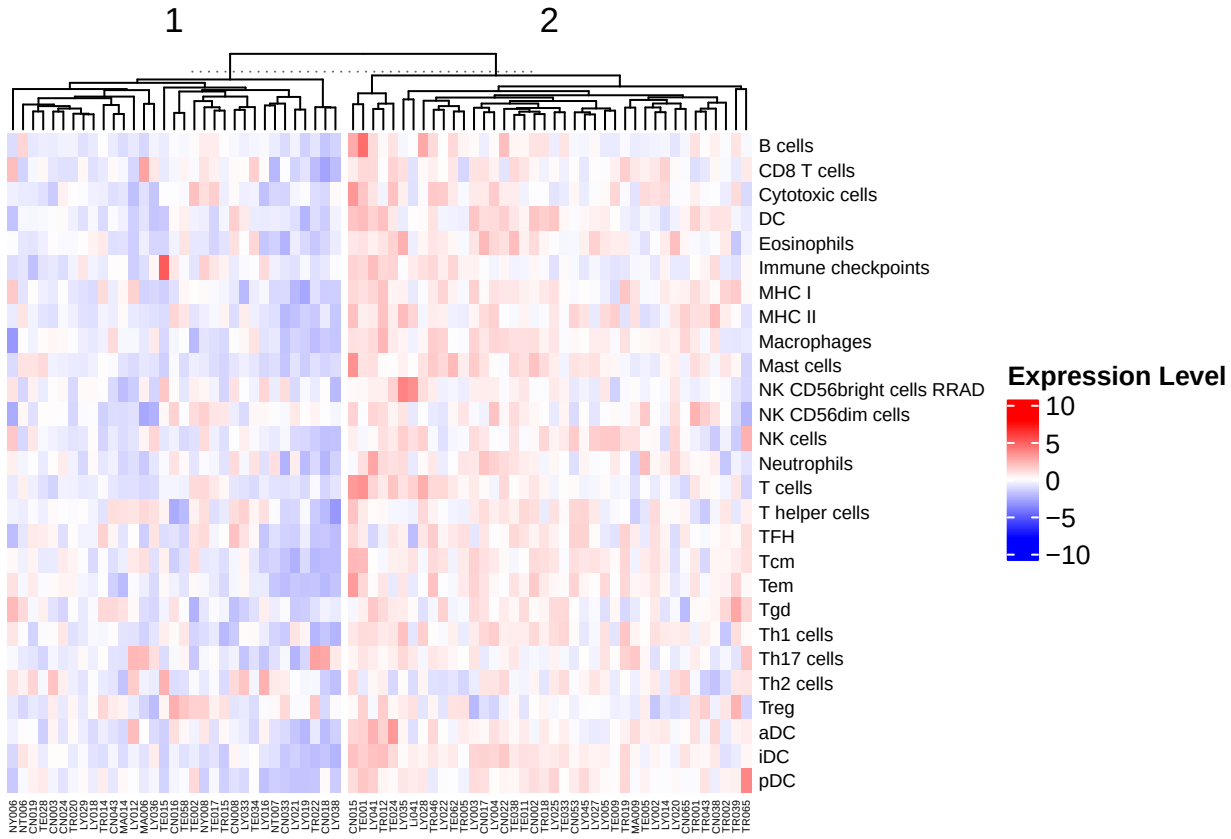


Figure 22: Heatmap with k-means clustering (k = 2) of immune cells expression (+ MHC) (n = 73)

The heatmap here visualizes immune cell expression for two clusters identified by K-means. We expect clearer clustering than in the previous all-patient dataset, which will allow us to perform more accurate differential expression analysis.

DEG analysis

```
# Extract clusters from the previous heatmap
column_clusters <- column_order(heatmap_km2_conv)

# Loop through each cluster to store patient IDs
cluster_list <- list()
for (i in 1:length(column_clusters)) {
    cluster_data <- data.frame(PatientID = colnames(as.matrix(df_imm_z_scores_73))[column_clusters[[i]]]
        Cluster = i)
    cluster_list[[i]] <- cluster_data
}

# Combine all clusters into a single data frame
patient_clusters <- do.call(rbind, cluster_list)

# Annotate the cluster with COLD and HOT
group_cluster <- patient_clusters

# Transform values
group_cluster$Cluster <- ifelse(group_cluster$Cluster == 1, "COLD",
    "HOT")

# New row
new_row <- setNames(rep(NA, ncol(df_CTA_immune_whole_clean_avg_73[,
    -c(1, 2)])), names(df_CTA_immune_whole_clean_avg_73[, -c(1,
    2)]))

# Assign COLD and HOT
for (i in seq_len(nrow(group_cluster))) {
    patient <- group_cluster$PatientID[i]
    cluster_value <- group_cluster$Cluster[i]
    matching_cols <- grep(patient, names(df_CTA_immune_whole_clean_avg_73[,
        -c(1, 2)]), value = TRUE)
    new_row[matching_cols] <- cluster_value
}

# Add the new column
df_whole_cold_hot_km2_conv <- rbind(new_row, df_CTA_immune_whole_clean_avg_73[,
    -c(1, 2)])

# DEG
groups <- df_whole_cold_hot_km2_conv[1, ]

# Create factors
f <- factor(groups, levels = c("COLD", "HOT"))
design <- model.matrix(~0 + f)  # 0 to compare all pairwises
colnames(design) <- c("COLD", "HOT")
```

```r
# Fit the linear model
data_fit <- lmFit(df_CTA_immune_whole_clean_avg_73[, -c(1, 2)],
    design)

# Define contrasts (HOT vs. COLD)
contrast_matrix <- makeContrasts(HOT - COLD, levels = design)
data_fit_contrast <- contrasts.fit(data_fit, contrast_matrix)

# Calculate the empirical Bayes statistics
data_fit_eb <- eBayes(data_fit_contrast)

# Extract the top genes
res <- topTable(data_fit_eb, adjust = "BH", sort.by = "P", number = Inf)
res_cta <- res[deg_cta, ]

# Volcano plot
EnhancedVolcano(res, lab = rownames(res), pCutoff = 0.01, FCcutoff = 0.8,
    x = "logFC", y = "adj.P.Val", pointSize = 1.5, legendLabSize = 10,
    labSize = 3, title = "Volcano plot with all genes", subtitle = "Cluster 1 vs cluster 2 from heatmap
```

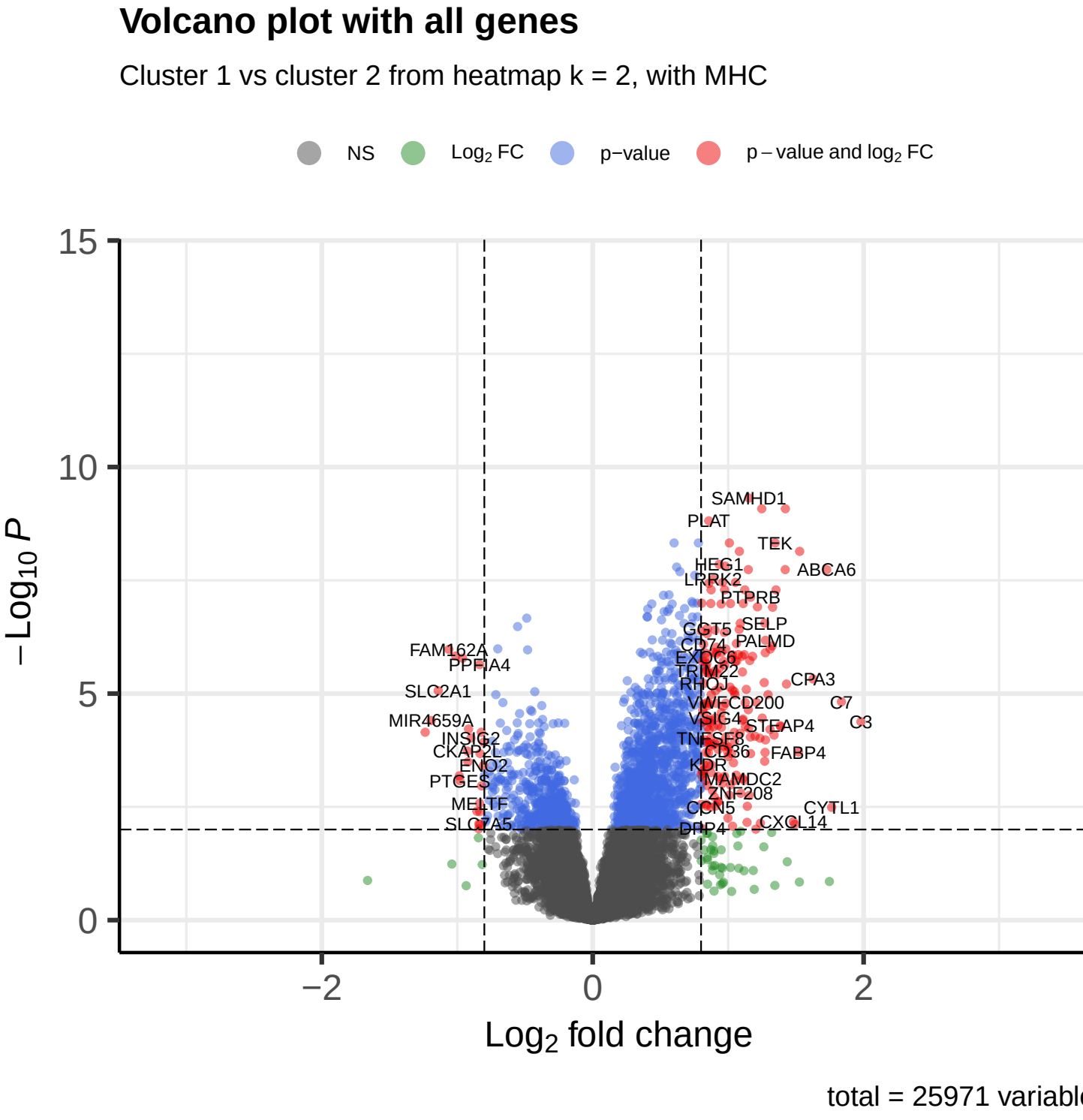


Figure 23: Volcano plot of DEG between C1 and C2 (from fig. 22)

```r
EnhancedVolcano(res_cta, lab = rownames(res_cta), pCutoff = 0.05,
    FCcutoff = 0.8, x = "logFC", y = "adj.P.Val", pointSize = 1.5,
    legendLabSize = 10, labSize = 3, title = "Volcano plot with CTA genes",
    subtitle = "Cluster 1 vs cluster 2 from heatmap k = 2, wth MHC")
```

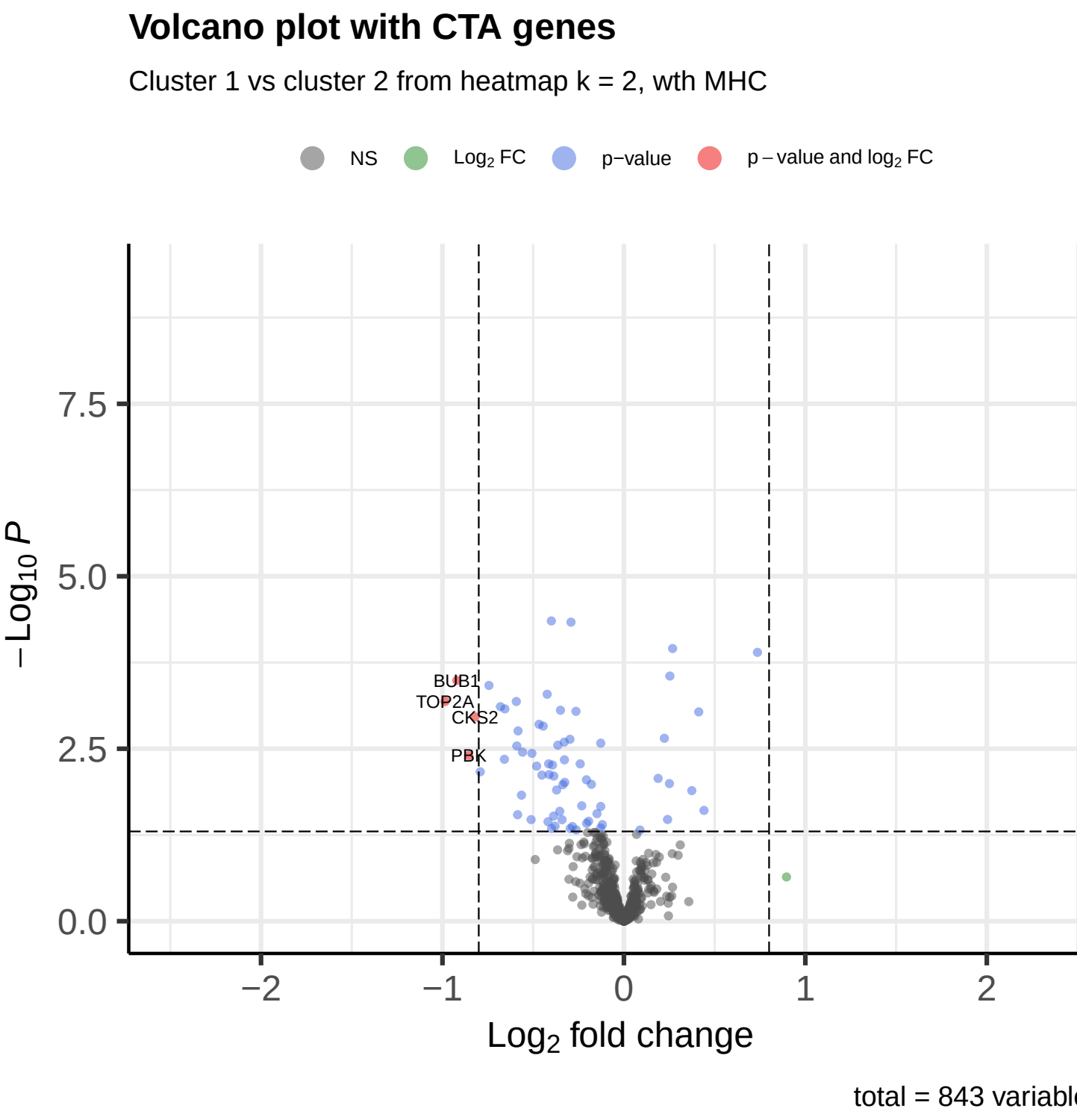


Figure 24: Volcano plot of differentially expressed CTA between C1 and C2 (from fig. 22)

Thanks to volcano plots, we see that there is DEG between the 2 groups. For the CTA, we retrieve BUB1 and PBK over expressed in cold tumors. This 2 CTA are interesting because in the literature, researchers have defined these like pan cancer targets.

2) Clustering with kmeans, k = 4

```
set.seed(1)
Heatmap(
    as.matrix(df_imm_z_scores_73),
    cluster_rows = FALSE,
    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")
)
```

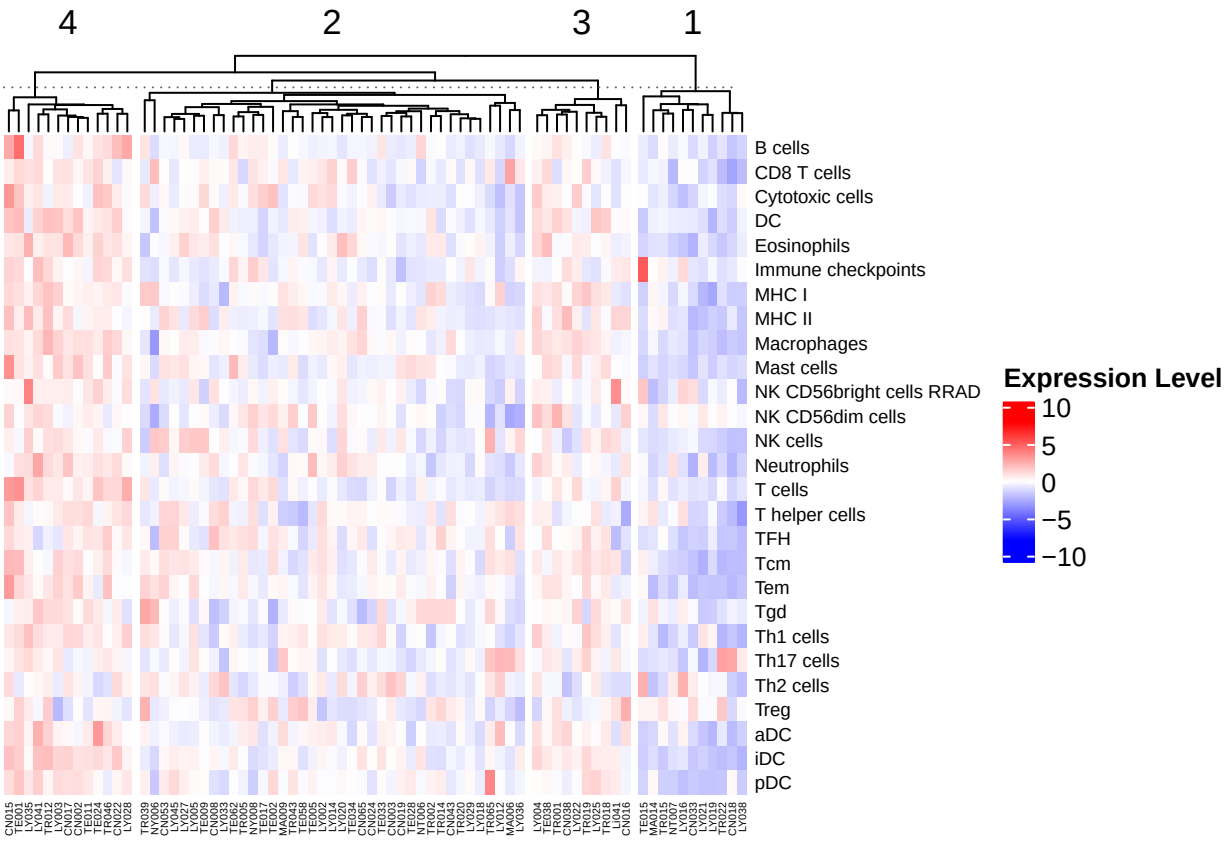


Figure 25: Heatmap with k-means clustering (k = 4) of immune cells expression (+ MHC) (n = 73)

Here, the clustering isn't better than with all patients so I didn't analyze the DEG.

3) Clustering with k = 3

```
set.seed(1)
heatmap_conv_k3 <- Heatmap(
    as.matrix(df_imm_z_scores_73),
    cluster_rows = FALSE,
    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),
    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_conv_k3 <- draw(heatmap_conv_k3)
```

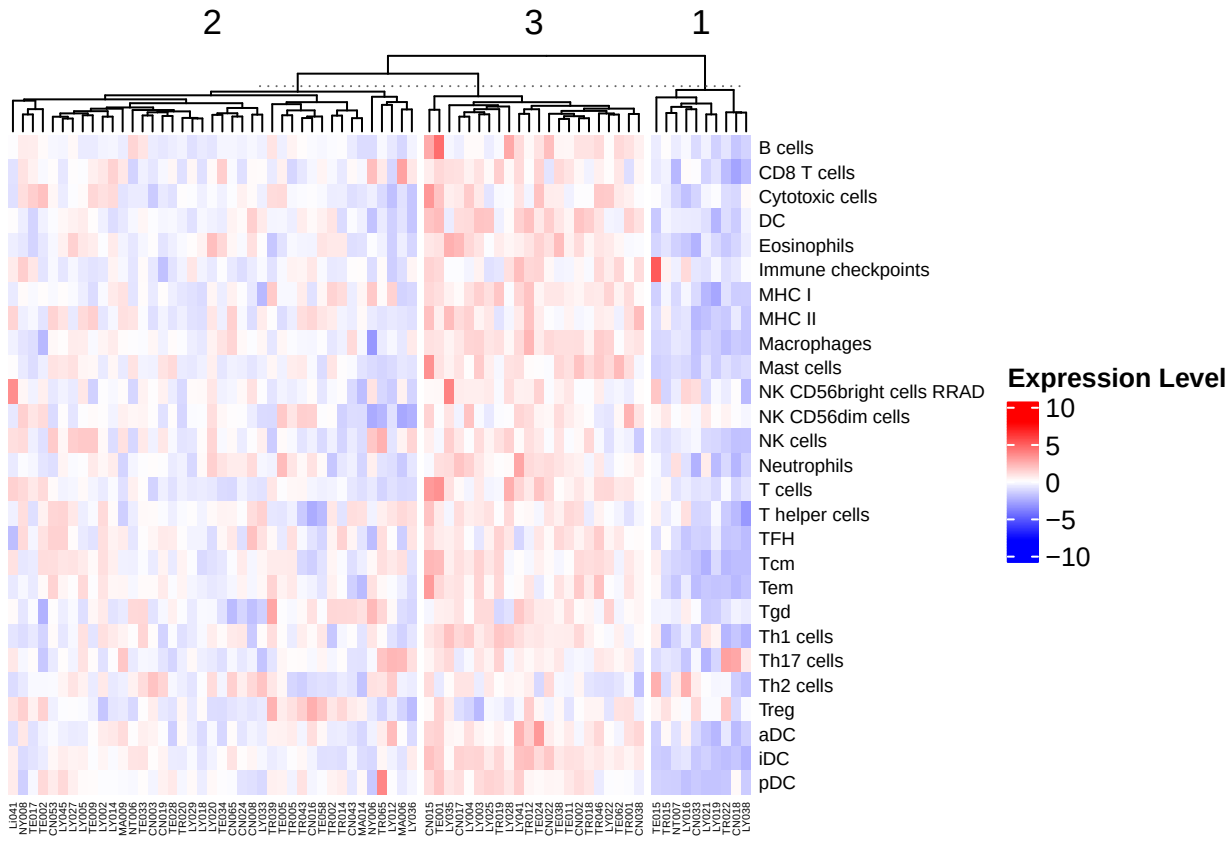


Figure 26: Heatmap with k-means clustering (k = 3) of immune cells expression (+ MHC) (n = 73)

We see a HOT cluster (C3), a COLD (C1) and an intermediate. We will investigate DEG to see the differences between each cluters

DEG analysis

```
# Store clusters Take col indexes
indiv_clust <- column_order(heatmap_conv_k3)

# Create table with indiv names
df_indiv_clusters_hm_anno <- data.frame(Cluster = c(rep(1, length(indiv_clust$`1`)),
    rep(2, length(indiv_clust$`2`)), rep(3, length(indiv_clust$`3`))),
    Patient = c(colnames(heatmap_data_mhc_all)[indiv_clust$`1`],
        colnames(heatmap_data_mhc_all)[indiv_clust$`2`], colnames(heatmap_data_mhc_all)[indiv_clust$`3`]

# DEG
f <- factor(df_indiv_clusters_hm_anno$Cluster)
design <- model.matrix(~0 + f)  # 0 to compare all pairwises
colnames(design) <- c("C1", "C2", "C3")
colnames(design) <- make.names(colnames(design))

# Fit the linear model
data_fit <- lmFit(df_CTA_immune_whole_clean_avg_73[, -c(1, 2)],
    design)

# Define contrasts (HOT vs. COLD)
contrast_matrix <- makeContrasts(C1_vs_C2 = C2 - C1, C2_vs_C3 = C3 -
    C2, C1_vs_C3 = C3 - C1, levels = design)

data_fit_contrast <- contrasts.fit(data_fit, contrast_matrix)

# Calculate the empirical Bayes statistics
data_fit_eb <- eBayes(data_fit_contrast)

# Extract genes
resultats <- list()
resultats$C1_vs_C2 <- topTable(data_fit_eb, coef = "C1_vs_C2",
    adjust = "BH", sort.by = "P", number = Inf)
resultats$C2_vs_C3 <- topTable(data_fit_eb, coef = "C2_vs_C3",
    adjust = "BH", sort.by = "P", number = Inf)
resultats$C1_vs_C3 <- topTable(data_fit_eb, coef = "C1_vs_C3",
    adjust = "BH", sort.by = "P", number = Inf)

# Volcano plot
EnhancedVolcano(resultats$C1_vs_C2, lab = rownames(resultats$C1_vs_C2),
    pCutoff = 0.05, FCcutoff = 0.8, x = "logFC", y = "adj.P.Val",
    pointSize = 1.5, legendLabSize = 10, labSize = 3, title = "Volcano plot with all genes",
    subtitle = "C1 vs C2")
```

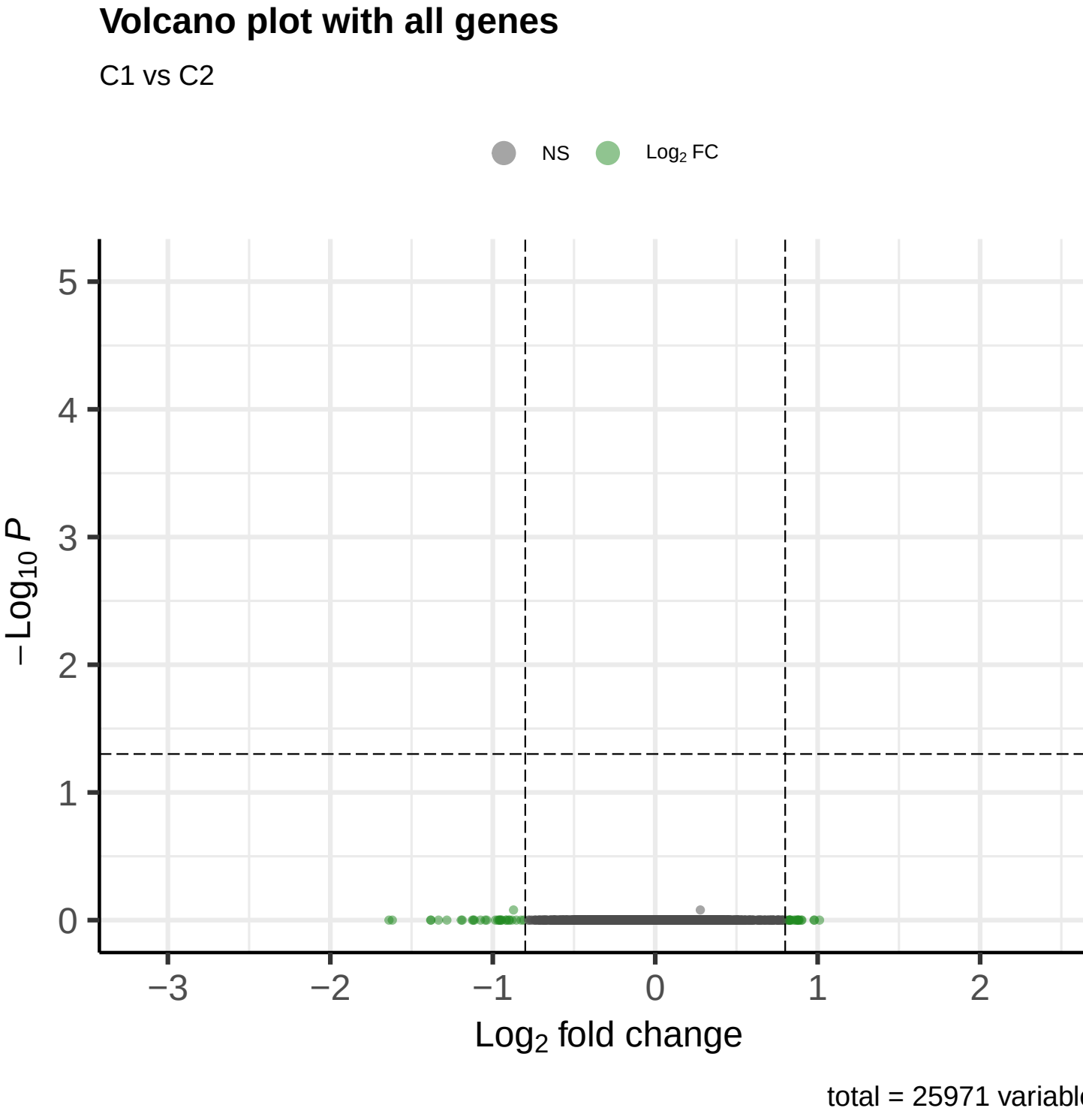


Figure 27: Volcano plot of DEG between C1 and C2 (from fig. 26)

Between C1 and C2, there is no DEG, because we can considered this cluster HOT and moderately infiltrated.

```
EnhancedVolcano(resultats$C2_vs_C3, lab = rownames(resultats$C2_vs_C3),
    pCutoff = 0.05, FCcutoff = 0.8, x = "logFC", y = "adj.P.Val",
    pointSize = 1.5, legendLabSize = 10, labSize = 3, title = "Volcano plot with all genes",
    subtitle = "C2 vs C3")
```

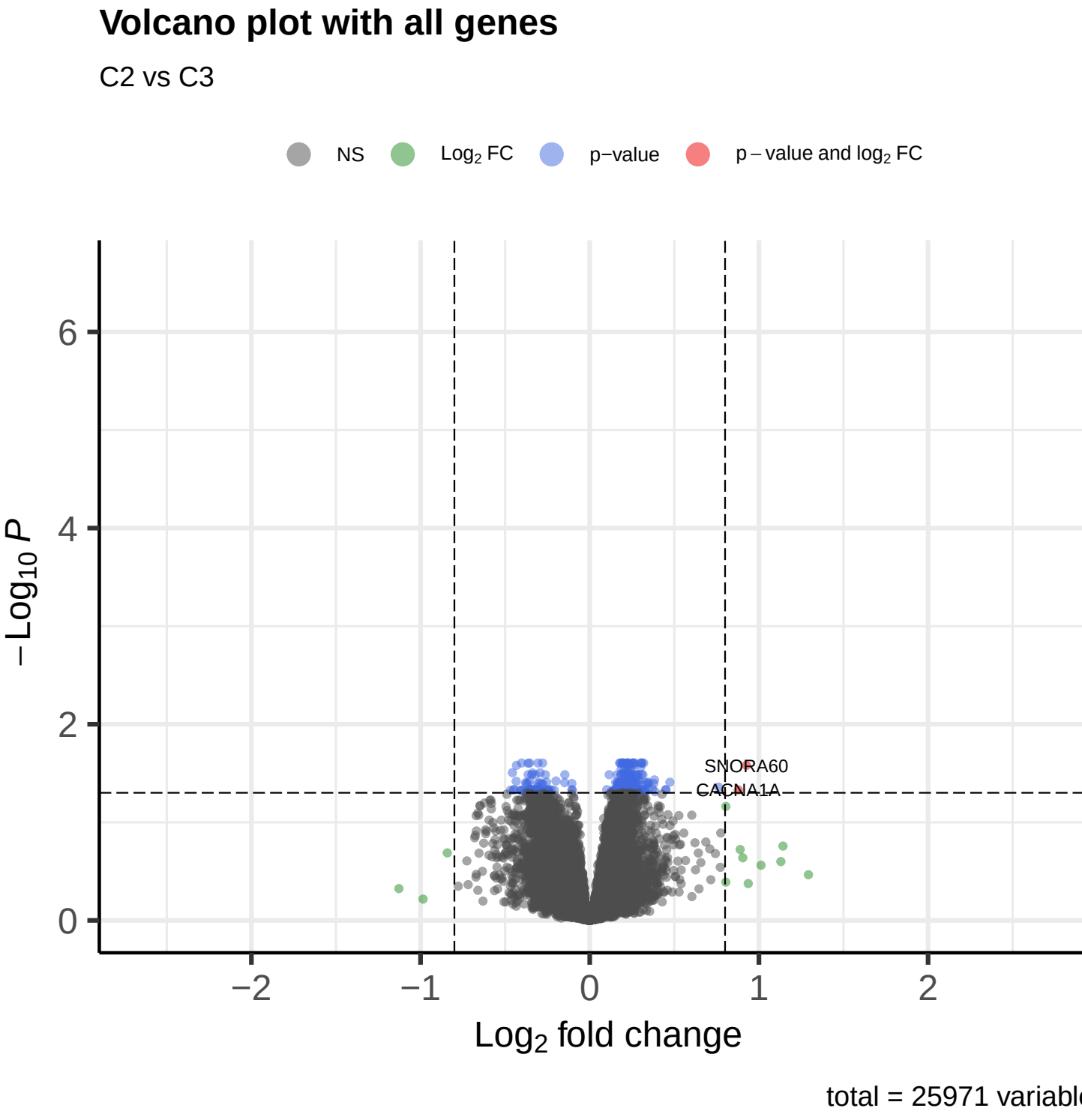


Figure 28: Volcano plot of DEG between C2 and C3 (from fig. 26)

```r
EnhancedVolcano(resultats$C1_vs_C3, lab = rownames(resultats$C1_vs_C3),
    pCutoff = 0.05, FCcutoff = 0.8, x = "logFC", y = "adj.P.Val",
    pointSize = 1.5, legendLabSize = 10, labSize = 3, title = "Volcano plot with all genes",
    subtitle = "C3 vs C1")
```

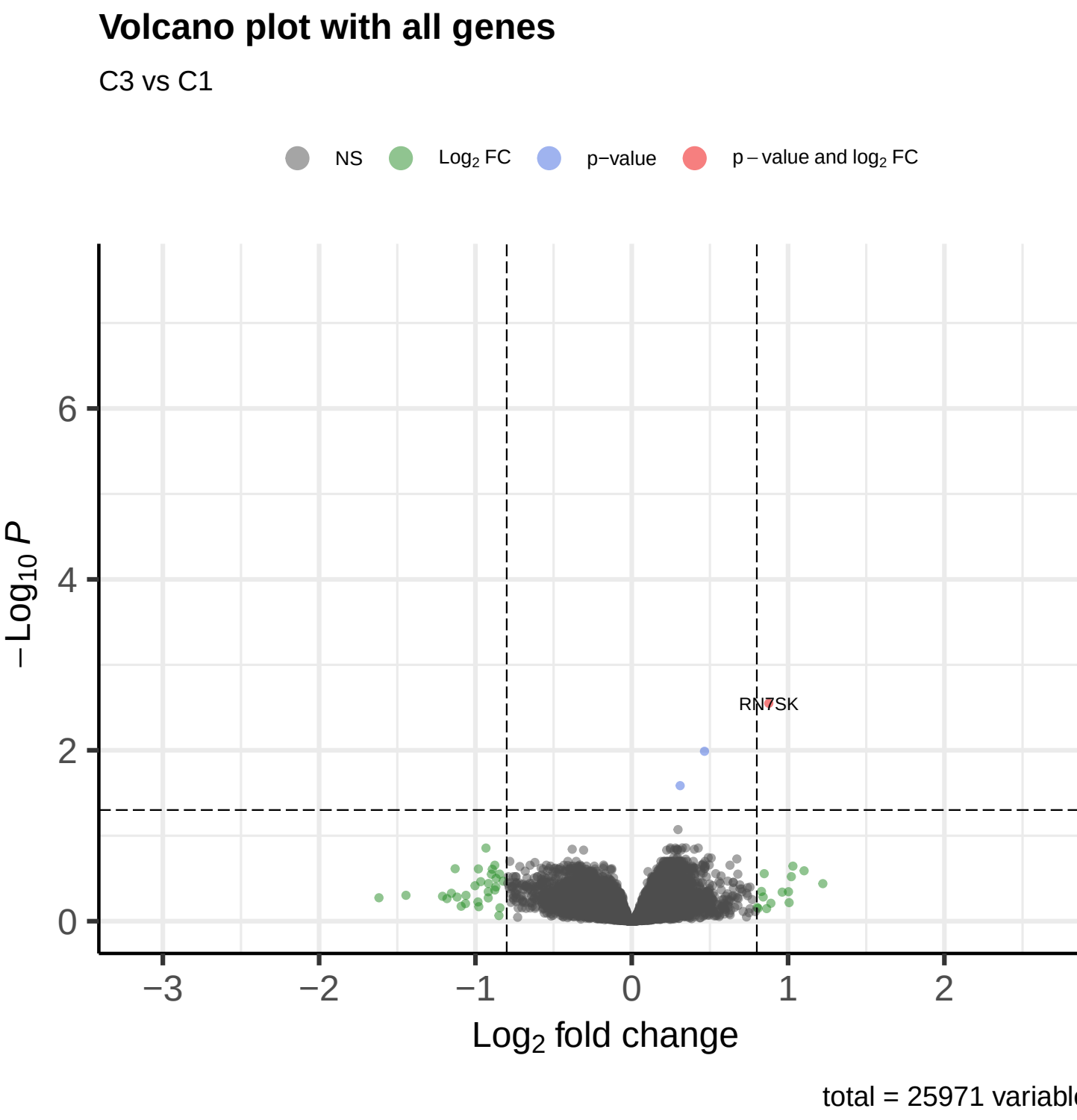


Figure 29: Volcano plot of DEG between C1 and C3 (from fig. 26)

Between C1 and C3, there is 2 DEG, so there is not really differences so the approach is'nt very good.

4) Clustering on the rows

a- Hierarchical clustering

```
Heatmap(as.matrix(df_imm_z_scores_73), cluster_rows = TRUE, cluster_columns = TRUE,
    cluster_column_slices = TRUE, clustering_distance_rows = "euclidean",
    clustering_method_rows = "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"))
```

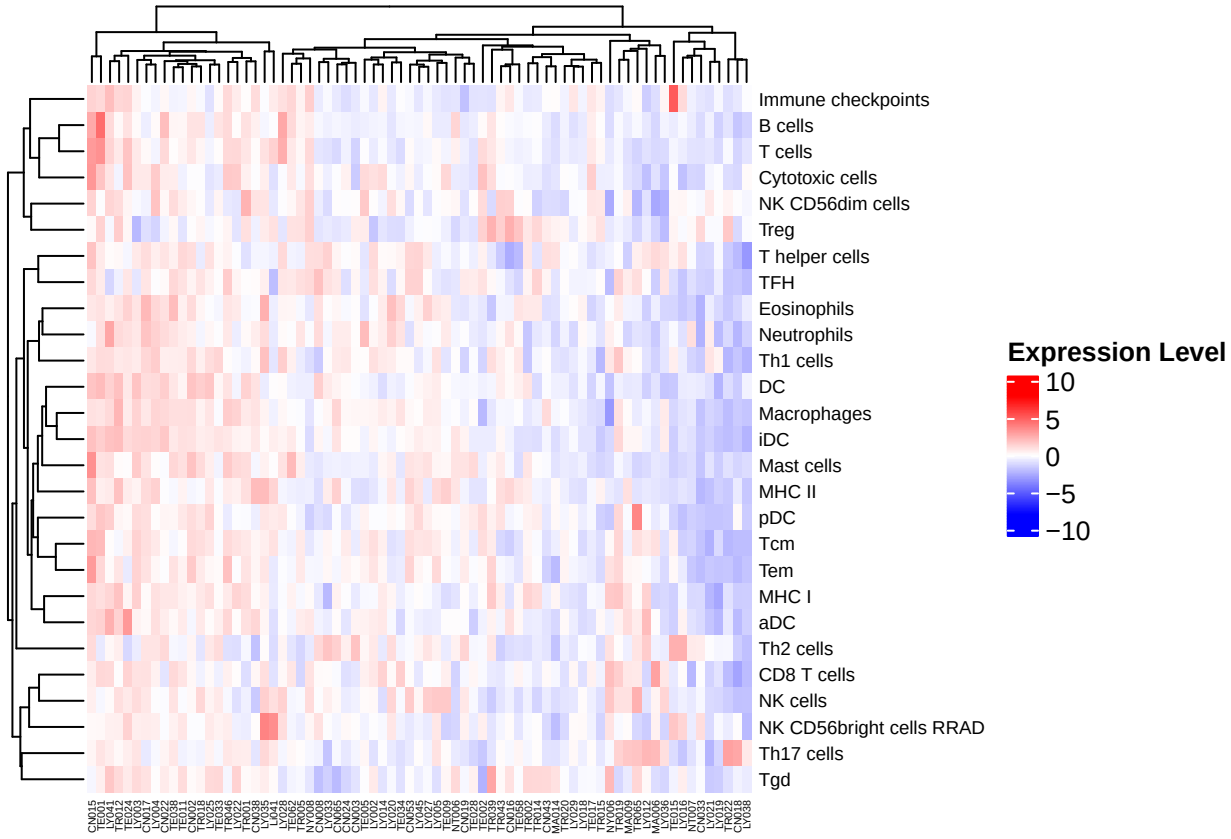


Figure 30: Heatmap and hierarchical clustering on relative immune cells expression (+ MHC) (n = 73)

We see that some immune cells are grouping, certain have a clear expression differences between patients.

b- Kmeans k = 2

```
set.seed(1)
Heatmap(
    as.matrix(df_imm_z_scores_73),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_rows = "euclidean",
    clustering_method_rows = "complete",
    show_column_dend = TRUE,
    row_km = 2, # 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")
)
```

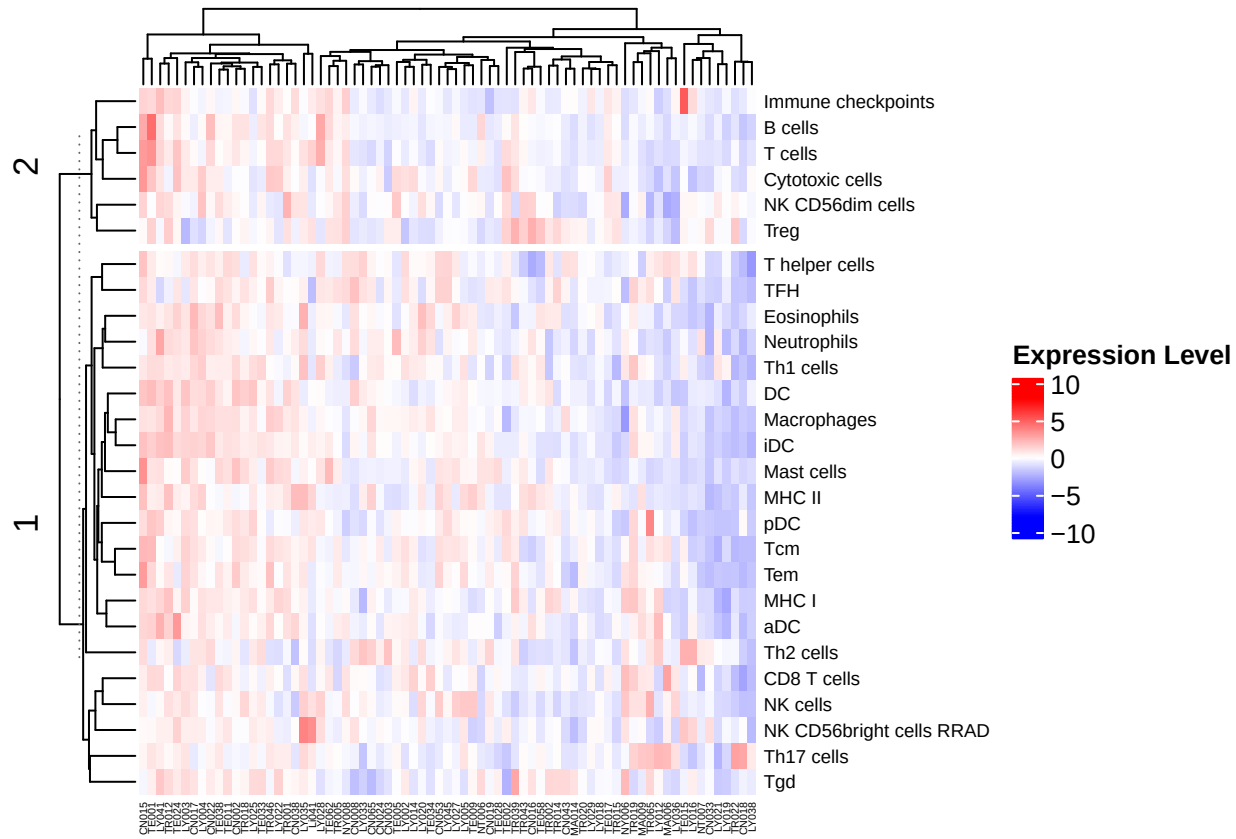


Figure 31: Heatmap with k-means clustering (k = 2) on relative immune cells expression (+ MHC) (n = 73)

We see that the C2 there is more a binary separation than C1.

c- Kmeans k = 3

```
set.seed(1)
Heatmap(
    as.matrix(df_imm_z_scores_73),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_rows = "euclidean",
    clustering_method_rows = "complete",
    show_column_dend = TRUE,
    row_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")
)
```

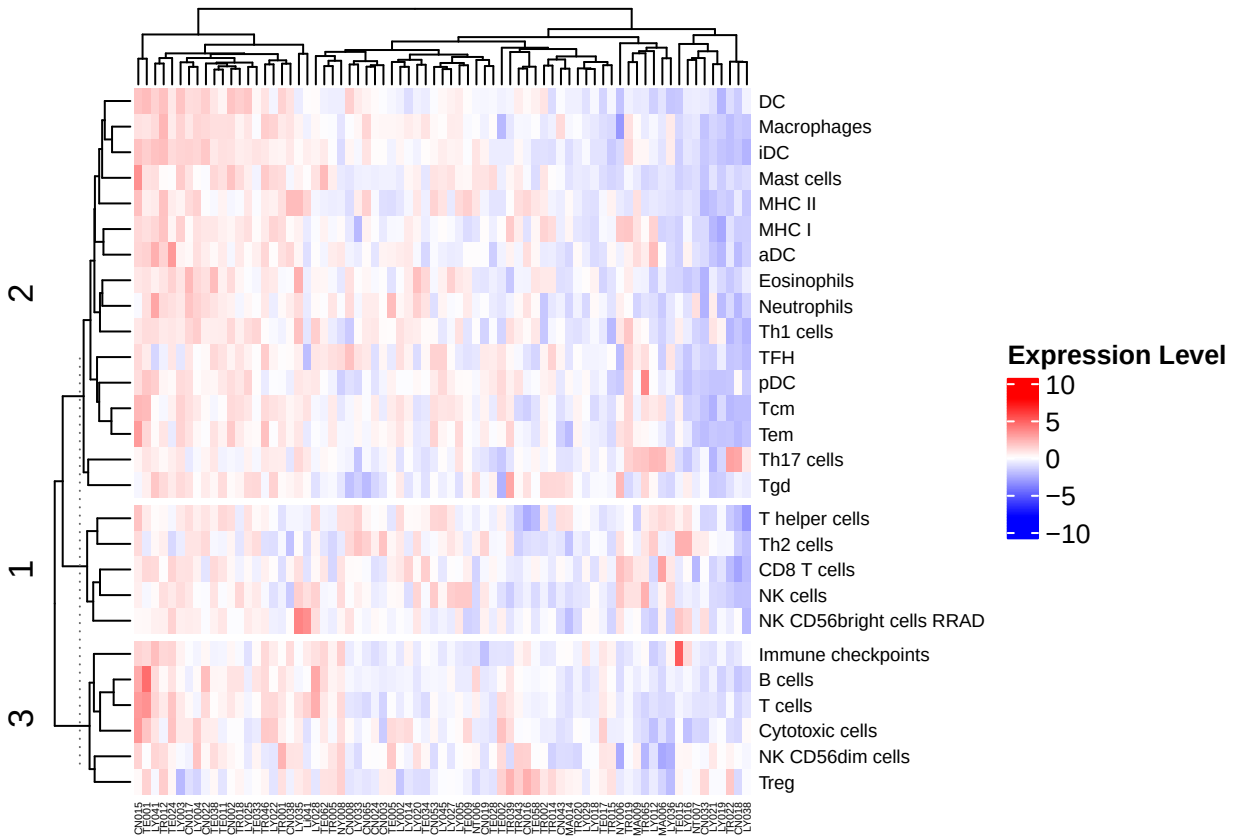


Figure 32: Heatmap with k-means clustering (k = 3) on relative immune cells expression (+ MHC) (n = 73)

Here, it is not very clear.

d- Kmeans k = 4

```
set.seed(1)
Heatmap(
    as.matrix(df_imm_z_scores_73),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_rows = "euclidean",
    clustering_method_rows = "complete",
    show_column_dend = TRUE,
    row_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")
)
```

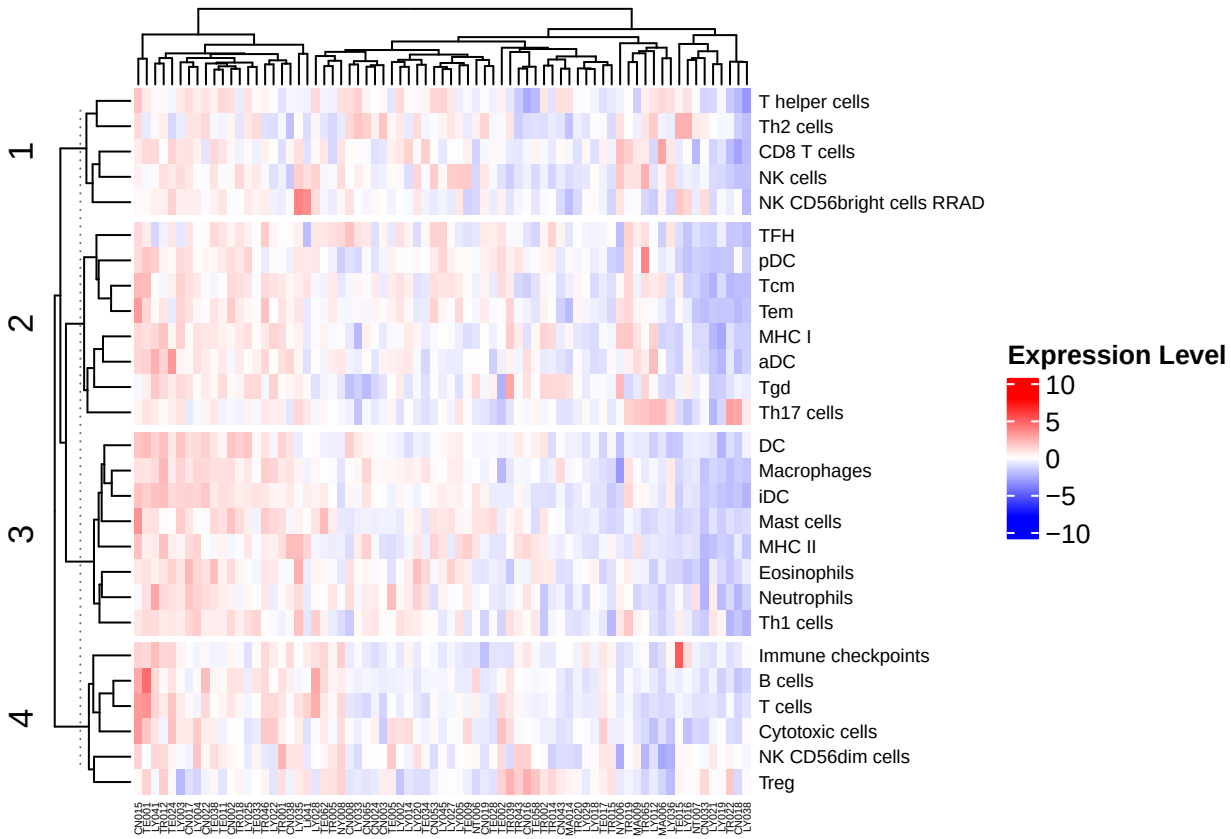


Figure 33: Heatmap with k-means clustering (k = 4) on relative immune cells expression (+ MHC) (n = 73)

3) Hierarchical clustering

```
Heatmap(as.matrix(df_imm_z_scores_73), cluster_rows = FALSE,
    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"))
```

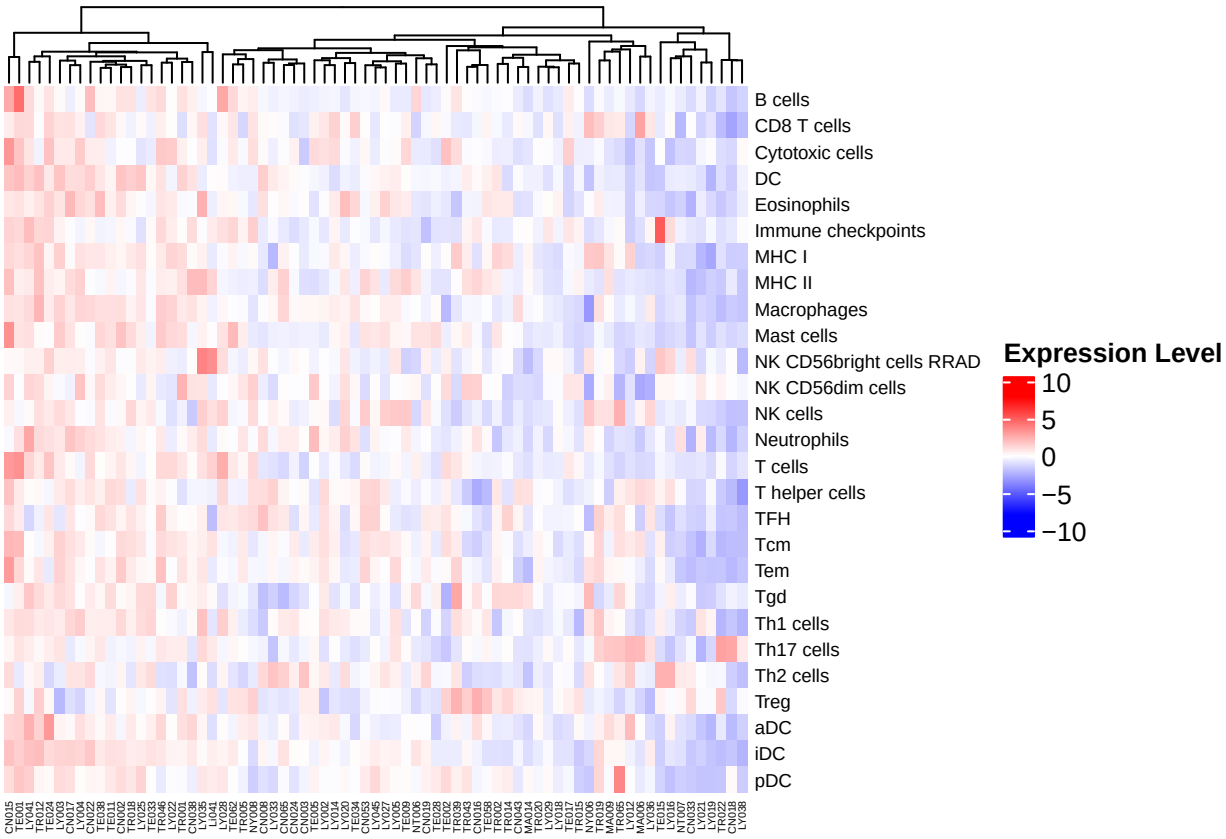


Figure 34: Heatmap and hierarchical clustering of relative immune cells expression (+ MHC) (n = 73)

This clustering show the COLD and HOT tumors. But at this moment, we see that some cell types have immunosuppressive effects so it is negative in hot tumor for example. So I can test by delete immunosupressive cells to see "real" hot and cold tumors.

4) Clustering conventional patients with survival data

```
df_complete_63 <- df_complete[, c("Signature", "CTA", colnames(df_complete)[colnames(df_complete) %in% c

# Average the expression between same immune cells types
# Take rows with immune cells signature from normalized data
df_avg_immune_sign_63 <- df_complete_63 %>%
  filter(Signature != "NA")

# Group by signature and calculate mean of expression values
df_avg_immune_sign_final_63 <- as.data.frame(df_avg_immune_sign_63 %>%
  select(-c(CTA)) %>%
  group_by(Signature) %>%
  summarise(across(where(is.numeric), \(x) mean(x, na.rm = TRUE))))
rownames(df_avg_immune_sign_final_63) <- df_avg_immune_sign_final_63$Signature
df_avg_immune_sign_final_63 <- df_avg_immune_sign_final_63[,-1]

# Compute z scores
df_imm_z_scores_63 <- t(scale(t(df_avg_immune_sign_final_63)))
#write.table(df_imm_z_scores_63, "../results/imm_sign_z_scores_63.tsv", sep = "\t", quote = F)

# Heatmap
set.seed(1)
Heatmap(
    as.matrix(df_imm_z_scores_63),
    cluster_rows = FALSE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete",
    show_column_dend = TRUE,
    column_km = 2, # 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")
)
```

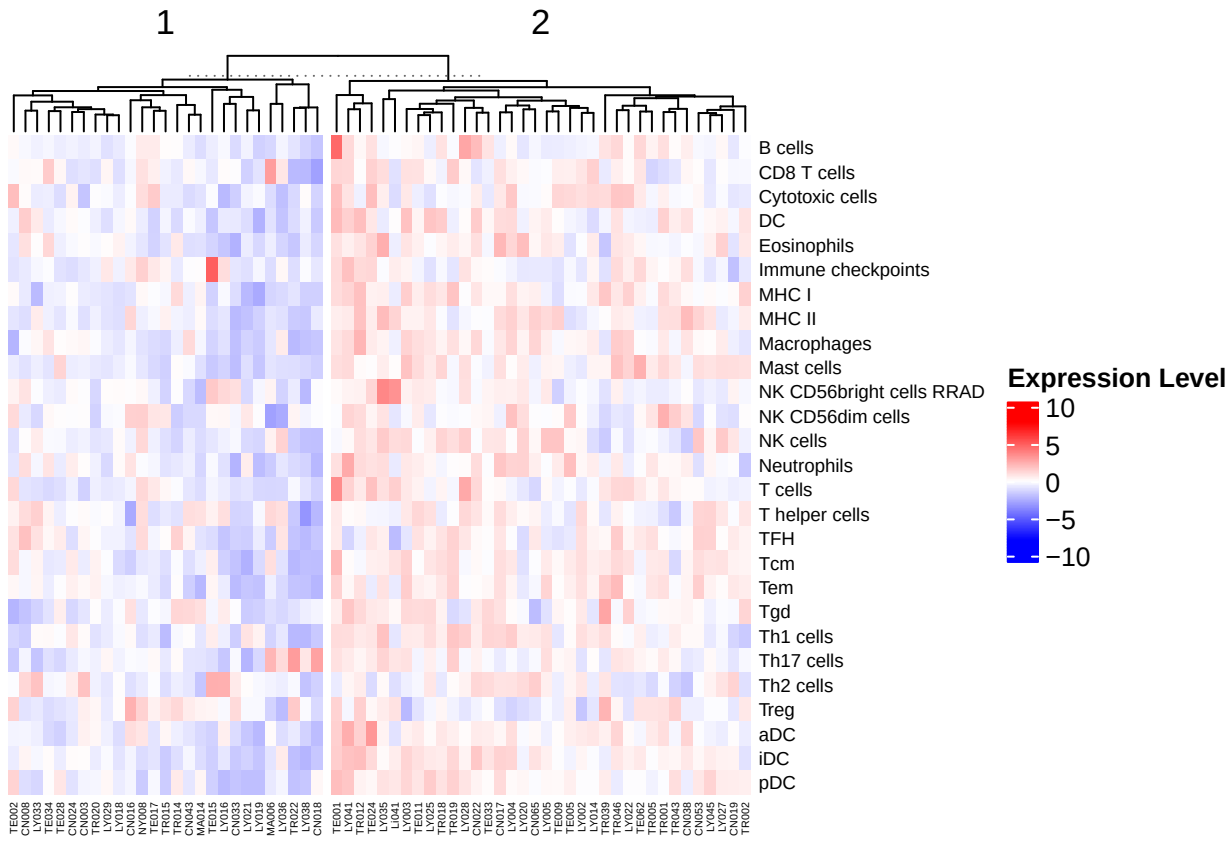


Figure 35: Heatmap with k-means clustering (k = 2) on relative immune cells expression (+ MHC) (n = 63)

V. Relative immune cells expression without immunosuppressive cells

We notice that some cells are immunosuppressive so I try to analyze another way the data. Actually, the "real" HOT tumors should expressed immune signatures but shouldn't express immunosuppressive cells signature.

1) K-means clustering with k = 4

This permit us to compare if the deletion of Treg and Immune Checkpoints have an impact on the clustering.

```
# Select data
heatmap_data_wo_immunosupp <- heatmap_data_mhc_all[!rownames(heatmap_data_mhc_all) %in% c("Treg", "Immu

# Heatmap
set.seed(1)
Heatmap(
    as.matrix(heatmap_data_wo_immunosupp),
    cluster_rows = FALSE,
    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")
)
```

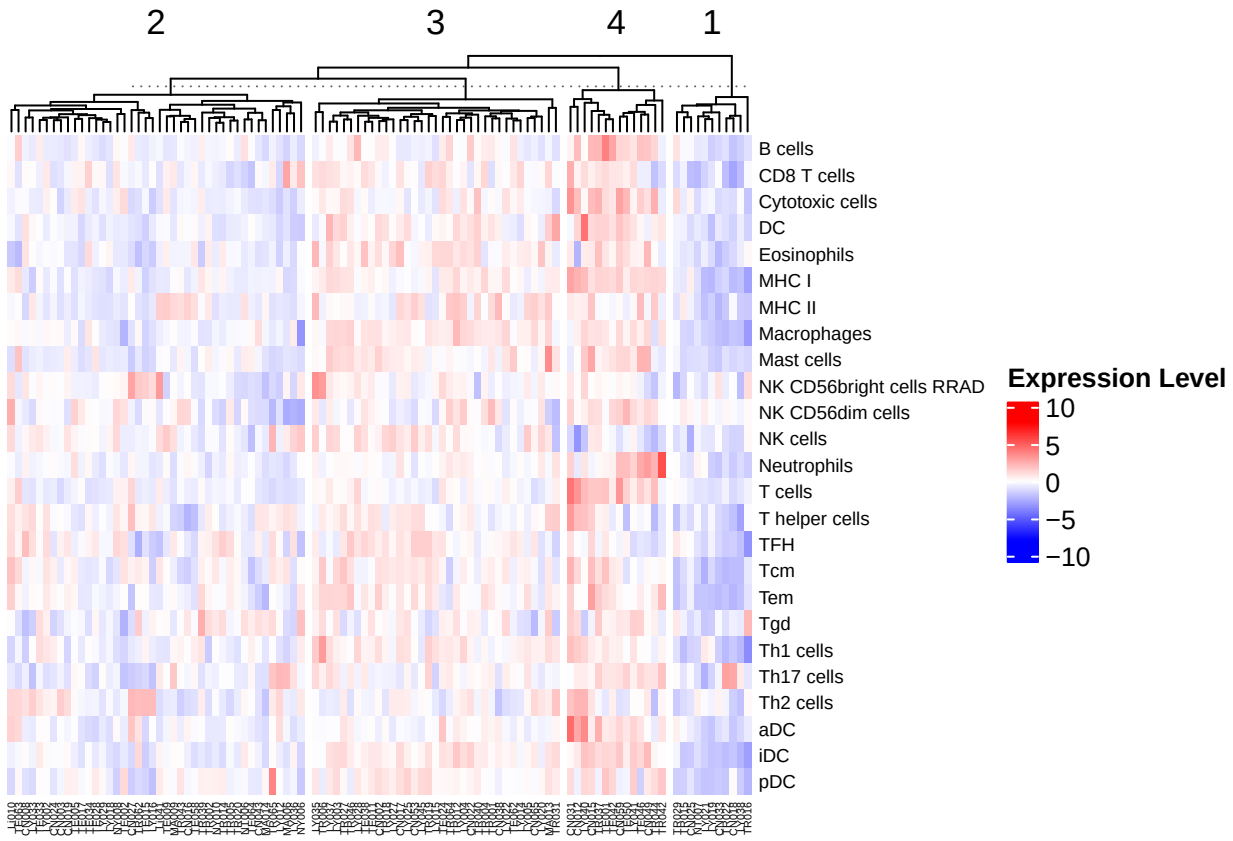


Figure 36: Heatmap of relative immune cells expression without immunosupressive cells (Treg and Immune checkpoints) (n = 102)

The extreme patients seems to be the same with immunosupp cells. Now, I want to see the expression for the 2 immunosupp cells.

2) Expression of immunosuppressive cells

```
# Select data
heatmap_data_immunosupp <- heatmap_data_mhc_all[rownames(heatmap_data_mhc_all) %in% c("Treg", "Immune cl

# Heatmap
set.seed(1)
Heatmap(
    as.matrix(heatmap_data_immunosupp),
    cluster_rows = FALSE,
    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")
)
```

We see that some patients that I have considered HOT expressed these immunosupp cells. So we can perform survival analysis to see the real impact in the individuals survival (see script 6).

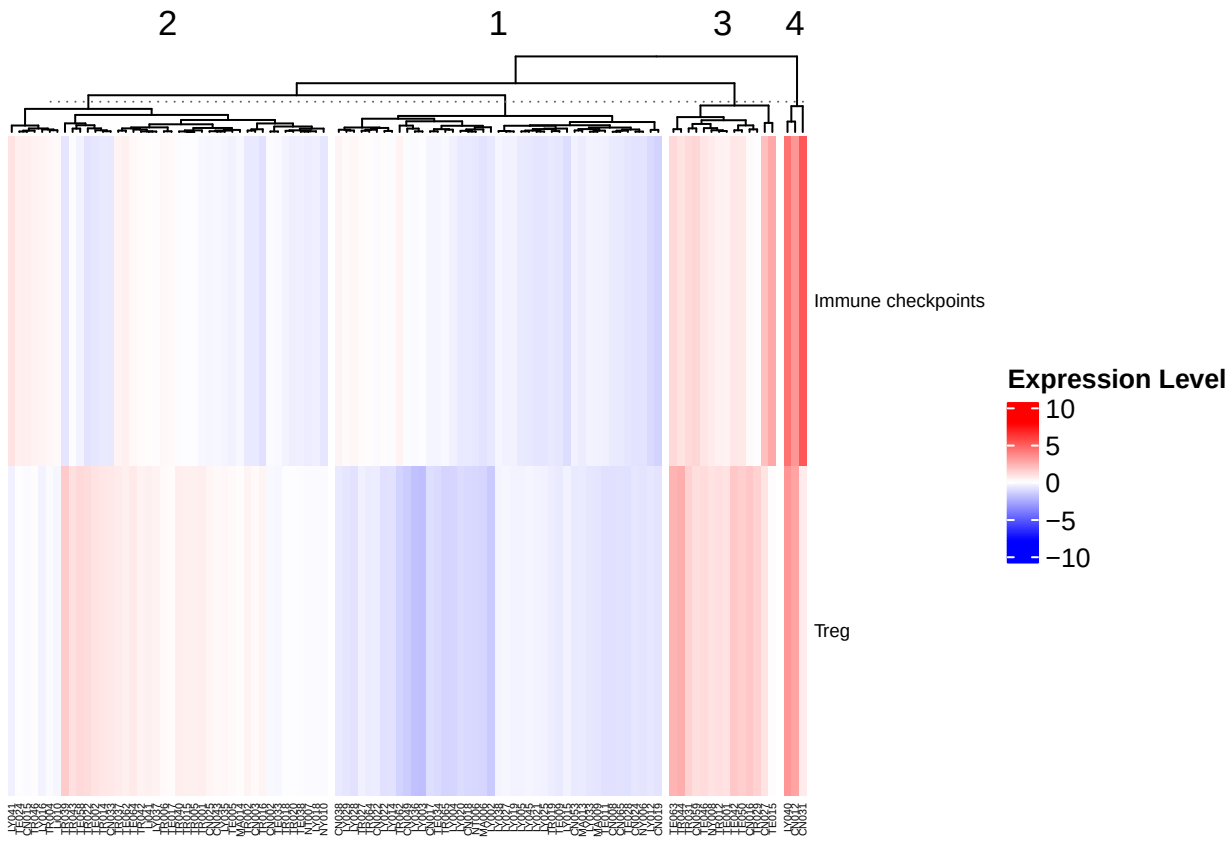


Figure 37: Heatmap of Treg and Immune checkpoints expression

VI. Adding metadata

In this section, we are adding metadata to the heatmap to further analyze the clustering results based on patient characteristics. The goal is to visualize and interpret the relationship between patient subtypes, histology, genetic mutations, and other relevant factors.

1) Adding metadata on heatmap with all patients

```
# Select colors
histology_colors <- c("benign" = "#E74C3C",
                      "dedifferentiated" = "#3498DB",
                      "G1" = "#2ECC71",
                      "G2" = "#F1C40F",
                      "G3" = "#9B59B6",
                      "N/A" = "#000000")

subtype_colors <- c("E1" = "#E74C3C",
                    "E2" = "#3498DB")

methy_colors <- c("M1" = "#F1C40F",
                  "M2" = "#9B59B6",
                  "M3" = "#2ECC71")

mir_colors <- c("mir14q32High" = "#E74C3C",
                "mir14q32Low" = "#3498DB")

idh_colors <- c("wt" = "#2ECC71",
                "mut" = "#F1C40F")

tp53_colors <- c("frameshift deletion" = "#F1C40F",
                 "nonsynonymous SNV" = "#9B59B6",
                 "wt" = "#2ECC71")

multiomics_colors <- c("C1" = "#F1C40F",
                       "C2" = "#9B59B6",
                       "C3" = "#2ECC71",
                       "C4" = "#E74C3C",
                       "C5" = "#3498DB",
                       "C6" = "#1B5E20")

# Heatmap
set.seed(1)
heatmap_anno_all <- Heatmap(
    as.matrix(heatmap_data_mhc_all),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    show_column_dend = TRUE,
    show_row_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"),
    top_annotation = columnAnnotation(
      Histology = df_metadata$Histology,
      Subtype = df_metadata$EXP.subtype,
      Methy_subtype = df_metadata$METH.subtype,
      miRNA_subtype = df_metadata$MIR.simplifiedSubtype,
      Multiomics_subtype = df_metadata$MOM.subtype,
      IDH1_mut = ifelse(df_metadata$IDH1.AAmut != "wt", "mut", df_metadata$IDH1.AAmut),
      IDH2_mut = ifelse(df_metadata$IDH2.AAmut != "wt", "mut", df_metadata$IDH2.AAmut),
      TP53 = df_metadata$TP53,
      col = list(Histology = histology_colors,
                 Subtype = subtype_colors,
                 Methy_subtype = methy_colors,
                 miRNA_subtype = mir_colors,
                 Multiomics_subtype = multiomics_colors,
                 IDH1_mut = idh_colors,
                 IDH2_mut = idh_colors,
                 TP53 = tp53_colors))
)
set.seed(1)
heatmap_anno_all <- draw(heatmap_anno_all)
```

```
# Store clusters
# Take col indexes
indiv_clust <- column_order(heatmap_anno_all)

# Create table with indiv names
df_indiv_clusters_hm_anno <- data.frame(
  Cluster = c(rep(1, length(indiv_clust$`1`)),
              rep(2, length(indiv_clust$`2`)),
              rep(3, length(indiv_clust$`3`)),
              rep(4, length(indiv_clust$`4`))),
  Patient = c(colnames(heatmap_data_mhc_all)[indiv_clust$`1`],
              colnames(heatmap_data_mhc_all)[indiv_clust$`2`],
              colnames(heatmap_data_mhc_all)[indiv_clust$`3`],
              colnames(heatmap_data_mhc_all)[indiv_clust$`4`])
)

# Save
#write.table(df_indiv_clusters_hm_anno, file = "../results/clusters_indiv/clusters_all_indiv_mhc.tsv",
```

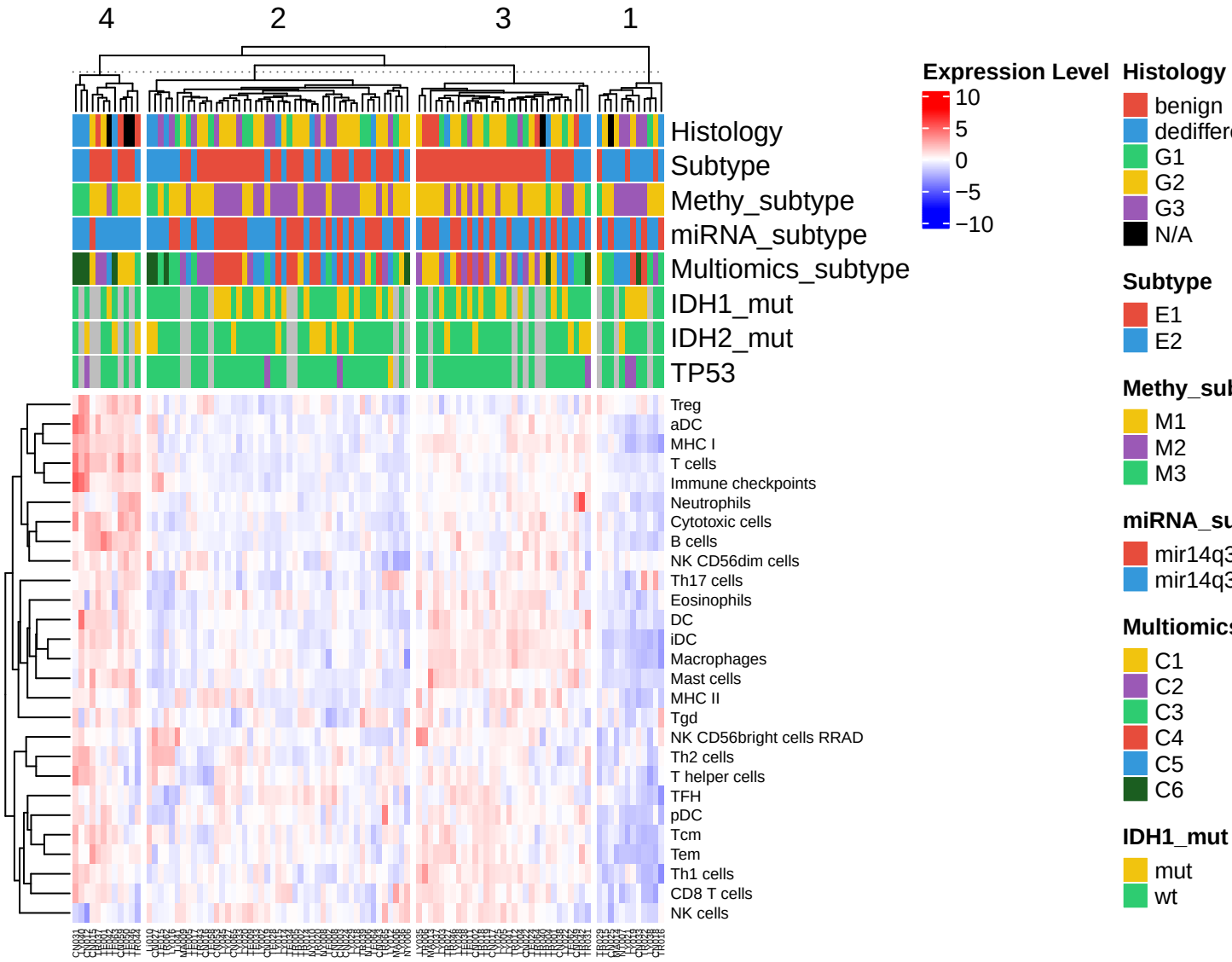


Figure 38: Heatmap (from fig. 19) with metadata (n = 102)

We see in the cluster 4 dedifferentiated, G2 and benign histology. In C1, we retrieve G3, G2 and some dediff. In C2 and C3, there is more G2 and G1 with some benign and dediff. We kwnow that dediff are generally bad diagnosis but here we see that they are infiltrated. So this could be induce errors.

```
Heatmap(
    as.matrix(heatmap_data_mhc_all),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    show_column_dend = TRUE,
    show_row_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"),
    top_annotation = columnAnnotation(
      Histology = df_metadata$Histology,
      col = list(Histology = histology_colors))
)
```

2) Adding metadata on heatmap with conventional patients

a- k = 2

```
# Heatmap
set.seed(1)
heatmap_anno_conv <- Heatmap(
    as.matrix(df_imm_z_scores_73),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    show_column_dend = TRUE,
    show_row_dend = TRUE,
    column_km = 2, # 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(
      Histology = df_metadata_conv$Histology,
      Subtype = df_metadata_conv$EXP.subtype,
      Methy_subtype = df_metadata_conv$METH.subtype,
      miRNA_subtype = df_metadata_conv$MIR.simplifiedSubtype,
      Multiomics_subtype = df_metadata_conv$MOM.subtype,
      IDH1_mut = ifelse(df_metadata_conv$IDH1.AAmut != "wt", "mut", df_metadata_conv$IDH1.AAmut),
      IDH2_mut = ifelse(df_metadata_conv$IDH2.AAmut != "wt", "mut", df_metadata_conv$IDH2.AAmut),
      TP53 = df_metadata_conv$TP53,
      col = list(Histology = histology_colors,
```

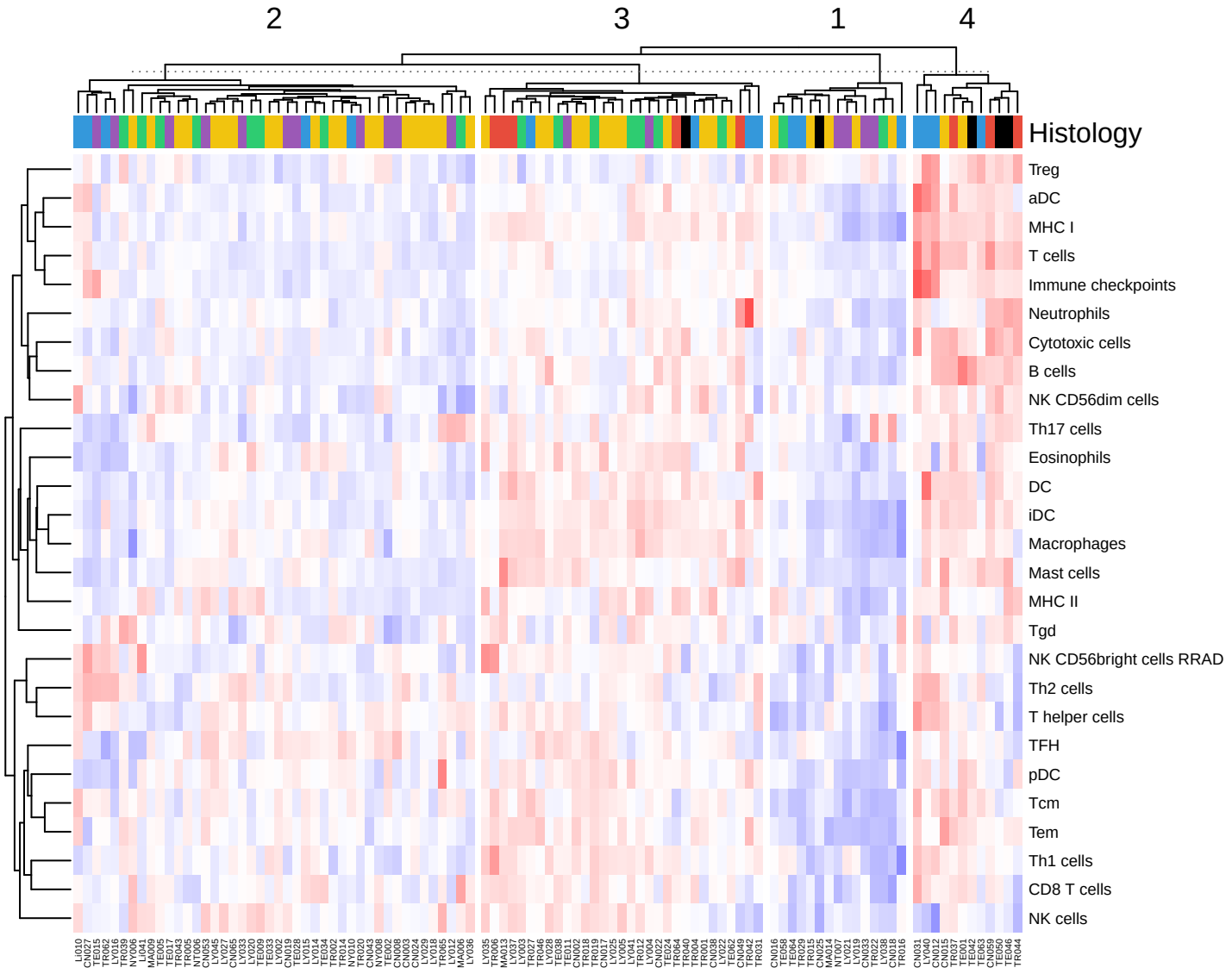


Figure 39: Heatmap (from fig. 19) with grade data (n = 102)

```
                  Subtype = subtype_colors,
                  Methy_subtype = methy_colors,
                  miRNA_subtype = mir_colors,
                  Multiomics_subtype = multiomics_colors,
                  IDH1_mut = idh_colors,
                  IDH2_mut = idh_colors,
                  TP53 = tp53_colors))
)
set.seed(1)
heatmap_anno_conv <- draw(heatmap_anno_conv)
```

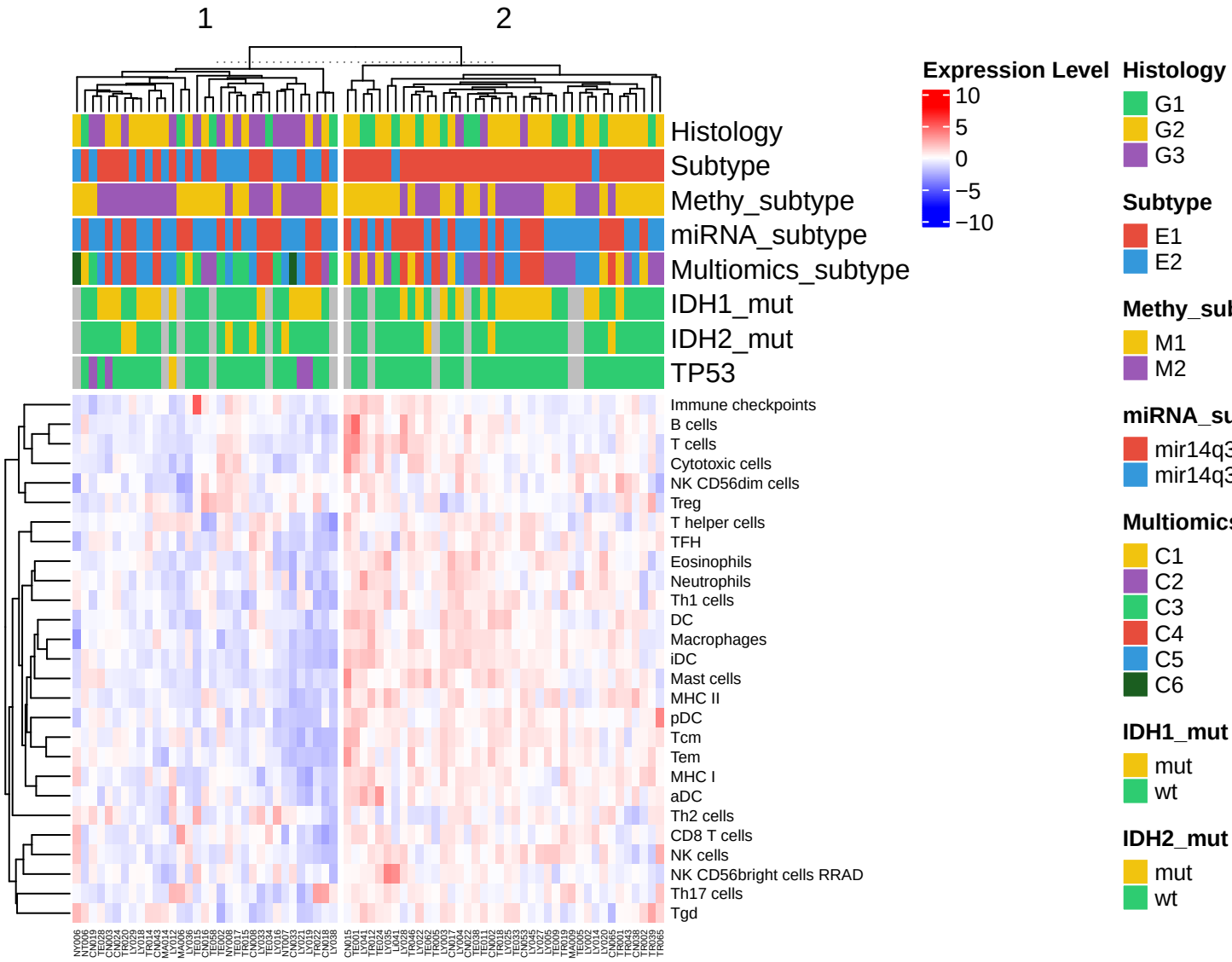


Figure 40: Heatmap (from fig. 22) with metadata (n = 73)

```
# Store clusters
# Take col indexes
indiv_clust <- column_order(heatmap_anno_conv)
```

```r
# Create table with indiv names
df_indiv_clusters_hm_anno_63 <- data.frame(
  Cluster = c(rep(1, length(indiv_clust$`1`)),
              rep(2, length(indiv_clust$`2`))),
  Patient = c(colnames(df_imm_z_scores_73)[indiv_clust$`1`],
              colnames(df_imm_z_scores_73)[indiv_clust$`2`])
)

# Save
#write.table(df_indiv_clusters_hm_anno_63, file = "../results/clusters_indiv/clusters_conv_indiv_mhc.ts
```

We see 2 distinct clusers but the limit is not binary. However, we see that there is more G3 in C1 than C2.

b- k = 3

```
# Heatmap
set.seed(1)
heatmap_anno_conv <- Heatmap(
    as.matrix(df_imm_z_scores_73),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    show_column_dend = TRUE,
    show_row_dend = TRUE,
    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(
      Histology = df_metadata_conv$Histology,
      Subtype = df_metadata_conv$EXP.subtype,
      Methy_subtype = df_metadata_conv$METH.subtype,
      miRNA_subtype = df_metadata_conv$MIR.simplifiedSubtype,
      Multiomics_subtype = df_metadata_conv$MOM.subtype,
      IDH1_mut = ifelse(df_metadata_conv$IDH1.AAmut != "wt", "mut", df_metadata_conv$IDH1.AAmut),
      IDH2_mut = ifelse(df_metadata_conv$IDH2.AAmut != "wt", "mut", df_metadata_conv$IDH2.AAmut),
      TP53 = df_metadata_conv$TP53,
      col = list(Histology = histology_colors,
                 Subtype = subtype_colors,
                 Methy_subtype = methy_colors,
                 miRNA_subtype = mir_colors,
                 Multiomics_subtype = multiomics_colors,
                 IDH1_mut = idh_colors,
                 IDH2_mut = idh_colors,
                 TP53 = tp53_colors))
)
set.seed(1)
heatmap_anno_conv <- draw(heatmap_anno_conv)
```

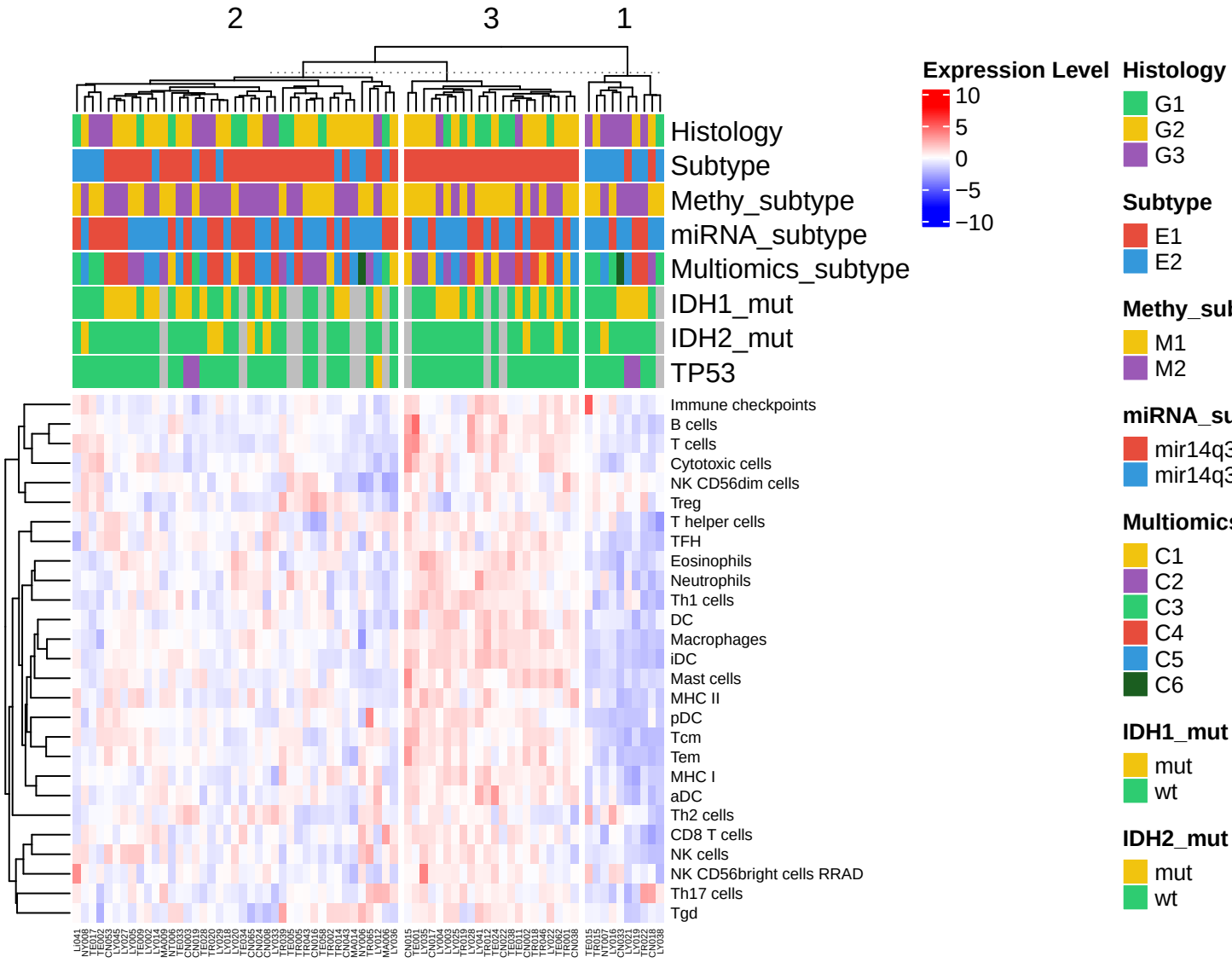


Figure 41: Heatmap (from fig. 26) with metadata (n = 73)

We see 2 distinct clusters, the limit is more clear than k = 2. We see, one cluster COLD, another HOT and the last is more or less infiltrated.

3) Adding metadata on heatmap with conventional patients that have survival data

```
metadata_surv_conv <- subset(df_metadata_conv, df_metadata_conv$OS.delay != "NA")
set.seed(1)
Heatmap(
    as.matrix(df_imm_z_scores_63),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    show_column_dend = TRUE,
    show_row_dend = TRUE,
    column_km = 2, # Number of clusters
    column_km_repeats = 20,
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
    border = NA,
    show_column_names = TRUE,
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 7),
    heatmap_legend_param = list(title = "Expression Level"),
    top_annotation = columnAnnotation(
      Histology = metadata_surv_conv$Histology,
      Subtype = metadata_surv_conv$EXP.subtype,
      Methy_subtype = metadata_surv_conv$METH.subtype,
      miRNA_subtype = metadata_surv_conv$MIR.simplifiedSubtype,
      Multiomics_subtype = metadata_surv_conv$MOM.subtype,
      IDH1_mut = ifelse(metadata_surv_conv$IDH1.AAmut != "wt", "mut", metadata_surv_conv$IDH1.AAmut),
      IDH2_mut = ifelse(metadata_surv_conv$IDH2.AAmut != "wt", "mut", metadata_surv_conv$IDH2.AAmut),
      TP53 = metadata_surv_conv$TP53,
      col = list(Histology = histology_colors,
                 Subtype = subtype_colors,
                 Methy_subtype = methy_colors,
                 miRNA_subtype = mir_colors,
                 Multiomics_subtype = multiomics_colors,
                 IDH1_mut = idh_colors,
                 IDH2_mut = idh_colors,
                 TP53 = tp53_colors))
)
```

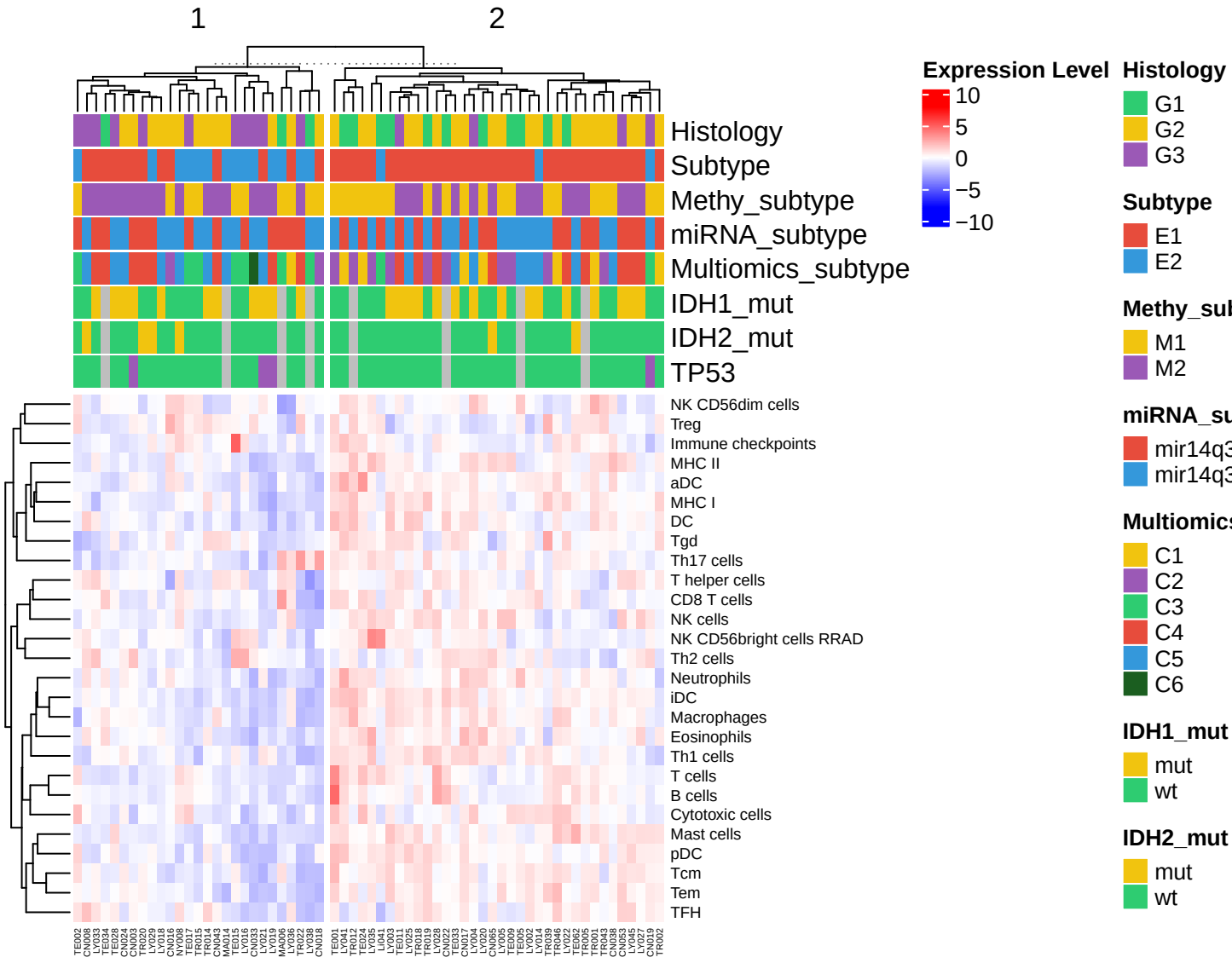


Figure 42: Heatmap with k=2 with metadata (n = 63)

4) Adding CTA heatmaps metadata

a- Figure 6

```
Heatmap(as.matrix(data_cta_63), 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 with significant su
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"),
    top_annotation = columnAnnotation(Histology = metadata_surv_conv$Histology,
        Subtype = metadata_surv_conv$EXP.subtype, Methy_subtype = metadata_surv_conv$METH.subtype,
        miRNA_subtype = metadata_surv_conv$MIR.simplifiedSubtype,
        Multiomics_subtype = metadata_surv_conv$MOM.subtype,
        IDH1_mut = ifelse(metadata_surv_conv$IDH1.AAmut != "wt",
            "mut", metadata_surv_conv$IDH1.AAmut), IDH2_mut = ifelse(metadata_surv_conv$IDH2.AAmut !=
            "wt", "mut", metadata_surv_conv$IDH2.AAmut), TP53 = metadata_surv_conv$TP53,
        col = list(Histology = histology_colors, Subtype = subtype_colors,
            Methy_subtype = methy_colors, miRNA_subtype = mir_colors,
            Multiomics_subtype = multiomics_colors, IDH1_mut = idh_colors,
            IDH2_mut = idh_colors, TP53 = tp53_colors)))
```

```
Heatmap(as.matrix(data_cta_63), 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 with significant su
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"),
    top_annotation = columnAnnotation(Histology = metadata_surv_conv$Histology,
        col = list(Histology = histology_colors)))
```

b- Figure 7

```
Heatmap(as.matrix(data_selected_CTA_conv_63), cluster_rows = TRUE,
    cluster_columns = TRUE, cluster_column_slices = TRUE, clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete", col = colorRamp2(seq(-8,
        8, length.out = 100), colors), border = NA, show_column_names = TRUE,
    show_row_names = TRUE, column_title = "Heatmap of selected CTA with significant survival impact",
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"),
    top_annotation = columnAnnotation(Histology = metadata_surv_conv$Histology,
        Subtype = metadata_surv_conv$EXP.subtype, Methy_subtype = metadata_surv_conv$METH.subtype,
        miRNA_subtype = metadata_surv_conv$MIR.simplifiedSubtype,
        Multiomics_subtype = metadata_surv_conv$MOM.subtype,
        IDH1_mut = ifelse(metadata_surv_conv$IDH1.AAmut != "wt",
            "mut", metadata_surv_conv$IDH1.AAmut), IDH2_mut = ifelse(metadata_surv_conv$IDH2.AAmut !=
            "wt", "mut", metadata_surv_conv$IDH2.AAmut), TP53 = metadata_surv_conv$TP53,
```

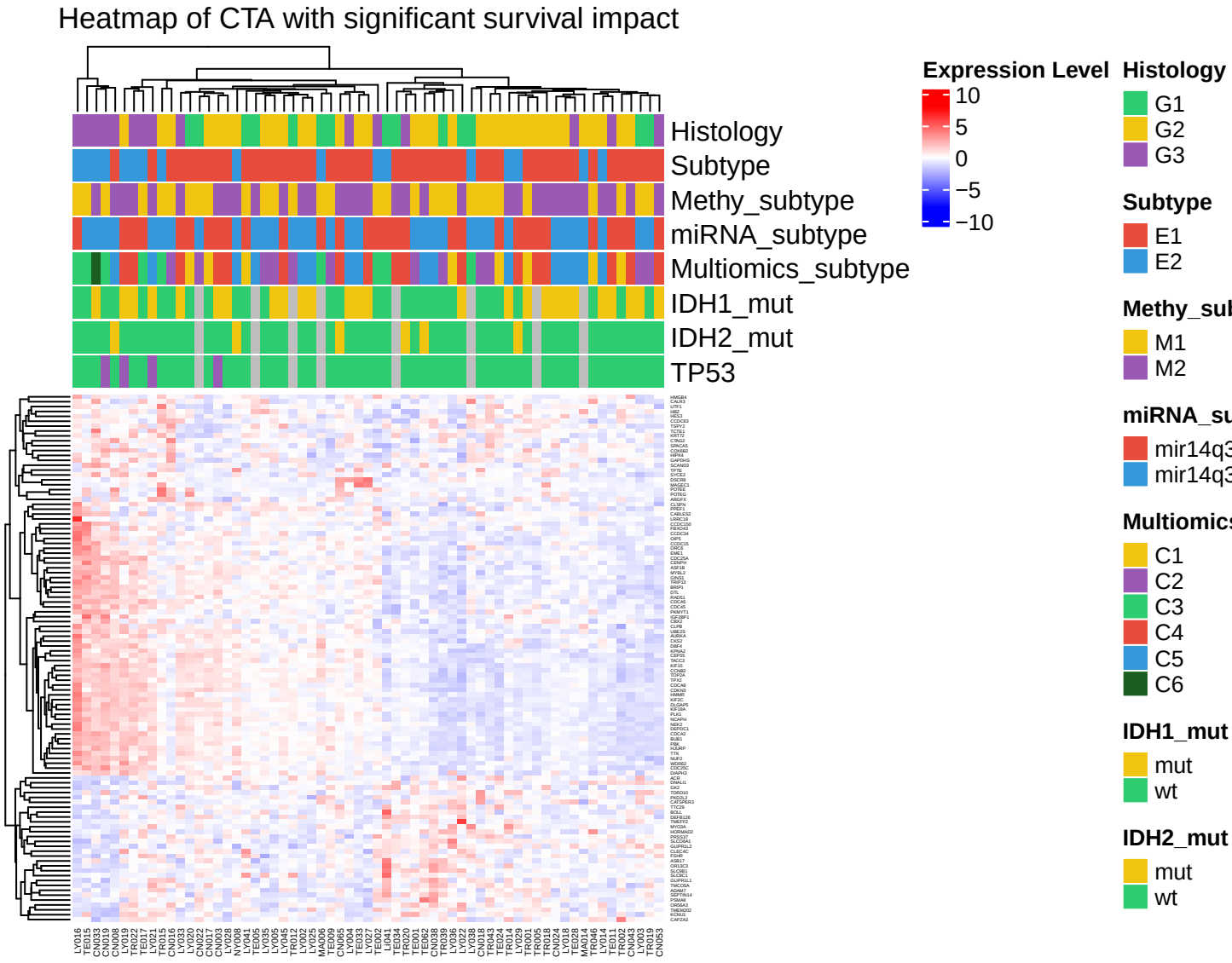


Figure 43: Significant CTAs heatmap (from fig.6) with metadata (n = 63)

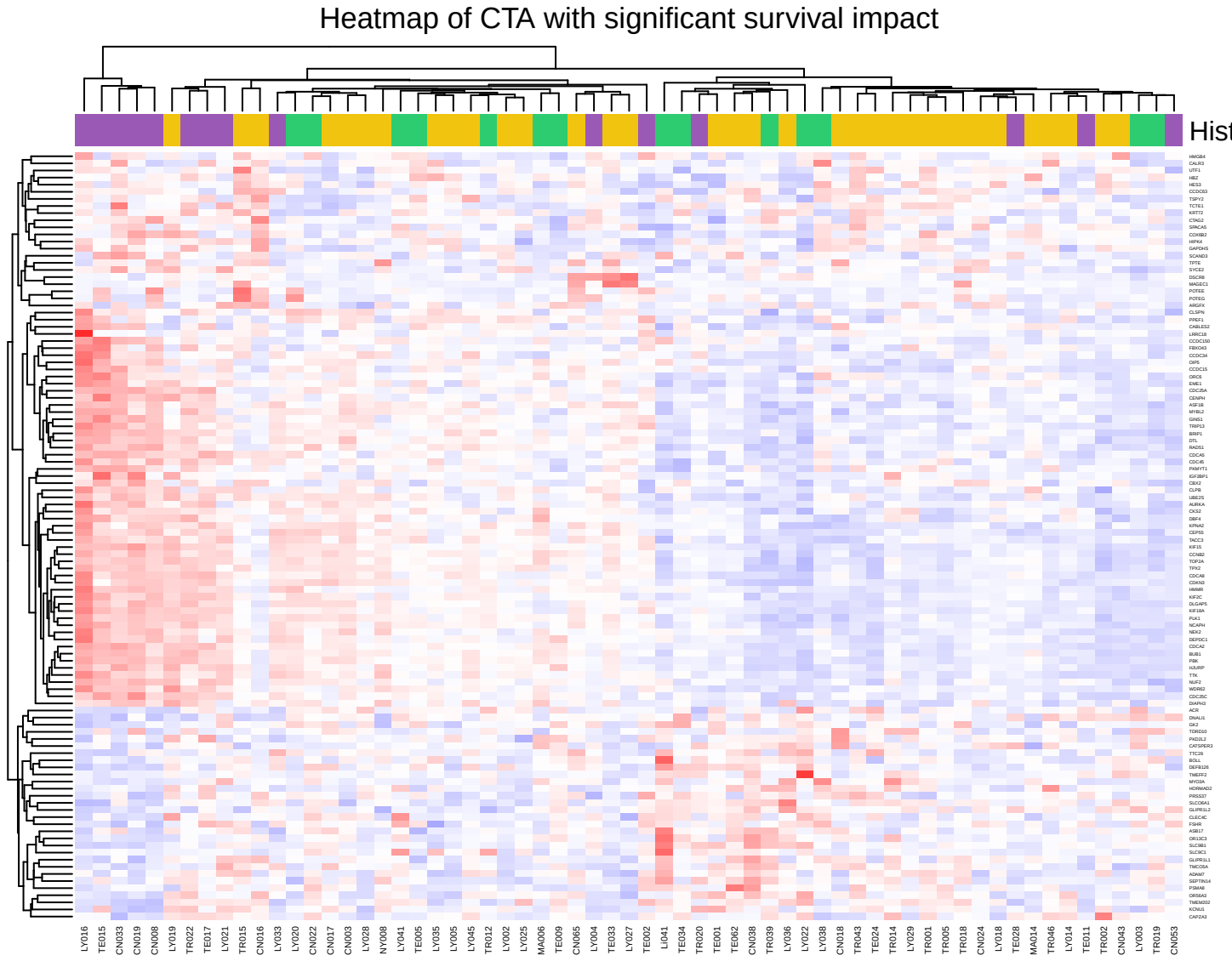


Figure 44: Significant CTAs heatmap (from fig.6) with grade data (n = 63)

```
        col = list(Histology = histology_colors, Subtype = subtype_colors,
            Methy_subtype = methy_colors, miRNA_subtype = mir_colors,
            Multiomics_subtype = multiomics_colors, IDH1_mut = idh_colors,
            IDH2_mut = idh_colors, TP53 = tp53_colors)))
```

Heatmap of selected CTA with significant survival impact

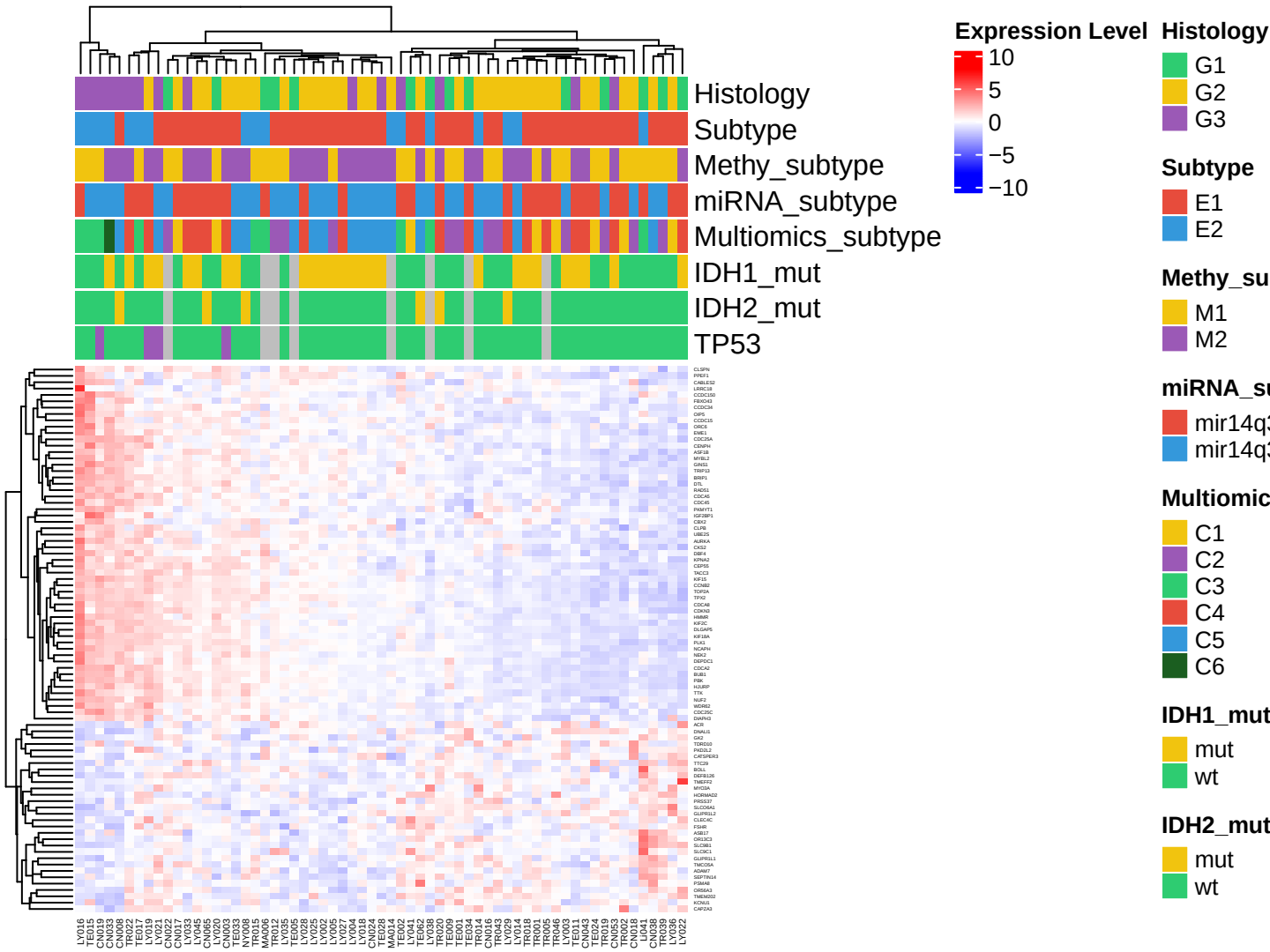


Figure 45: Selected significant CTAs heatmap (from fig.7) with metadata (n = 63)

c- Figure 8

```
Heatmap(as.matrix(data_non_selected_CTA_63), cluster_rows = TRUE,
    cluster_columns = TRUE, cluster_column_slices = TRUE, clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete", col = colorRamp2(seq(-8,
        8, length.out = 100), colors), border = NA, show_column_names = TRUE,
    show_row_names = TRUE, column_title = "Heatmap of other selected CTA with significant survival impa
```

```
    column_names_gp = gpar(fontsize = 4), row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level"),
    top_annotation = columnAnnotation(Histology = metadata_surv_conv$Histology,
        Subtype = metadata_surv_conv$EXP.subtype, Methy_subtype = metadata_surv_conv$METH.subtype,
        miRNA_subtype = metadata_surv_conv$MIR.simplifiedSubtype,
        Multiomics_subtype = metadata_surv_conv$MOM.subtype,
        IDH1_mut = ifelse(metadata_surv_conv$IDH1.AAmut != "wt",
            "mut", metadata_surv_conv$IDH1.AAmut), IDH2_mut = ifelse(metadata_surv_conv$IDH2.AAmut !=
            "wt", "mut", metadata_surv_conv$IDH2.AAmut), TP53 = metadata_surv_conv$TP53,
        col = list(Histology = histology_colors, Subtype = subtype_colors,
            Methy_subtype = methy_colors, miRNA_subtype = mir_colors,
            Multiomics_subtype = multiomics_colors, IDH1_mut = idh_colors,
            IDH2_mut = idh_colors, TP53 = tp53_colors)))
```

ther selected CTA with significant survival impact

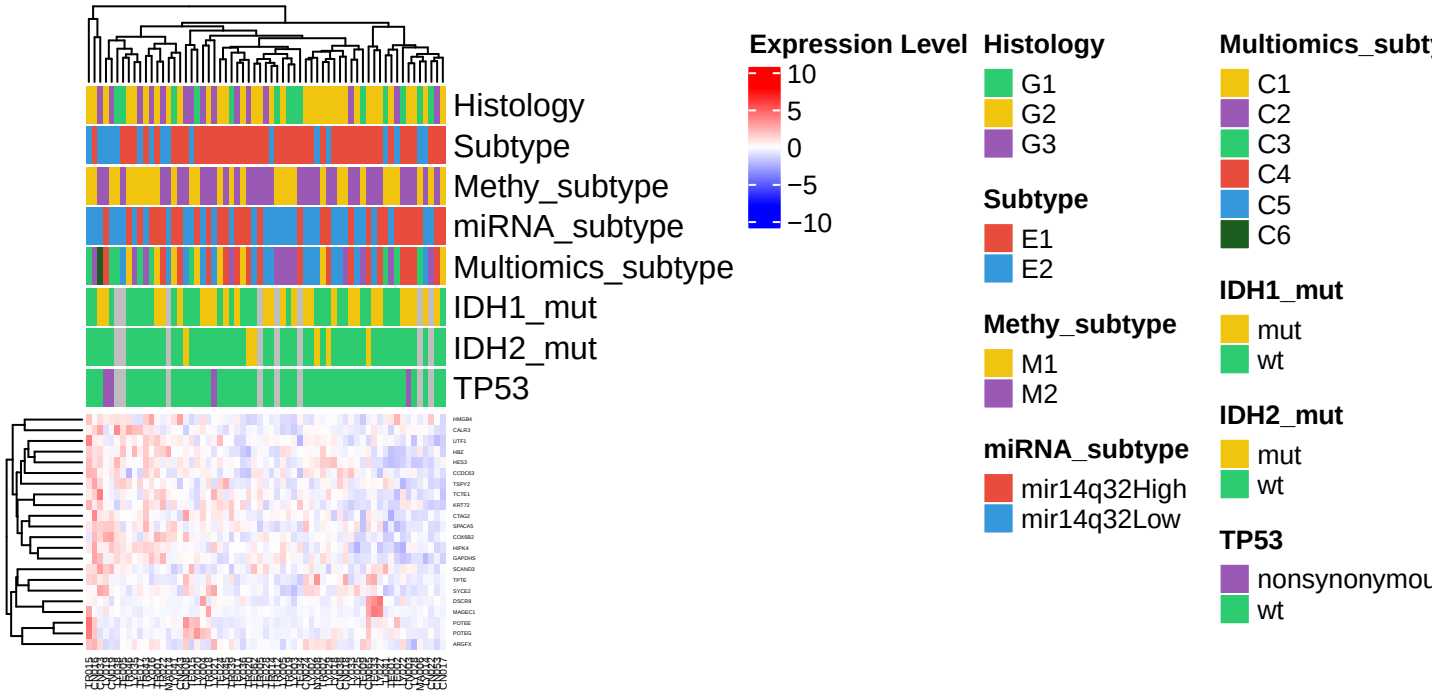


Figure 46: Not selected ignificant CTAs heatmap (from fig.8) with metadata (n = 63)

VII. Expression analysis benign vs others

In this part, we want to see the expression differences between all the histology types.

1) PCA to see groups

```
# Perform PCA to see the groups by histology
df_metadata_histo <- df_metadata[, c("Patient", "Histology")]
df_indiv <- as.data.frame(t(df_CTA_immune_whole_clean_avg_102[,
```

```
    -c(1:3)]))
df_indiv_zscores <- as.data.frame(t(df_CTA_immune_whole_clean_z_scores_102[,
    -c(1:3)]))
histo <- factor(df_metadata_histo$Histology)
pca <- prcomp(df_indiv, scale. = TRUE)
pca_plot(pca, histo, TRUE)
```

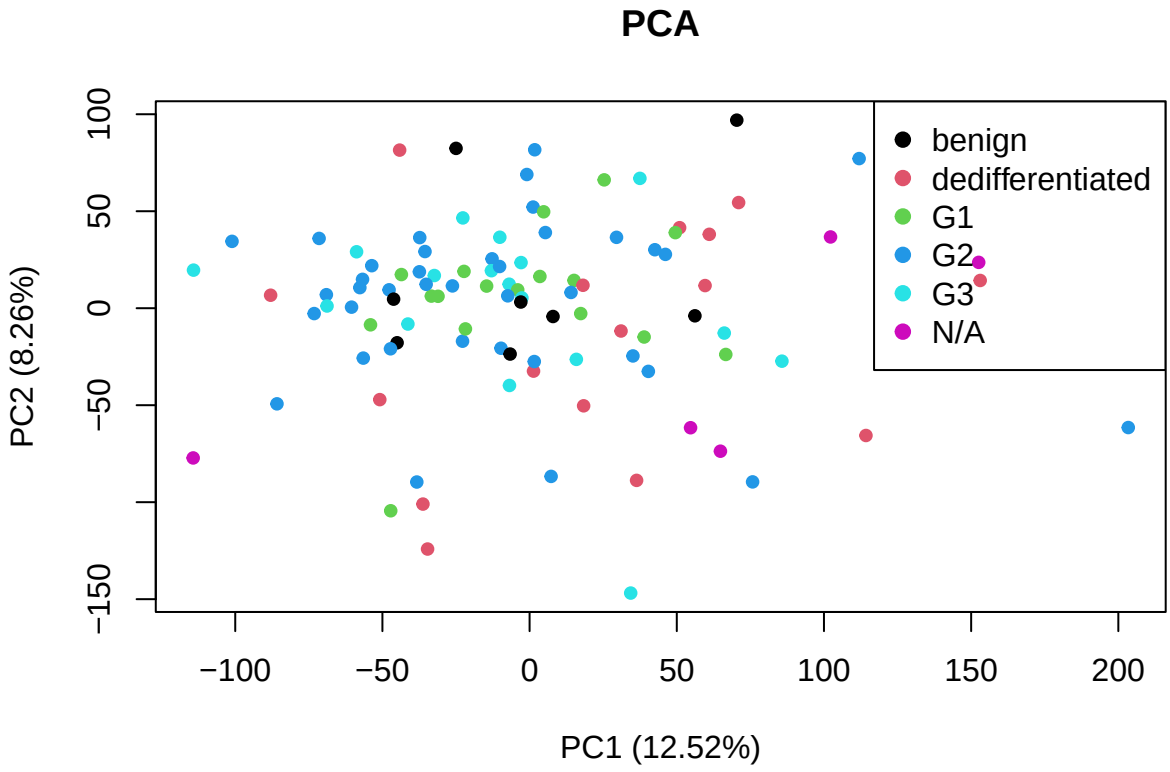


Figure 47: PCA plot with histology type (n = 102)

We see that there is not really groups, but I try DEG analysis

2) DEG analysis between benign tumors and malignant tumors

```
# Create factors
histo_benign <- ifelse(histo == "benign", "benign", "malignant")
f <- factor(histo_benign)
design <- model.matrix(~0 + f)  # 0 to compare all pairwises
colnames(design) <- c("benign", "malignant")
colnames(design) <- make.names(colnames(design))

# Fit the linear model
data_fit <- lmFit(df_CTA_immune_whole_clean_avg_102[, -c(1:2)],
    design)

# Define contrasts
contrast_matrix <- makeContrasts(benign - malignant, levels = design)
```

```r
data_fit_contrast <- contrasts.fit(data_fit, contrast_matrix)

# Calculate the empirical Bayes statistics
data_fit_eb <- eBayes(data_fit_contrast)

# Extract the top genes
res <- topTable(data_fit_eb, adjust = "BH", sort.by = "P", number = Inf)
res_cta <- res[deg_cta, ]

# Volcano plot
EnhancedVolcano(res, lab = rownames(res), pCutoff = 0.01, FCcutoff = 0.8,
    x = "logFC", y = "adj.P.Val", pointSize = 1.5, legendLabSize = 10,
    labSize = 3, title = "Volcano plot with all genes", subtitle = "benign vs malignant")
```

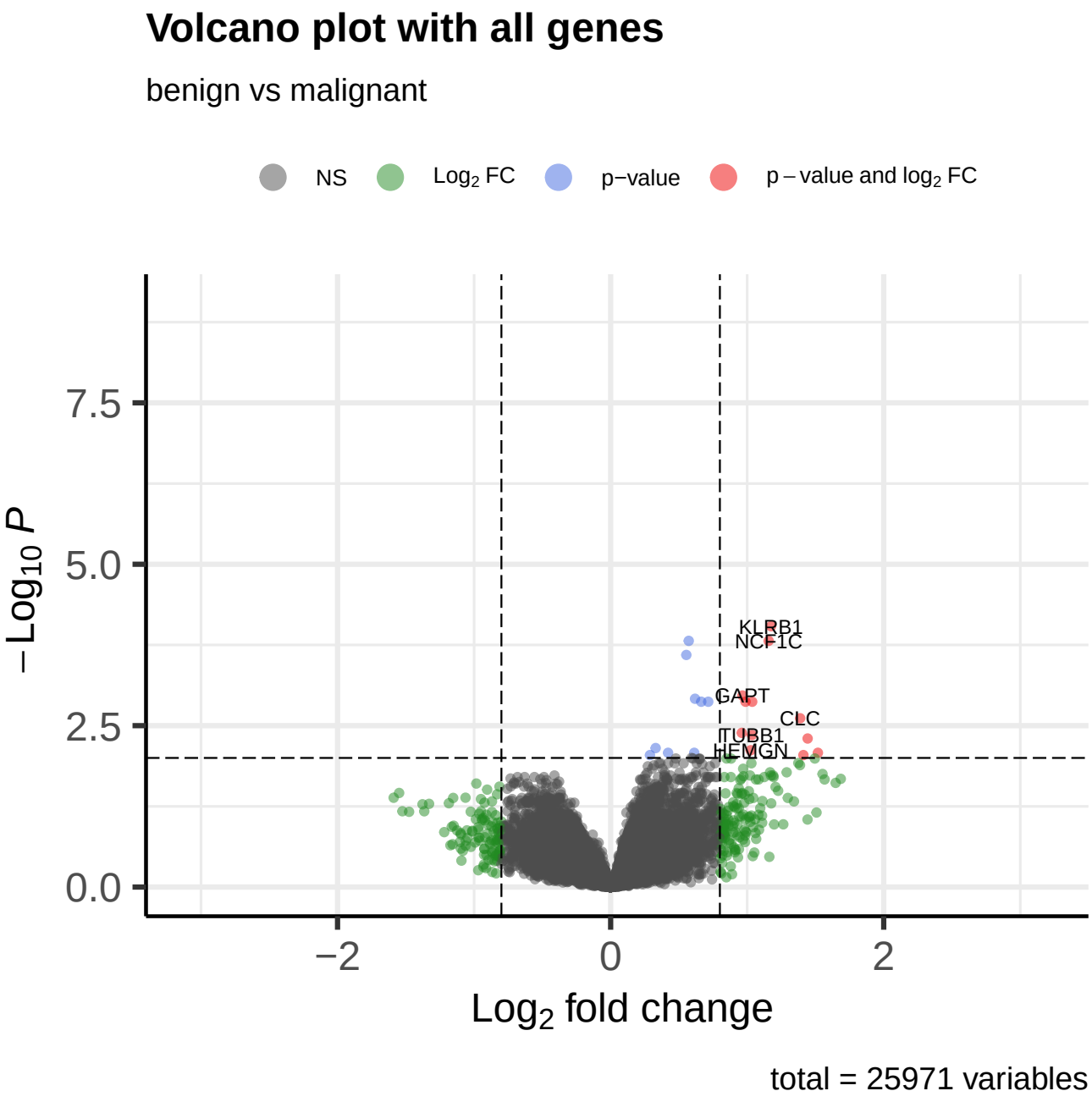


Figure 48: Volcano plot of DEG between benign and malignant tumors

```r
# Volcano plot CTA
EnhancedVolcano(res_cta, lab = rownames(res_cta), pCutoff = 0.05,
    FCcutoff = 0.8, x = "logFC", y = "adj.P.Val", pointSize = 1.5,
    legendLabSize = 10, labSize = 3, title = "Volcano plot with CTA genes",
    subtitle = "benign vs malignant")
```

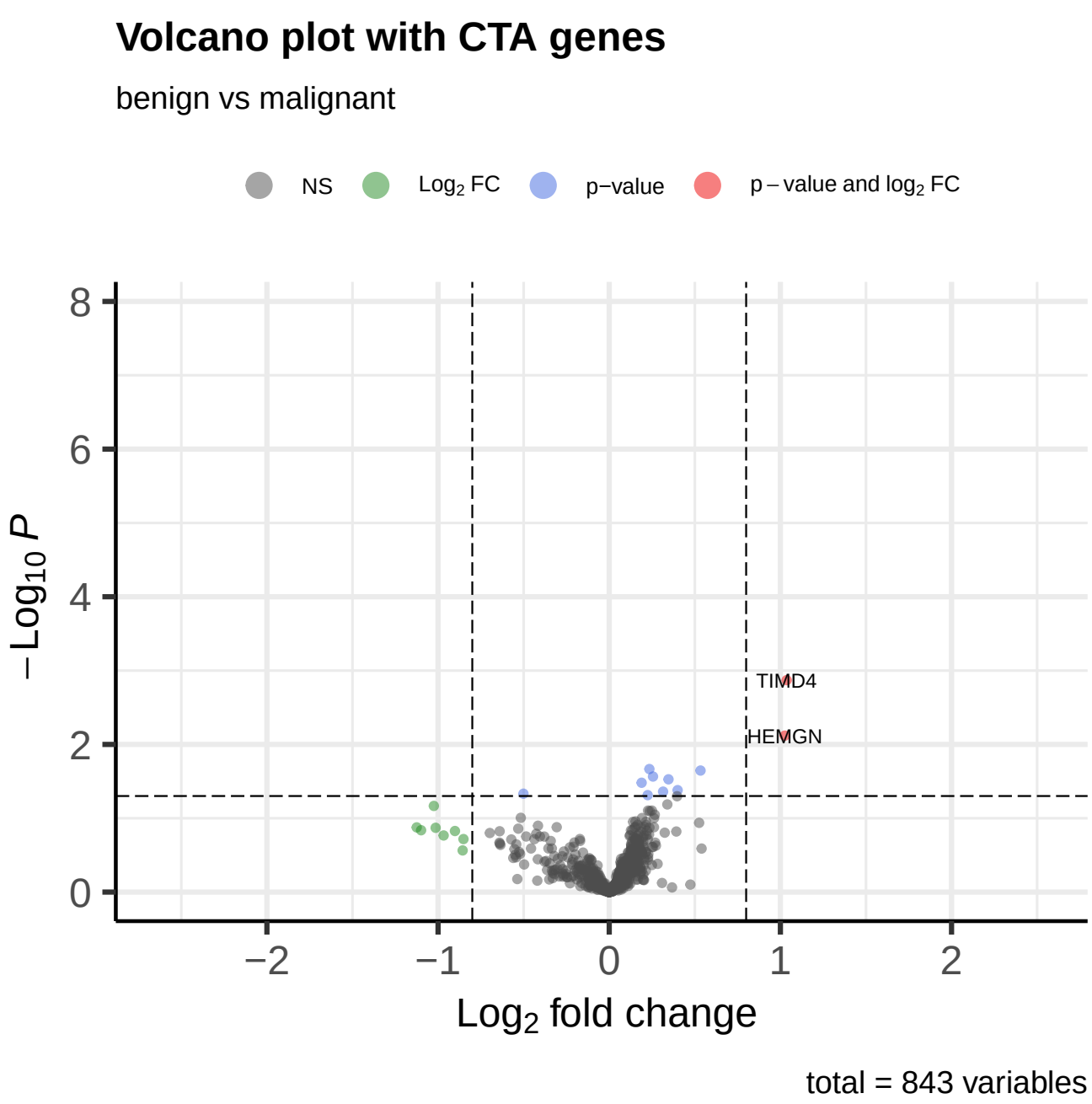


Figure 49: Volcano plot of differentially expressed CTA between benign and malignant tumors

The DEG are over expressed in benign so we can't conclude something clearly.

3) DEG analysis between all the groups

In this section, we compare benign with all the other groups.

```
f <- factor(histo)
design <- model.matrix(~0 + f)  # 0 to compare all pairwises
colnames(design) <- c("benign", "dedifferentiated", "G1", "G2",
    "G3", "na")
colnames(design) <- make.names(colnames(design))

# Fit the linear model
data_fit <- lmFit(df_CTA_immune_whole_clean_avg_102[, -c(1:2)],
    design)

# Define contrasts (HOT vs. COLD)
contrast_matrix <- makeContrasts(benign_vs_dediff = benign -
    dedifferentiated, benign_vs_G1 = benign - G1, benign_vs_G2 = benign -
    G2, benign_vs_G3 = benign - G3, benign_vs_na = benign - na,
    levels = design)

data_fit_contrast <- contrasts.fit(data_fit, contrast_matrix)

# Calculate the empirical Bayes statistics
data_fit_eb <- eBayes(data_fit_contrast)

# Extract the top genes res <- topTable(data_fit_eb,
# adjust='BH', sort.by='P', number =Inf)

resultats <- list()
resultats$benign_vs_dediff <- topTable(data_fit_eb, coef = "benign_vs_dediff",
    adjust = "BH", sort.by = "P", number = Inf)
resultats$benign_vs_G1 <- topTable(data_fit_eb, coef = "benign_vs_G1",
    adjust = "BH", sort.by = "P", number = Inf)
resultats$benign_vs_G2 <- topTable(data_fit_eb, coef = "benign_vs_G2",
    adjust = "BH", sort.by = "P", number = Inf)
resultats$benign_vs_G3 <- topTable(data_fit_eb, coef = "benign_vs_G3",
    adjust = "BH", sort.by = "P", number = Inf)
resultats$benign_vs_na <- topTable(data_fit_eb, coef = "benign_vs_na",
    adjust = "BH", sort.by = "P", number = Inf)

# Volcano plot benign vs dediff
EnhancedVolcano(resultats$benign_vs_dediff, lab = rownames(resultats$benign_vs_dediff),
    pCutoff = 0.01, FCcutoff = 0.8, x = "logFC", y = "adj.P.Val",
    pointSize = 1.5, legendLabSize = 10, labSize = 3, title = "Volcano plot with all genes",
    subtitle = "benign vs dediff")
```

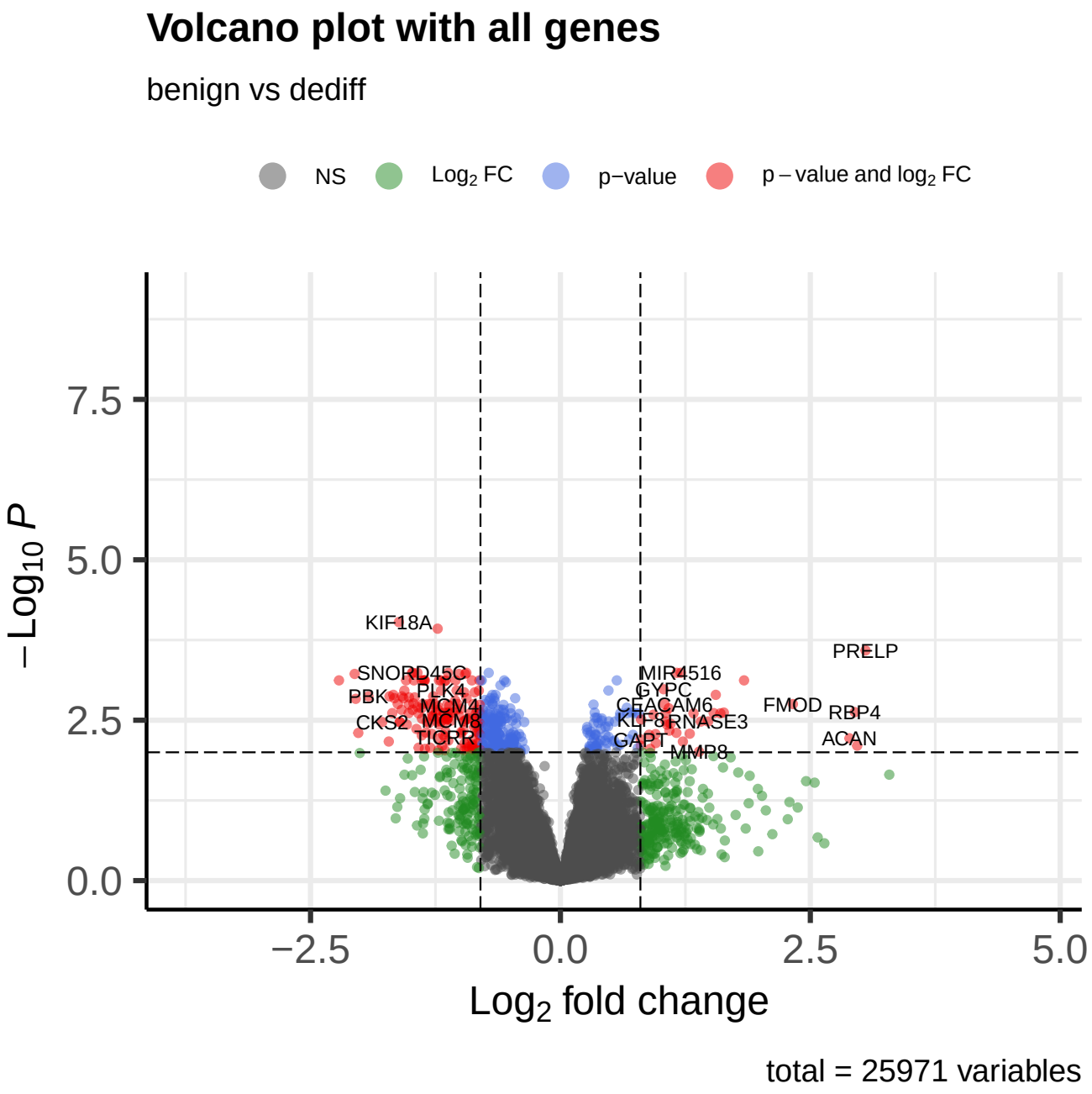


Figure 50: Volcano plot of DEG between benign and malignant tumors

```r
# Volcano plot benign vs G1
EnhancedVolcano(resultats$benign_vs_G1, lab = rownames(resultats$benign_vs_G1),
    pCutoff = 0.05, FCcutoff = 0.8, x = "logFC", y = "adj.P.Val",
    pointSize = 1.5, legendLabSize = 10, labSize = 3, title = "Volcano plot with all genes",
    subtitle = "benign vs G1")
```

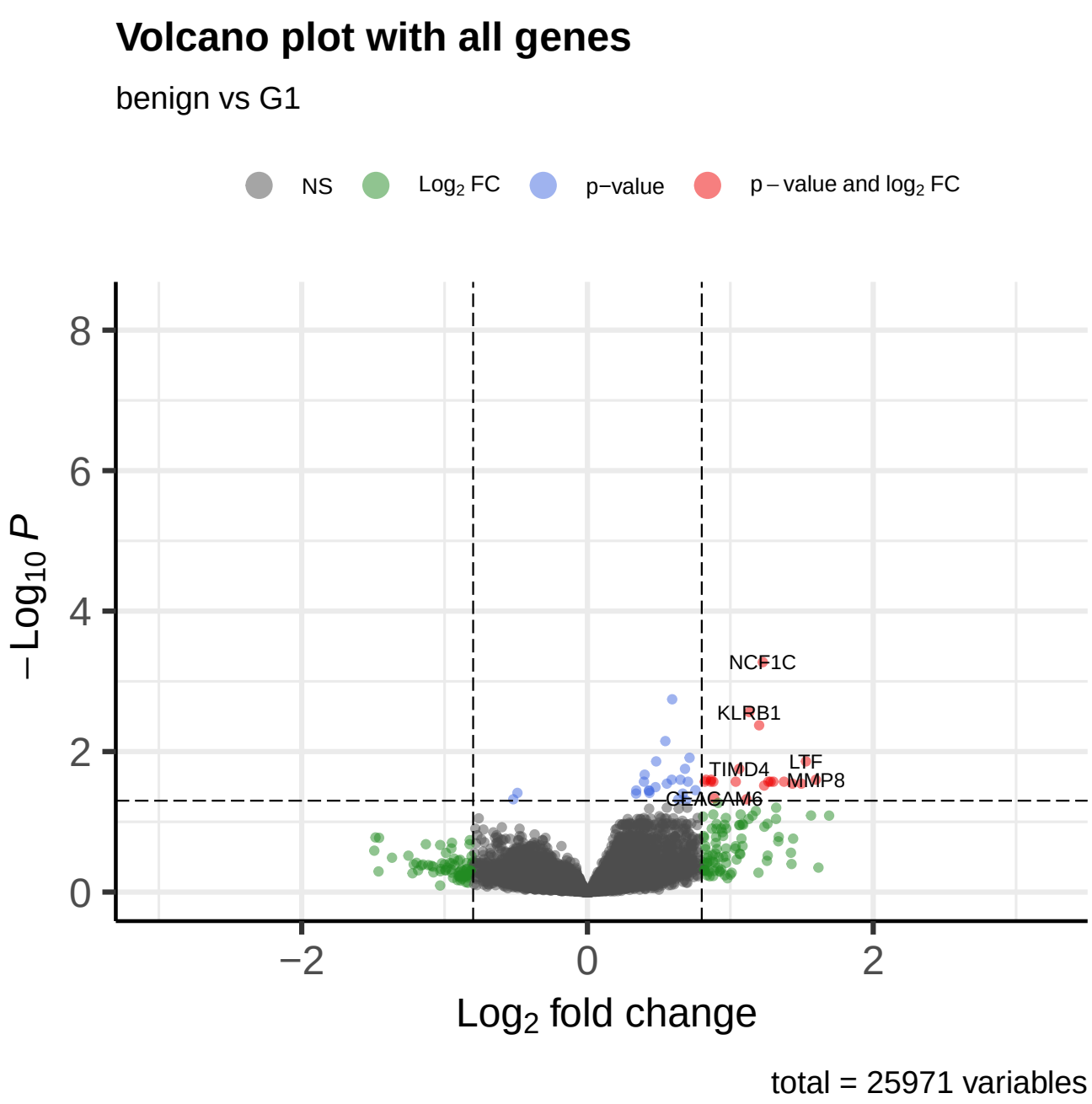


Figure 51: Volcano plot of DEG between benign and G1

```r
# Volcano plot benign vs G2
EnhancedVolcano(resultats$benign_vs_G2, lab = rownames(resultats$benign_vs_G2),
    pCutoff = 0.05, FCcutoff = 0.8, x = "logFC", y = "adj.P.Val",
    pointSize = 1.5, legendLabSize = 10, labSize = 3, title = "Volcano plot with all genes",
    subtitle = "benign vs G2")
```

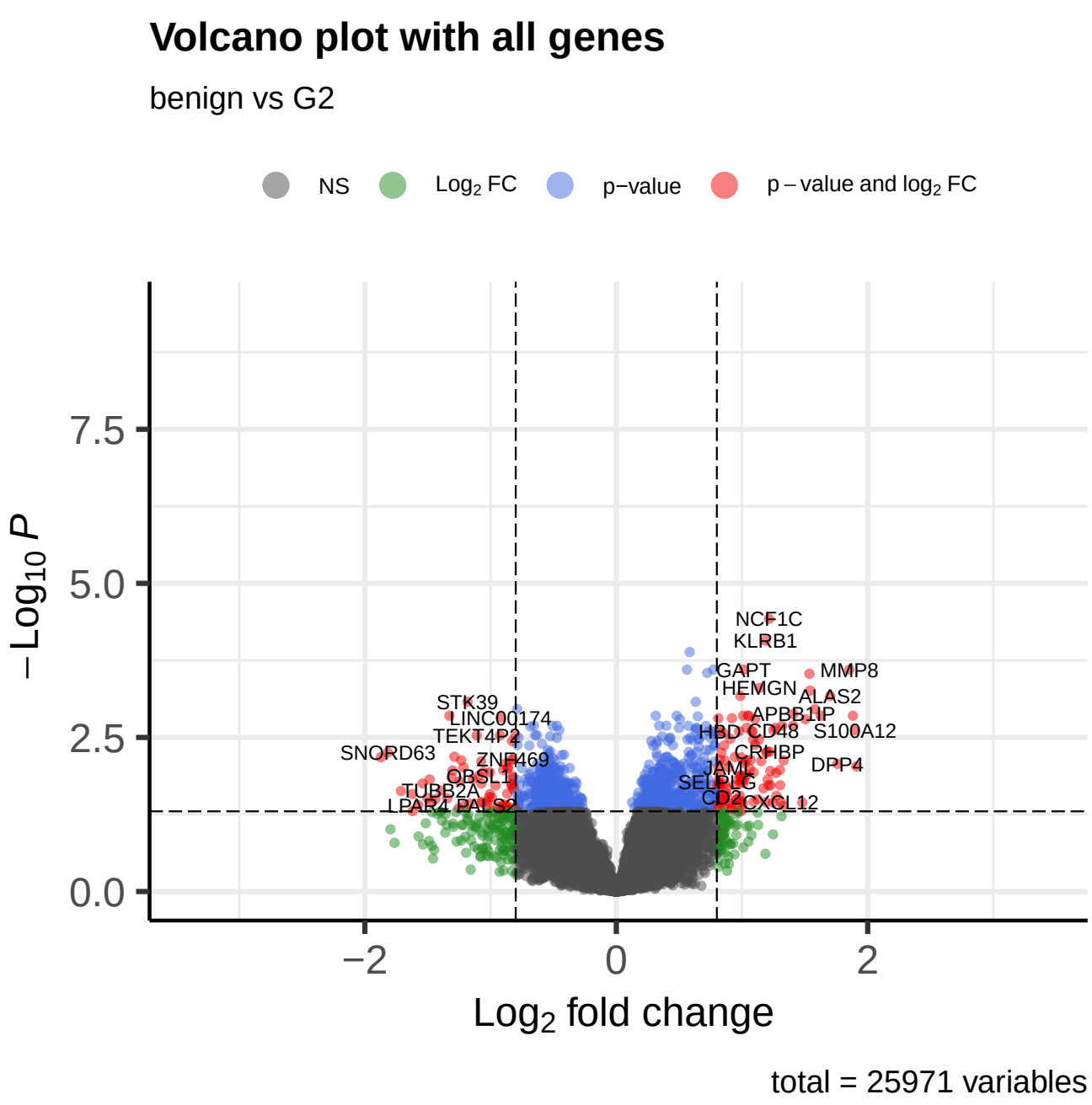


Figure 52: Volcano plot of DEG between benign and G2

```r
# Volcano plot benign vs G3
EnhancedVolcano(resultats$benign_vs_G3, lab = rownames(resultats$benign_vs_G3),
    pCutoff = 0.05, FCcutoff = 0.8, x = "logFC", y = "adj.P.Val",
    pointSize = 1.5, legendLabSize = 10, labSize = 3, title = "Volcano plot with all genes",
    subtitle = "benign vs G3")
```

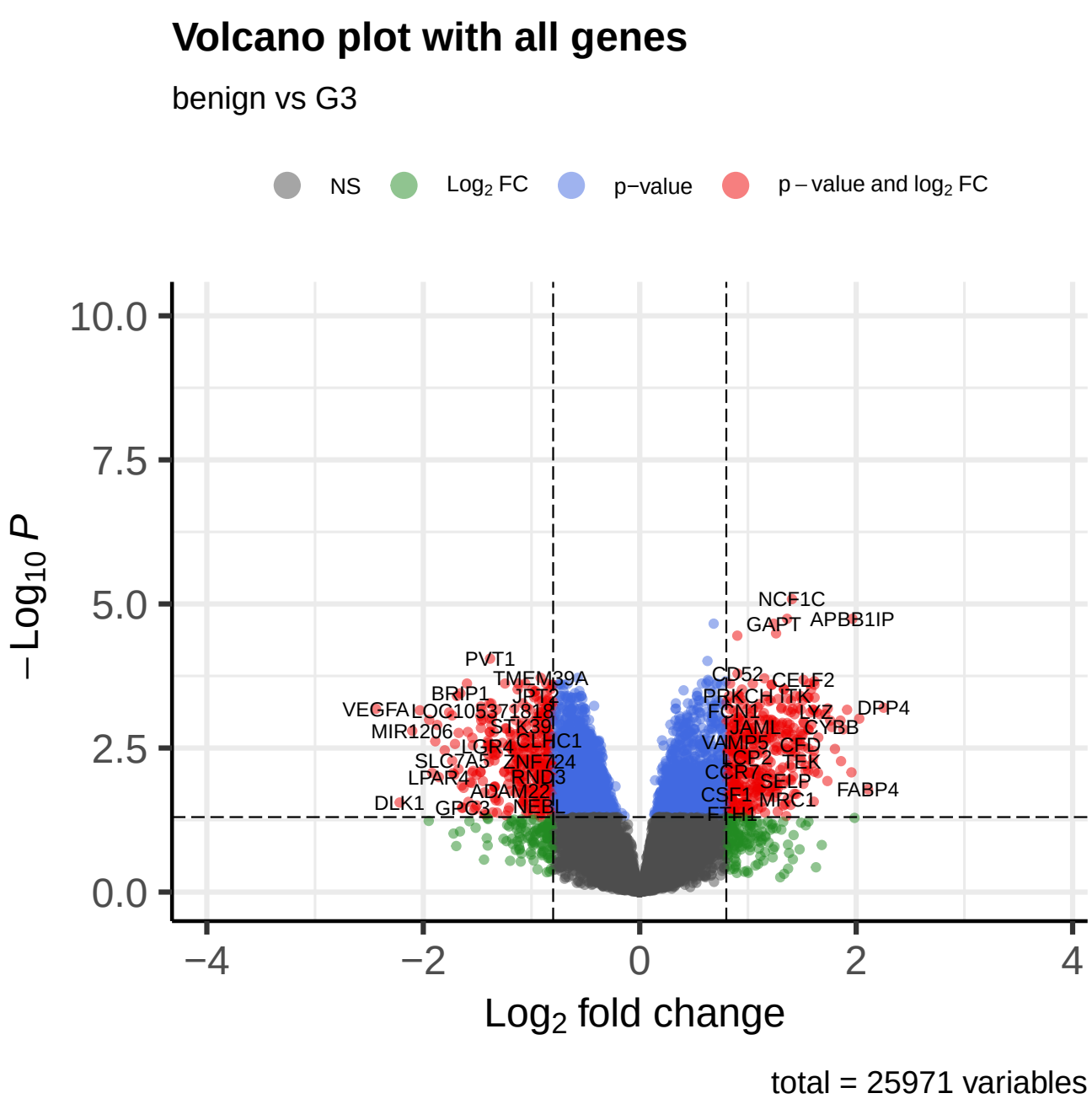


Figure 53: Volcano plot of DEG between benign and G3

VIII. 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

```
# Create the heatmap
heatmap_cta_signif_conv_sorted <- Heatmap(t(data_selected_CTA_conv_63),
    cluster_rows = TRUE,
    cluster_columns = TRUE,
    cluster_column_slices = TRUE,
    clustering_distance_columns = "euclidean",
    clustering_method_columns = "complete",
    col = colorRamp2(seq(-8, 8, length.out = 100), colors),
    border = NA,
    show_column_names = TRUE,
    show_row_names = TRUE,
    column_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 2),
    heatmap_legend_param = list(title = "Expression Level")
)

# Kmeans heatmap
ht_k2_conv <- Heatmap(
    as.matrix(t(df_imm_z_scores_63[, colnames(df_imm_z_scores_63)])),
    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")
)
heatmap_cta_signif_conv_sorted + ht_k2_conv
```

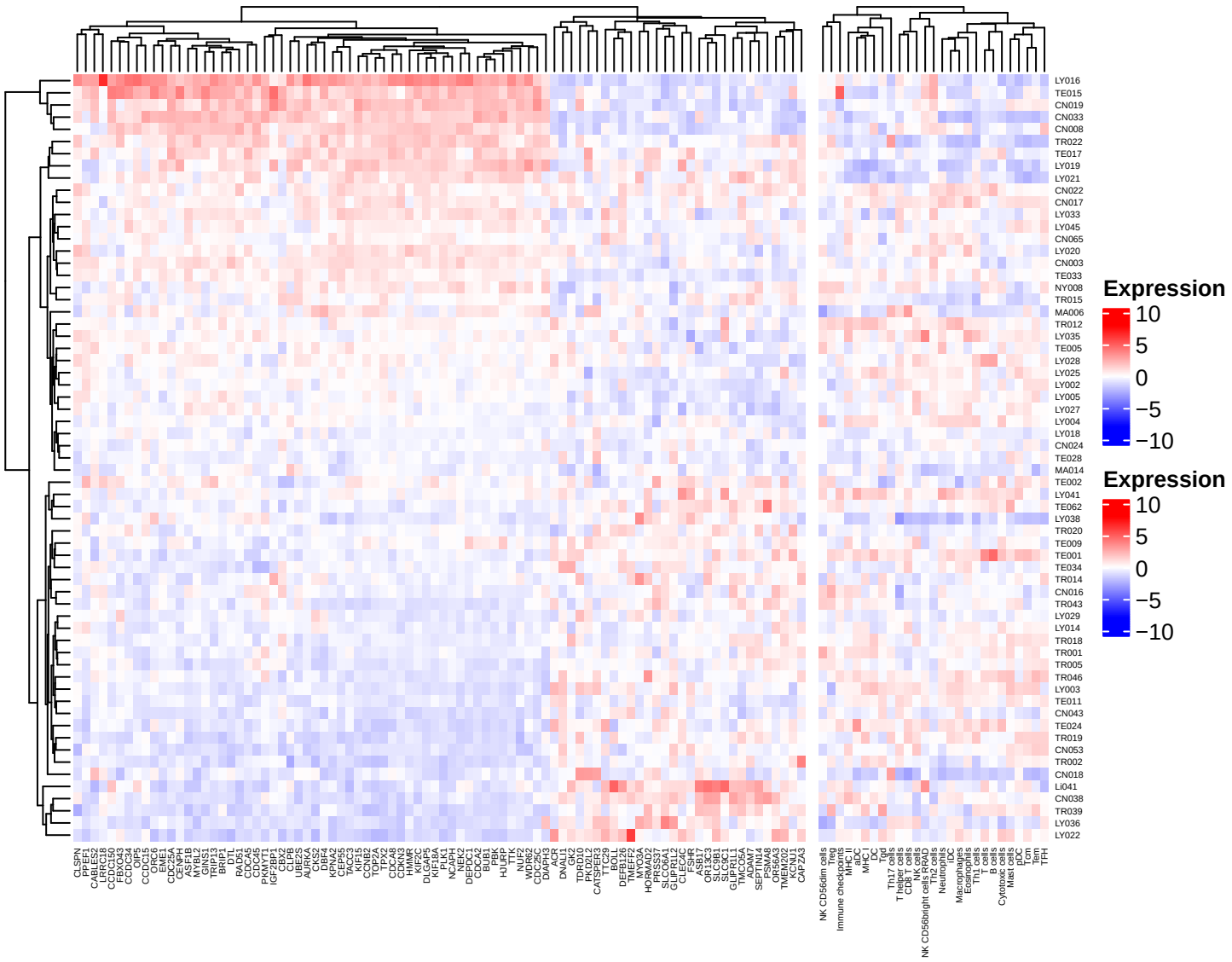


Figure 54: Heatmap of CTA that impact survival analysis and immune cells expression (n = 63)

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.

b- Kmeans clustering

```
set.seed(1)
kmeans_result <- kmeans(t(data_cta_63), centers = 2, nstart = 25)  # 2 clusters

# CTA heatmap
ht_cta <- Heatmap(
  t(data_cta_63),
  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_names_gp = gpar(fontsize = 4),
  row_names_gp = gpar(fontsize = 4),
  heatmap_legend_param = list(title = "Expression Level"),
  row_split = kmeans_result$cluster
)

# kmeans heatmap
ht_k2_conv <- Heatmap(
    as.matrix(t(df_imm_z_scores_63)),
    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))
ht_cta + ht_k2_conv
```

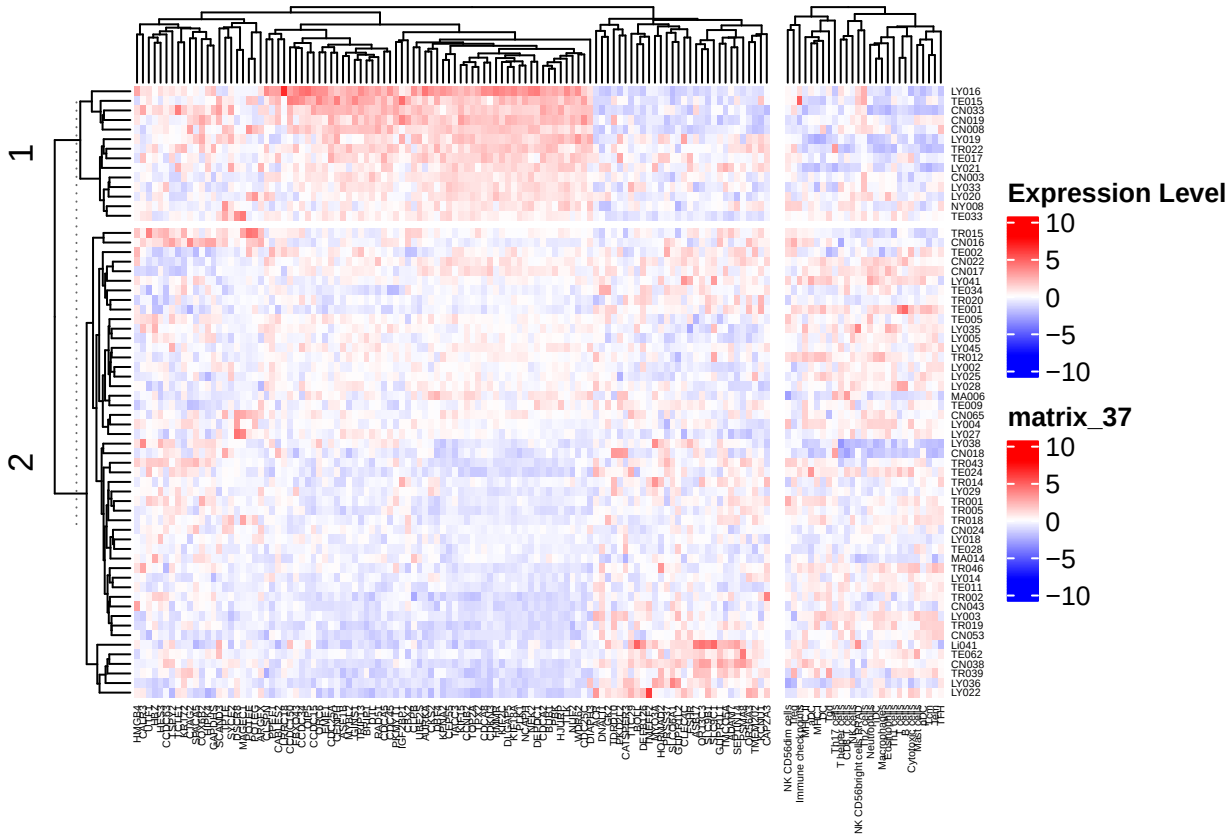


Figure 55: Heatmap with kmeans clustering of CTA that impact survival analysis and immune cells expression (n = 63)

With kmeans k = 2 on the patients like previous heatmap (IV.1), we confirm the previous conclusion that there are a link for some patients between expression of CTA and immune cell infiltration. Now, we will try to quantify this co-expression.

2) Weighted correlation network analysis (WGCNA)

The R package WGCNA allows to see this co-expression between genes. So here, we want to observe the link between CTA genes that have an impact on the survival for conventional chondrosarcoma and the immune cell infiltration.

```
# Prepare data and select the CTA
df <- as.data.frame(t(df_z_scores_63[rownames(df_z_scores_63) %in%
    l_CTA_conv, -c(1, 2)]))

# Immune cells data
rownames(df_indiv_clusters_hm_anno_63) <- df_indiv_clusters_hm_anno_63$Patient
df_imm_data <- as.data.frame(t(df_imm_z_scores_63))

# Re-cluster samples
sample_hclust <- hclust(dist(df), method = "ward.D")

# Plot the sample dendrogram
plot(sample_hclust, xlab = "Samples", ylab = "Height", cex = 0.7,
    sub = "", main = "Sample dendrogram")
```

This plot show the hierarchical clustering on the distance betwenn samples. Here, we see that LY016 is outlier, so I delete it.

```
# Delete Ly016
df_imm_data <- df_imm_data[!rownames(df_imm_data) %in% "LY016",
    ]
df <- df[!rownames(df) %in% "LY016", ]

# Re-cluster samples
sample_hclust <- hclust(dist(df), method = "complete")

# Plot the sample dendrogram
plot(sample_hclust, xlab = "Samples", ylab = "Height", cex = 0.7,
    sub = "", main = "Samples dendrogram")
```

a- Threshold power selection for WGCNA

In this section, we perform the **network topology** analysis to select the **optimal threshold power** for WGCNA. We use the function `pickSoftThreshold()` to evaluate different topology indices and then view the results.

```
# Choose a set of soft-thresholding powers
powers <- c(c(1:12), seq(from = 12, to = 20, by = 2))

# Call the network topology analysis function
soft_threshold <- pickSoftThreshold(df, powerVector = powers,
    verbose = 0)
```

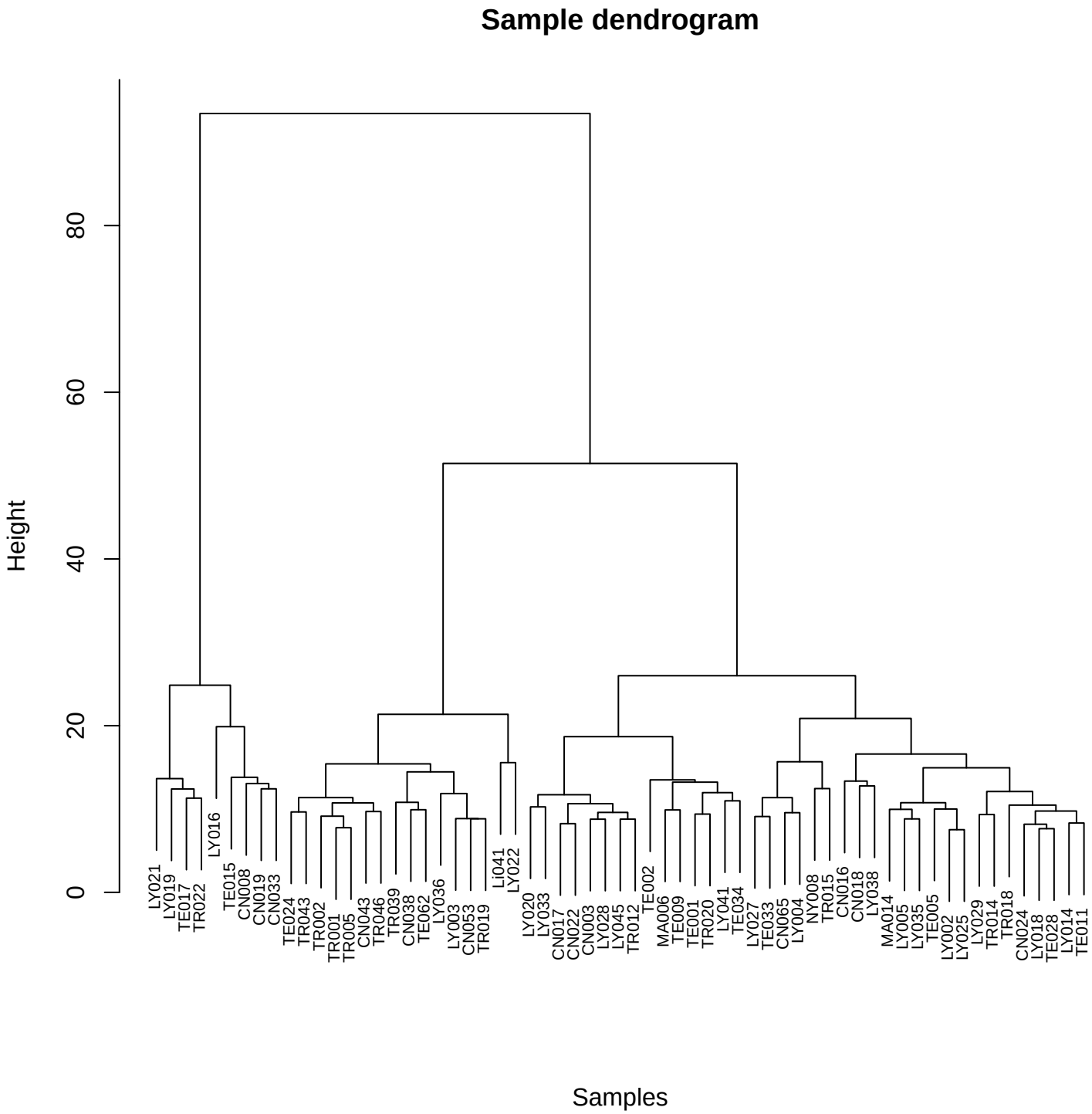


Figure 56: Dendrogram of conventionnal patients (from fig. 6)

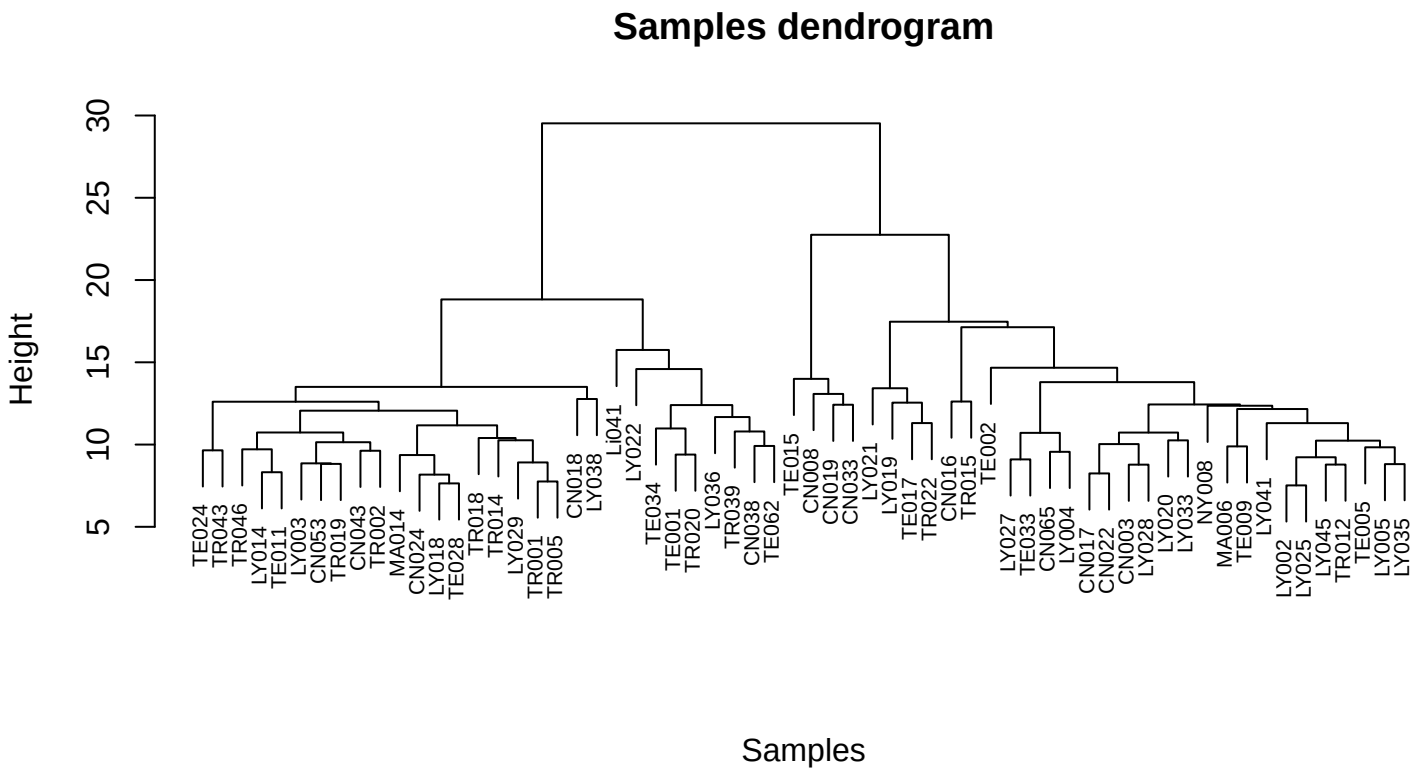


Figure 57: Corrected dendrogram of conventionnal patients (from fig. 6)

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

```
##    Power SFT.R.sq   slope truncated.R.sq mean.k. median.k. max.k.
## 1      1 0.000761  0.0257         0.0860  30.500  3.13e+01  49.60
## 2      2 0.218000 -0.4410         0.5100  14.600  1.16e+01  33.30
## 3      3 0.351000 -0.5070         0.6550   8.720  4.84e+00  25.20
## 4      4 0.450000 -0.5570         0.5560   5.810  2.15e+00  20.00
## 5      5 0.734000 -0.5990         0.7800   4.110  9.93e-01  16.30
## 6      6 0.773000 -0.6390         0.7450   3.030  4.73e-01  13.50
## 7      7 0.819000 -0.6670         0.7880   2.290  2.87e-01  11.30
## 8      8 0.105000 -1.3800        -0.1490   1.770  2.35e-01   9.51
## 9      9 0.917000 -0.7310         0.8940   1.380  1.31e-01   8.08
## 10    10 0.972000 -0.7330         0.9680   1.100  7.38e-02   6.91
## 11    11 0.810000 -0.7880         0.7560   0.885  4.18e-02   5.93
## 12    12 0.103000 -1.2800        -0.1440   0.720  2.38e-02   5.12
## 13    12 0.103000 -1.2800        -0.1440   0.720  2.38e-02   5.12
## 14    14 0.920000 -0.8750         0.8990   0.488  7.84e-03   3.85
## 15    16 0.191000 -1.7900         0.0637   0.341  2.70e-03   2.94
## 16    18 0.193000 -1.8600         0.0611   0.244  1.22e-03   2.26
## 17    20 0.195000 -1.7900         0.0567   0.178  5.07e-04   1.76
```

```
# Plot the results
par(mfrow = c(1, 2))

# Scale-free topology fit index as a function of the
# soft-thresholding power
```

```r
plot(soft_threshold$fitIndices[, 1], -sign(soft_threshold$fitIndices[,
    3]) * soft_threshold$fitIndices[, 2], xlab = "Soft Threshold (power)",
    ylab = "Scale Free Topology Model Fit,signed R²", type = "n",
    main = paste("Scale independence"))
text(soft_threshold$fitIndices[, 1], -sign(soft_threshold$fitIndices[,
    3]) * soft_threshold$fitIndices[, 2], labels = powers, cex = 0.9,
    col = "red")

# this line corresponds to using an R² cut-off of h
abline(h = 0.9, col = "red")

# Mean connectivity as a function of the soft-thresholding
# power
plot(soft_threshold$fitIndices[, 1], soft_threshold$fitIndices[,
    5], xlab = "Soft Threshold (power)", ylab = "Mean Connectivity",
    type = "n", main = paste("Mean connectivity"))
text(soft_threshold$fitIndices[, 1], soft_threshold$fitIndices[,
    5], labels = powers, cex = 0.9, col = "red")
```

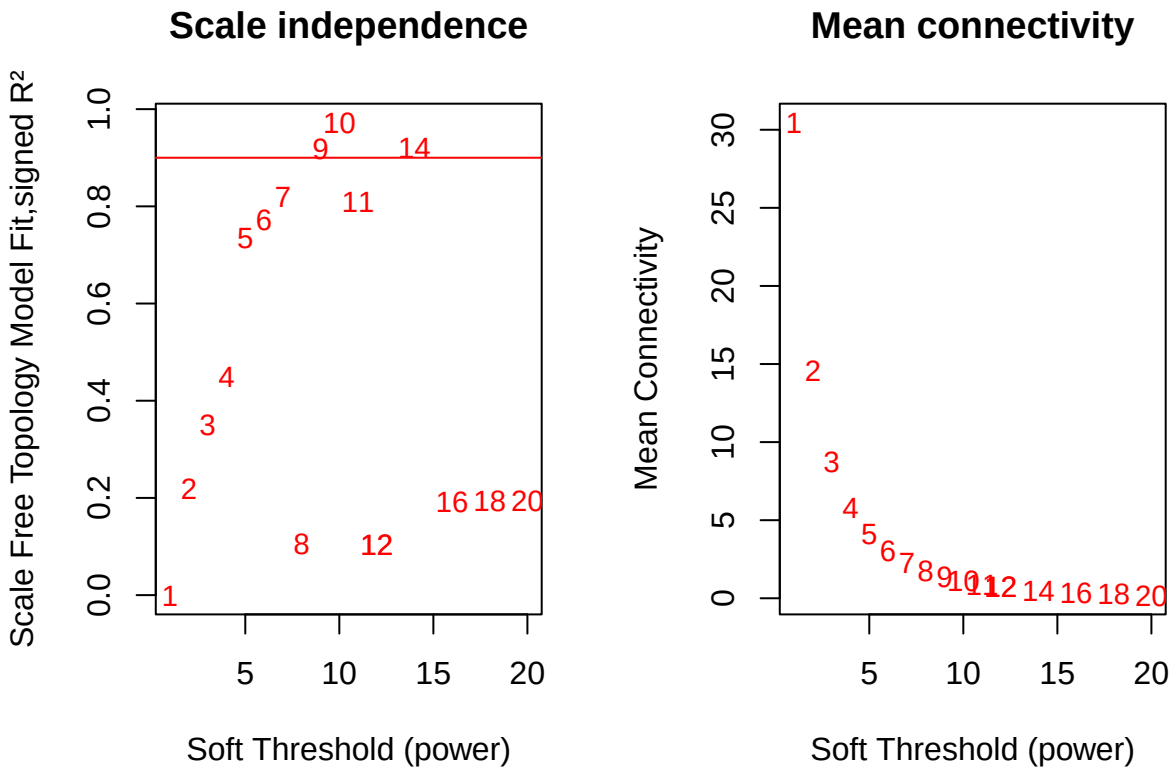


Figure 58: Scale independence and mean connectivity plots

The scale independence show that the power 9 is > than 0.9 and the mean connectivity show the smallest connectivity. So this is why we choose 9.

b- Gene clustering and module detection with the topological similarity matrix (TOM)

In this section, we apply an approach to detect gene modules using the topological similarity matrix (TOM). This method minimizes the effects of noise to identify groups of co-expressed genes. Then we use the method

cutreeDynamic() to cut the tree into modules, and then we visualize the dendrogram with the identified modules.

```
# Turn adjacency into topological overlap matrix (TOM), to
# minimize effects of noise and spurious associations, we
# transform the adjacency into Topological Overlap Matrix,
# and calculate the corresponding dissimilarity
adjacency <- adjacency(df, power = 9)
TOM_adj <- TOMsimilarity(adjacency)
```

```
## ..connectivity..
## ..matrix multiplication (system BLAS)..
## ..normalization..
## ..done.
```

```
dissimilarity_TOM_adj <- 1 - TOM_adj

# Clustering using TOM
dendro <- hclust(as.dist(dissimilarity_TOM_adj), method = "average")

# Plot the resulting clustering tree (dendogram)
plot(dendro, xlab = "", sub = "", main = "Gene Clustering on TOM-based disssimilarity",
    labels = FALSE, hang = 0.04)
```

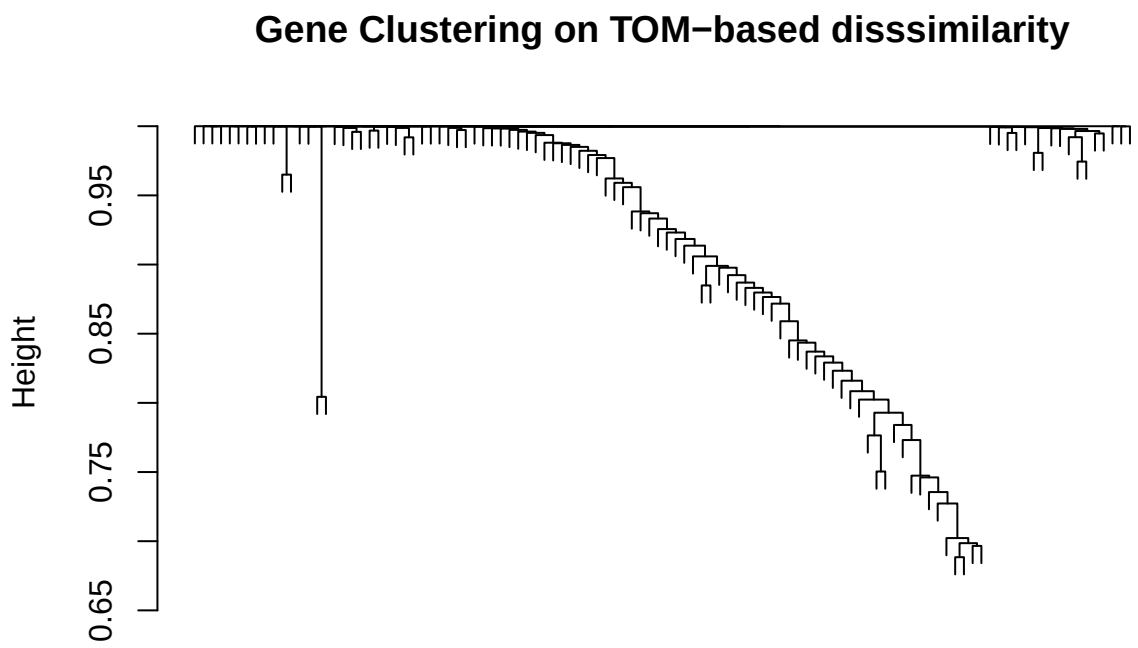


Figure 59: Dendogram of gene clustering based on topological overlap matrix dissimilarity (TOM)

We can suppose that the middle cluster correspond to the middle cluster on the heatmap.

```
# Detect the modules
min_module_size <- 30
dynamic_modules <- cutreeDynamic(dendro = dendro, distM = dissimilarity_TOM_adj,
    deepSplit = 2, pamRespectsDendro = FALSE, minClusterSize = min_module_size,
    verbose = 0)

# Convert numeric lables into colors
dynamic_colors <- labels2colors(dynamic_modules)

# Plot the dendrogram and colors underneath
plotDendroAndColors(dendro, dynamic_colors, "Dynamic Tree Cut",
    dendroLabels = FALSE, hang = 0.03, addGuide = TRUE, guideHang = 0.05,
    main = "Gene dendrogram and module colors")
```

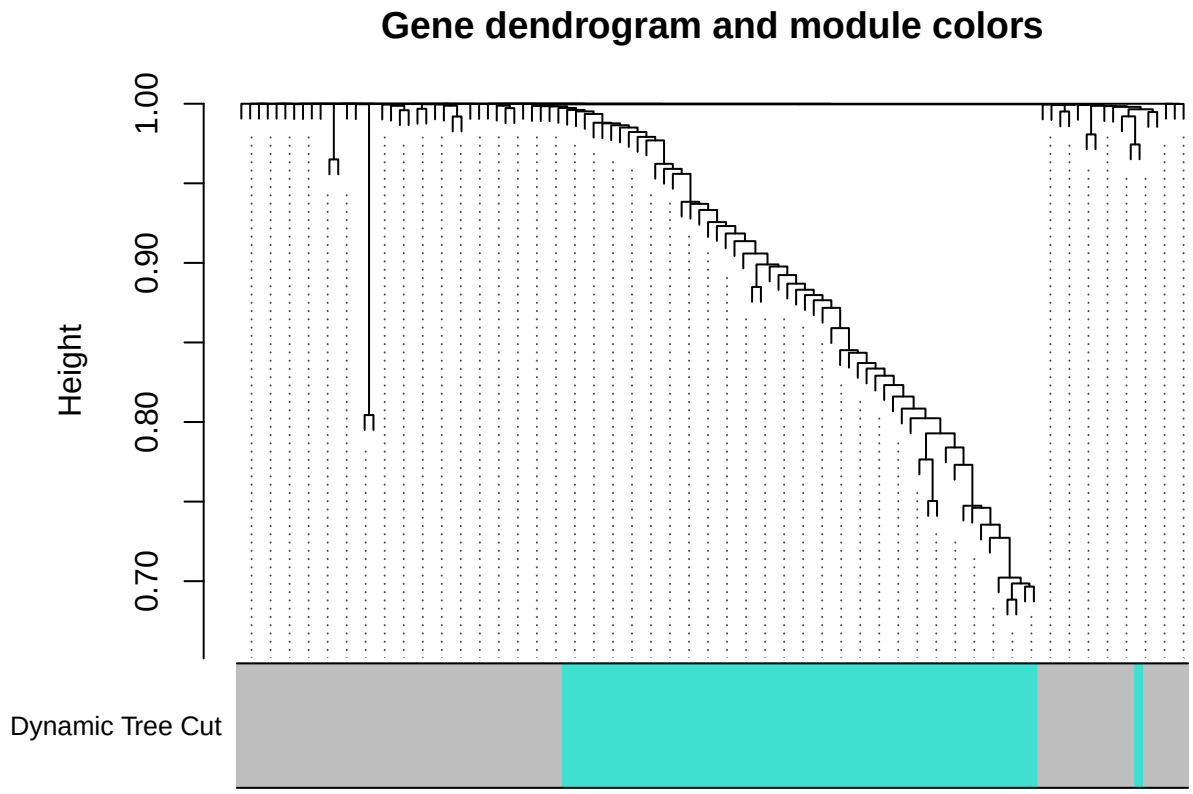


Figure 60: Dendogram with module detection

We see 2 groups, the blue seems to correspond to the middle cluster.

c- Relationship between gene modules and immune cell types: Correlation heatmap

In this section, we calculate the correlation between module genes (calculated from the values of modules' own vectors, i.e., module eigengenes) and traits of interest (here, immune cell expression data). Then we display these correlations as a heatmap.

```
n_samples <- nrow(df)

# Recalculate moduleEigengenes with color labels
module_eigengenes0 <- moduleEigengenes(df, dynamic_colors)$eigengenes
module_eigengenes <- orderMEs(module_eigengenes0)
names(module_eigengenes) <- substring(names(module_eigengenes0),
    3)

# Compute correlation
modules_cor <- cor(module_eigengenes0, df_imm_data, method = "pearson")
module_cor_pval = corPvalueStudent(modules_cor, n_samples)

# Plot
text <- paste(signif(modules_cor, 2), "\n(", signif(module_cor_pval,
    1), ")", sep = "")
dim(text) <- dim(modules_cor)

# Display the correlation values within a heatmap
par(mar = c(8, 6, 3, 3))  # Ajuster les marges (bas, gauche, haut, droite)
labeledHeatmap(Matrix = modules_cor, xLabels = names(df_imm_data),
    yLabels = names(module_eigengenes0), ySymbols = c("grey",
        "blue"), colorLabels = FALSE, colors = blueWhiteRed(50),
    textMatrix = text, setStdMargins = FALSE, cex.text = 0.5,
    zlim = c(-1, 1), main = paste("Module-trait Relationships"))
```

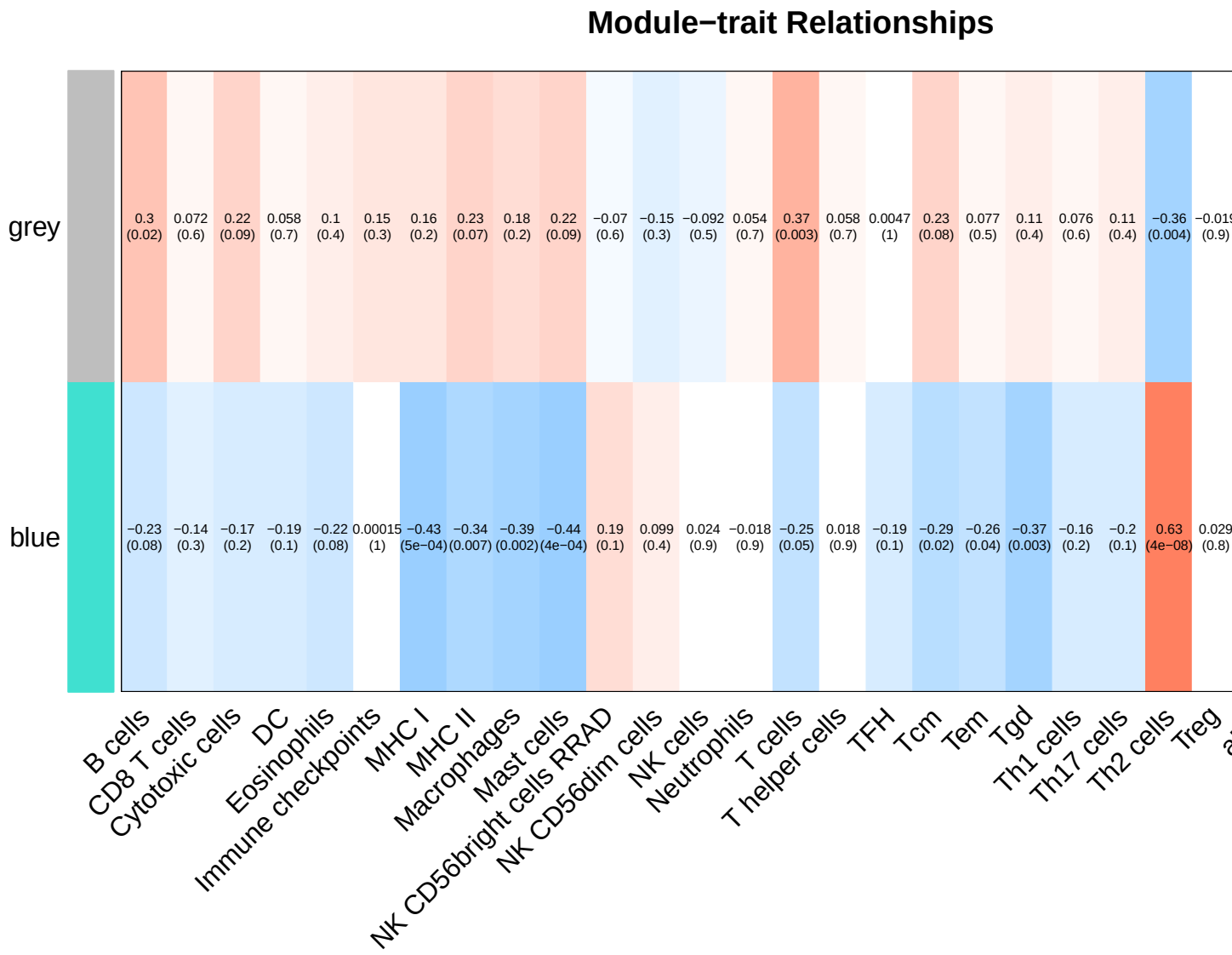


Figure 61: Pearson correlation matrix between modules and immune cells expression

This pearson correlation matrix shows that there is a positive correlation between th2 cells and blue group, and for MHC I, MHC II, macrophages, Mast cells, T cells, Tem, Tgd and DC cells there is a negative correlation.

```
# Df to have genes and their correspondent color
gene_module_df <- data.frame(gene = colnames(df), color = dynamic_colors)
rownames(gene_module_df) <- gene_module_df$gene

modules_colors <- c(grey = "grey", turquoise = "turquoise")

# Heatmap
Heatmap(data_cta_63, 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_names_gp = gpar(fontsize = 4),
    row_names_gp = gpar(fontsize = 2), right_annotation = rowAnnotation(Module = gene_module_df$color,
        col = list(Module = modules_colors)), heatmap_legend_param = list(title = "Expression Level"))
```

```
## Warning: The input is a data frame-like object, convert it to a matrix.
```

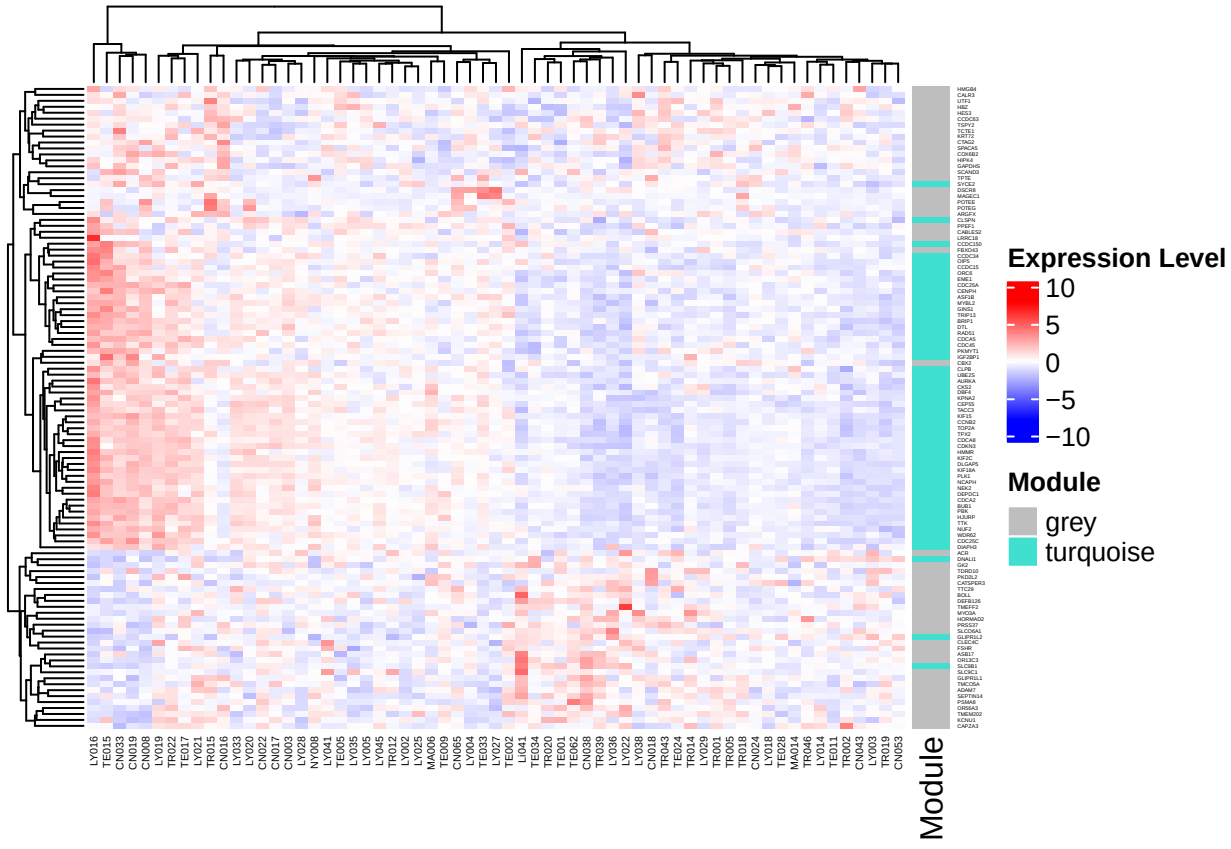


Figure 62: Heatmap of CTA colored with modules

3) Pearson correlation for conventional chondrosarcomas

a- Selected CTA

```
# Pearson correlation and p-value compute
cor_matrix <- cor(t(df_imm_z_scores_63), t(data_selected_CTA_conv_63),
    method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, 63)

# Significant p-val
pval_signif <- cor_pval < 0.05

# Heatmap
colors <- colorRampPalette(c("#75E05A", "white", "#EA4343"))(100)
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"))
        }
    })
```

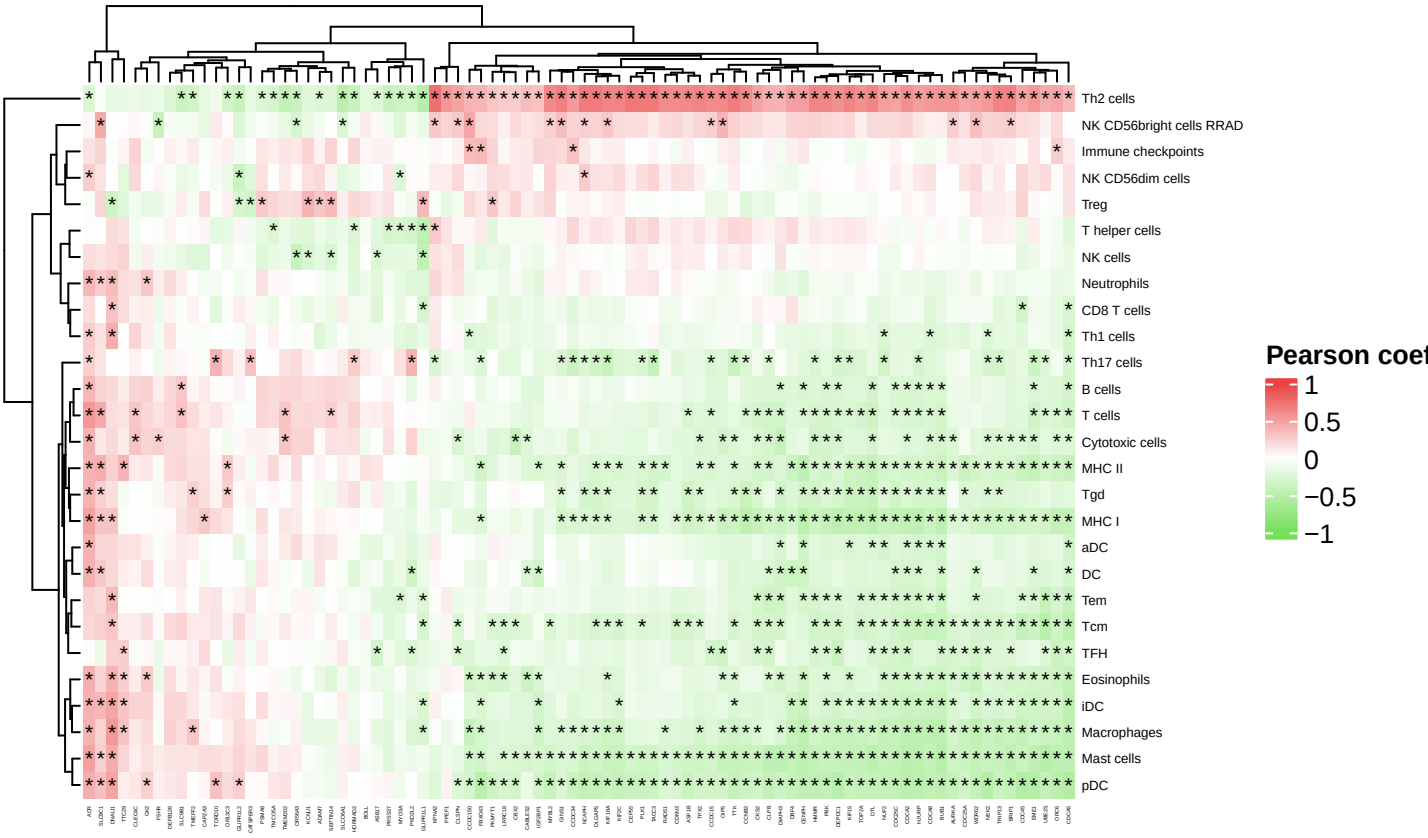


Figure 63: Pearson correlation matrix between selected CTA and immune cells expression (n = 63)

b- All CTA significant

```
data <- df_z_scores_63[rownames(df_z_scores_63) %in% l_CTA_conv,
    -c(1, 2)]

# Pearson correlation and p-value compute
cor_matrix <- cor(t(df_imm_z_scores_63), t(data), method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, 63)

# Significant p-val
pval_signif <- cor_pval < 0.05

# Heatmap
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`.
```

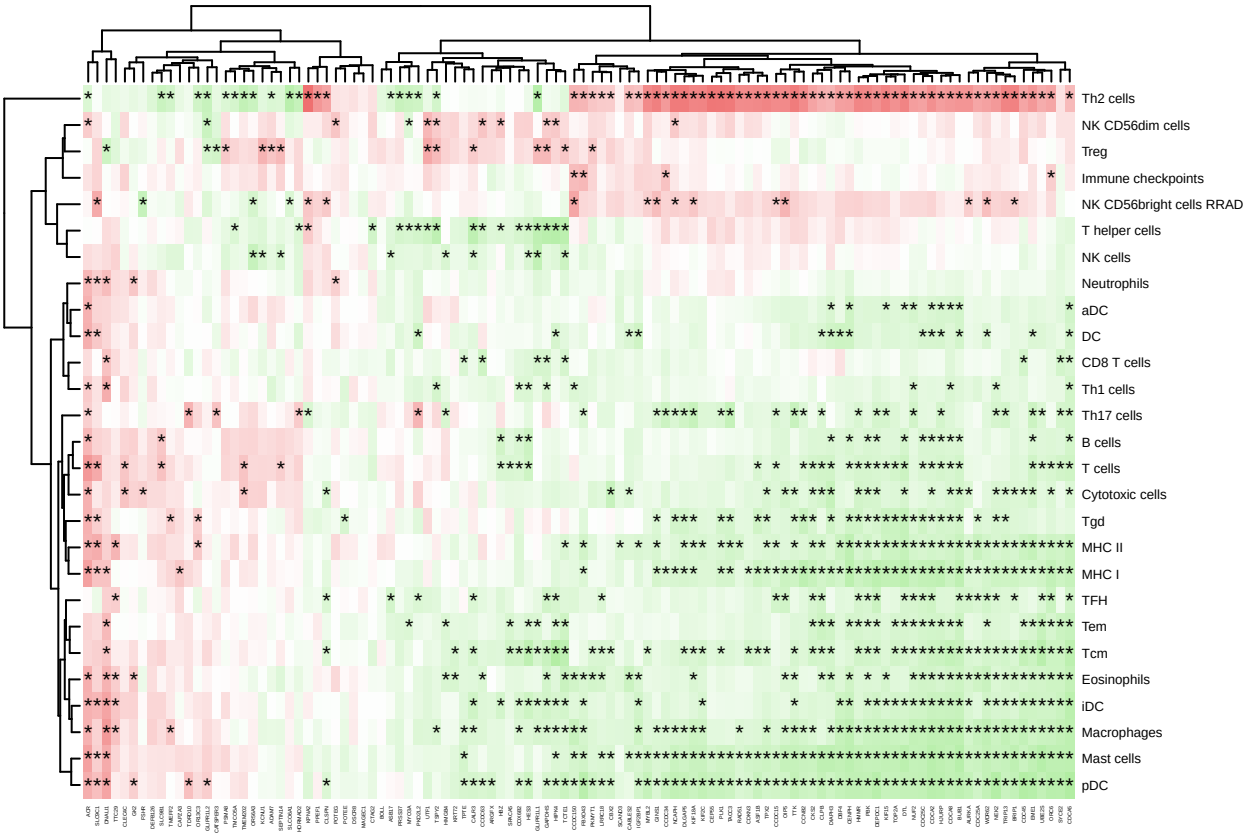


Figure 64: Pearson correlation matrix between significant CTA and immune cells expression (n = 63)

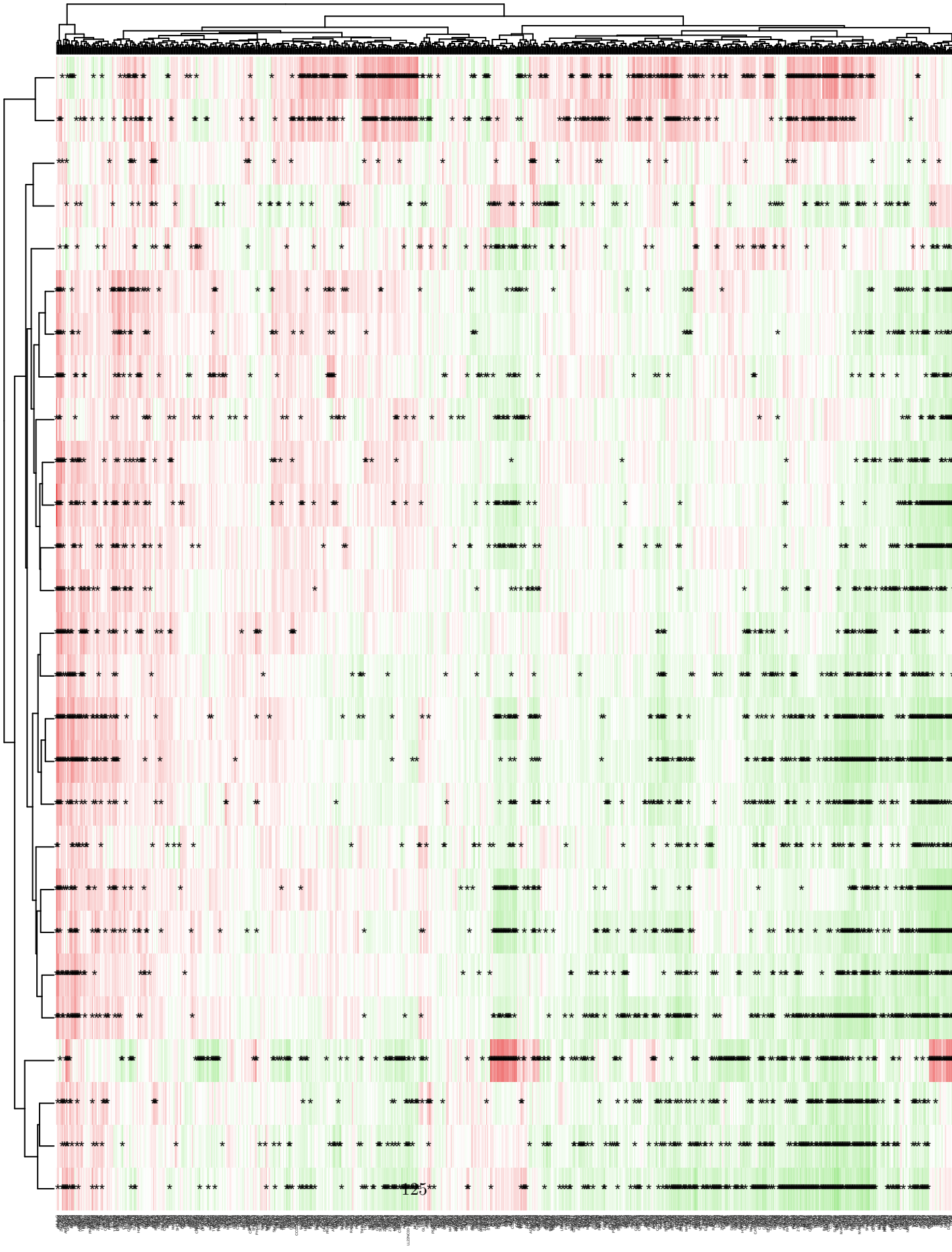
c- All CTA

```
data <- subset(df_z_scores_63, df_z_scores_63$CTA == "CTA")
data <- data[, -c(1, 2)]

# Pearson correlation and p-value compute
cor_matrix <- cor(t(df_imm_z_scores_63), t(data), method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, 63)

# Significant p-val
pval_signif <- cor_pval < 0.05

# Heatmap
# pdf('../results/figures/other_plots/correlation_matrix_all_CTA_63.pdf',
# height = 6, width = 20)
heatmap_all_cta_corr_63 <- 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_63 <- draw(heatmap_all_cta_corr_63)
```

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



```r
# Store clusters HOT and COLD
cta_clust <- column_order(heatmap_all_cta_corr_63)

# Create table with cta
df_cta_clusters_hm <- data.frame(Cluster = c(rep("HOT", length(cta_clust[1:404])),
    rep("COLD", length(cta_clust[405:843]))), SYMBOL = c(rownames(data)[cta_clust[1:404]],
    rownames(data)[cta_clust[405:843]]))

# Save write.table(df_cta_clusters_hm, file =
# '../results/clusters_indiv/clusters_cta_pearson_63.tsv',
# sep = '\t', quote = FALSE, row.names = FALSE)
```

```
plot(t(data["MS4A14", ]), df_imm_z_scores_63["Macrophages", ],
    main = "MS4A14 and macrophages correlation", xlab = "MS4A14 Z-scores",
    ylab = "Macrophages Z-scores", pch = 19)
abline(lm(df_imm_z_scores_63["Macrophages", ] ~ t(data["MS4A14",
    ])), col = "red", lwd = 2)
text(2.2, 2.1, paste("R² =", round(cor(t(data["MS4A14", ]),
    df_imm_z_scores_63["Macrophages", ]), 3)))
```

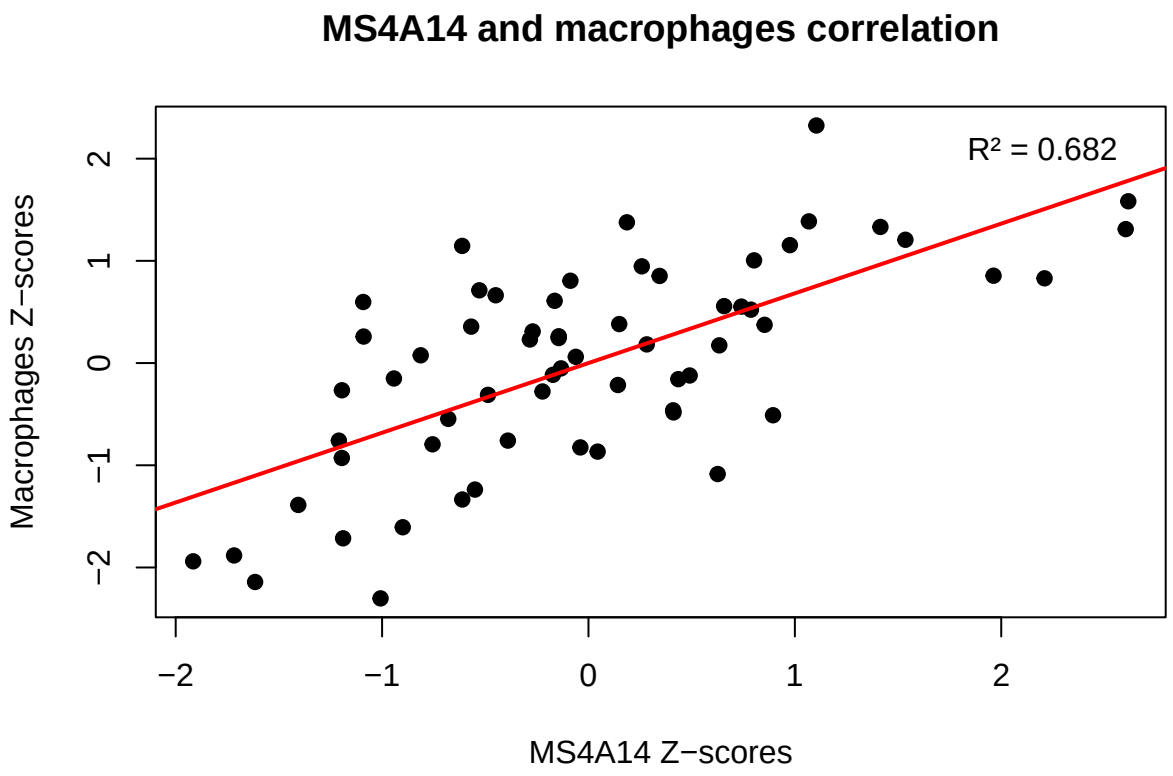


Figure 66: Exemple of positive correlation

```
plot(t(data["CDCA5", ]), df_imm_z_scores_63["pDC", ], main = "CDCA5 and macrophages correlation",
    xlab = "CDCA5 Z-scores", ylab = "pDC Z-scores", pch = 19)
abline(lm(df_imm_z_scores_63["pDC", ] ~ t(data["CDCA5", ])),
    col = "red", lwd = 2)
text(2.2, 2, paste("R² =", round(cor(t(data["MS4A14", ]), df_imm_z_scores_63["pDC",
    ]), 3)))
```

Correlation curve

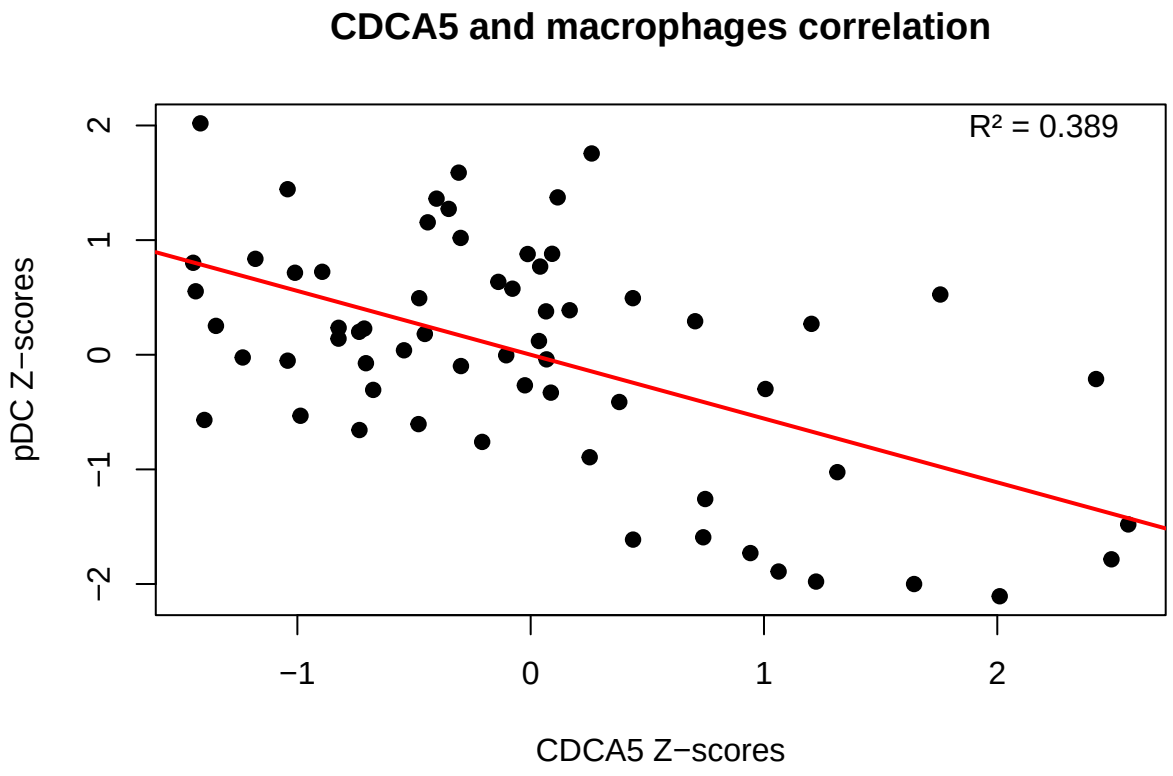


Figure 67: Exemple of negative correlation

d- Not selected CTA

```
# Pearson and p-value compute
cor_matrix <- cor(t(df_imm_z_scores_63), t(data_non_selected_CTA_63),
    method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, ncol(data_non_selected_CTA_63))

# Significant p-value
significant_mask <- cor_pval < 0.05

# Heatmap
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 = 4), 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 (significant_mask[i, j]) {
            grid.text("*", x, y, gp = gpar(fontsize = 8, col = "black"))
        }
    })
```

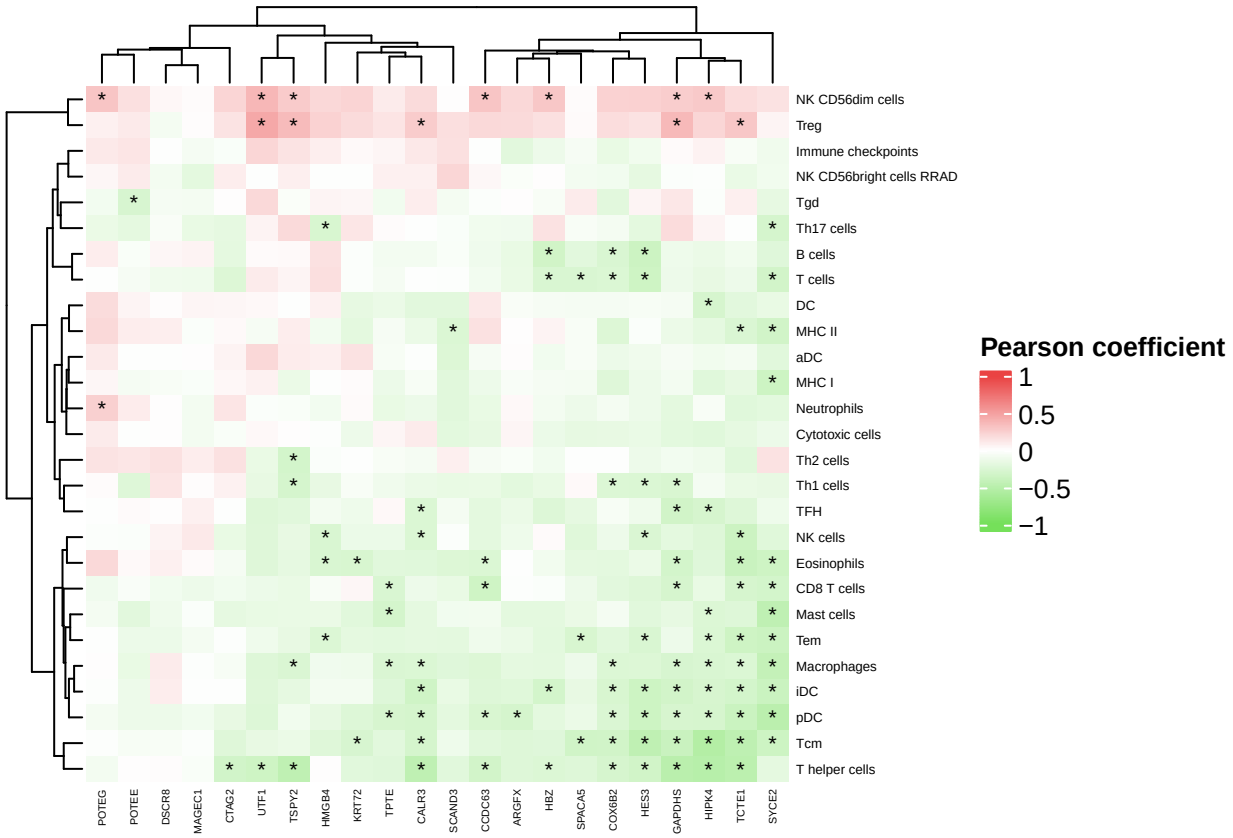


Figure 68: Pearson correlation matrix between non selected significant CTAs and patients with survival data (n = 63)

4) Pearson correlation for all chondrosarcomas

a- Selected CTA

```
# Take data
df_complete_82 <- df_complete[, c("Signature", "CTA", colnames(df_complete)[colnames(df_complete) %in%
    df_metadata_surv_all$Patient])]

# Average the expression between same immune cells types
# Take rows with immune cells signature from normalized
# data
df_avg_immune_sign_82 <- df_complete_82 %>%
    filter(Signature != "NA")

# Group by signature and calculate mean of expression
# values
df_avg_immune_sign_final_82 <- as.data.frame(df_avg_immune_sign_82 %>%
    select(-c(CTA)) %>%
    group_by(Signature) %>%
    summarise(across(where(is.numeric), \(x) mean(x,
        na.rm = TRUE))))

rownames(df_avg_immune_sign_final_82) <- df_avg_immune_sign_final_82$Signature
df_avg_immune_sign_final_82 <- df_avg_immune_sign_final_82[,
    -1]

df_imm_z_scores_82 <- t(scale(t(df_avg_immune_sign_final_82)))
# write.table(df_imm_z_scores_82,
# '../results/imm_sign_z_scores_82.tsv', sep = '\t', quote
# = F)

# Pearson and p-value compute
cor_matrix <- cor(t(df_imm_z_scores_82), t(data_selected_CTA_all),
    method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, ncol(data_selected_CTA_all))

# Significant p-value
significant_mask <- cor_pval < 0.05

# Heatmap
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 (significant_mask[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`.
```

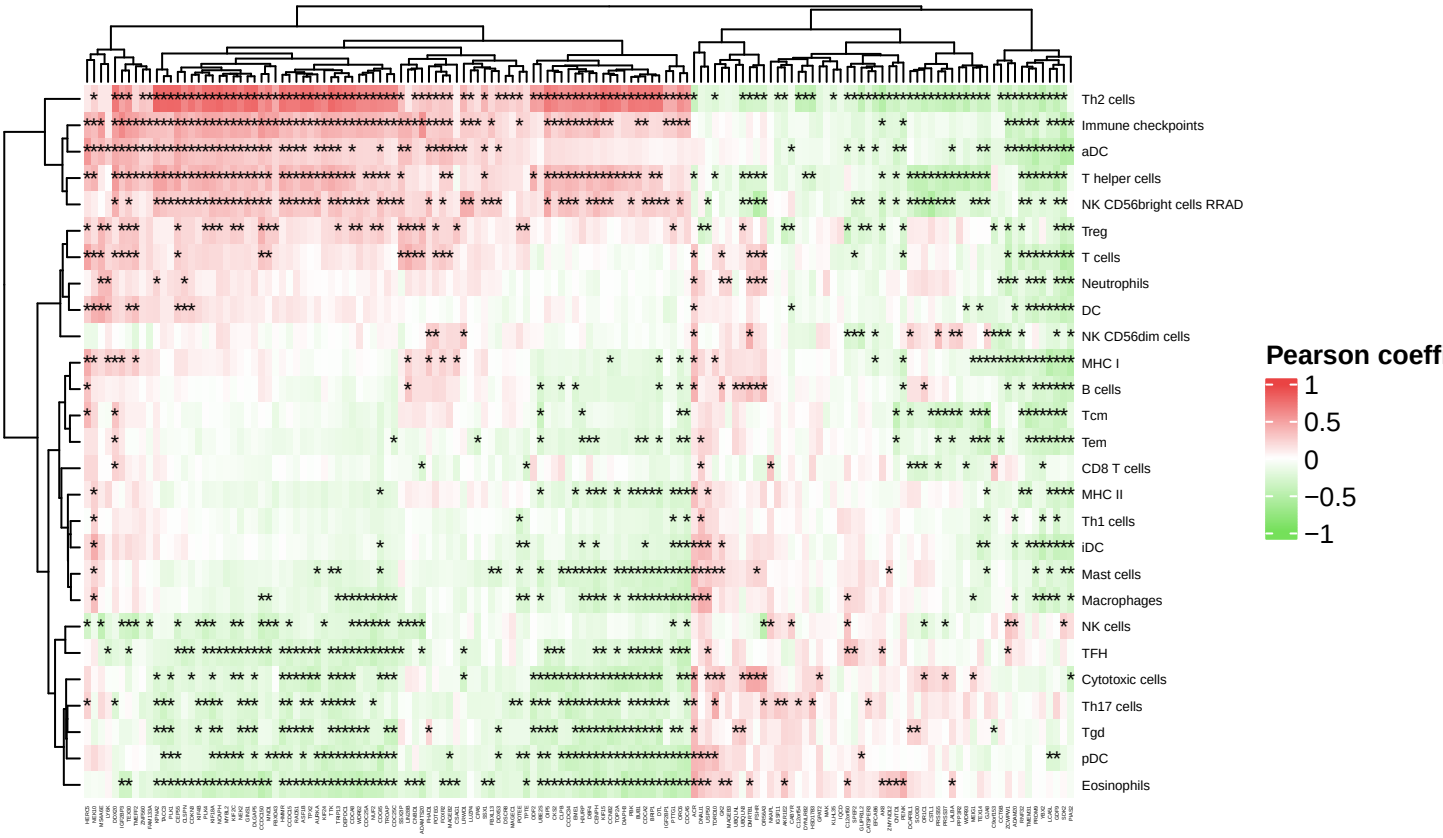


Figure 69: Pearson correlation matrix between selected CTA and immune cells expression (n = 82)

b- All significant CTA

```
# Pearson and p-value compute
cor_matrix <- cor(t(df_imm_z_scores_82), t(data_all_82), method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, ncol(data_all_82))

# Significant p-value
significant_mask <- cor_pval < 0.05

# Heatmap
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 (significant_mask[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`.
```

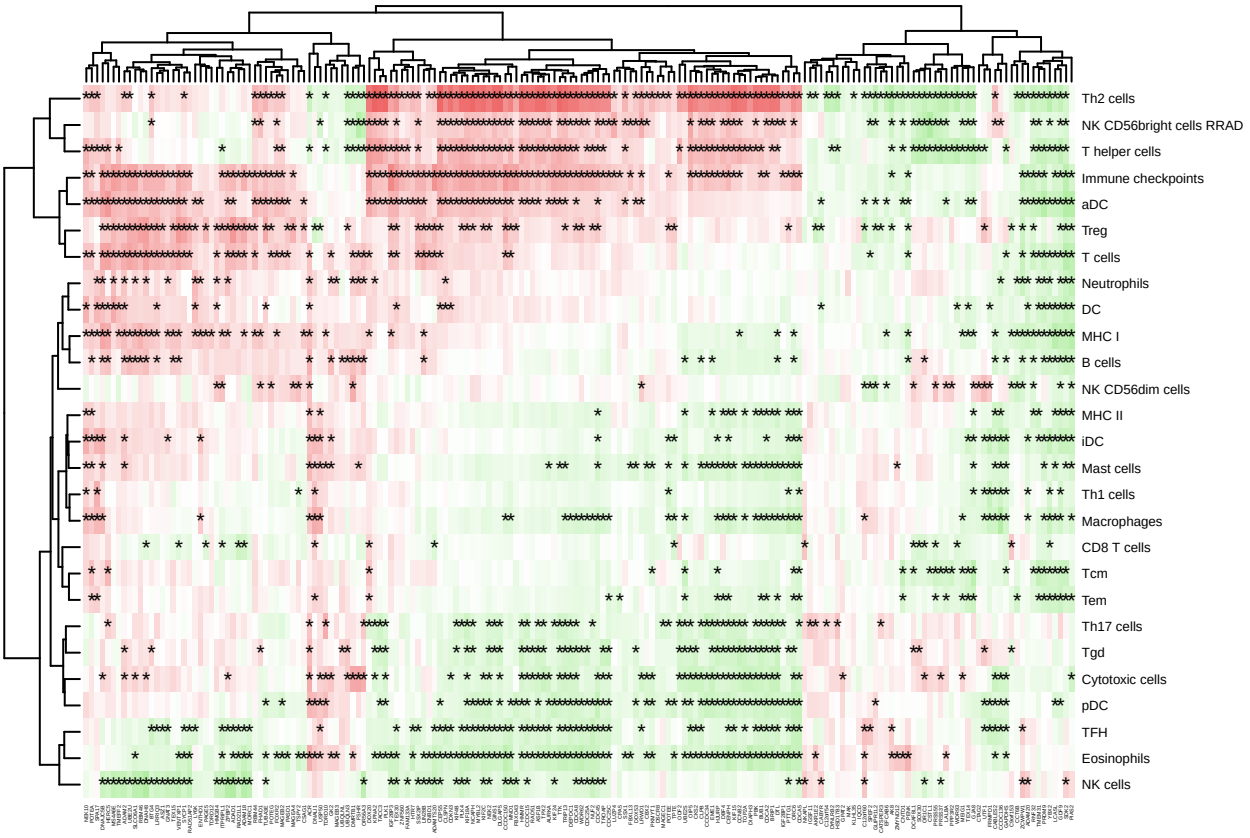


Figure 70: Pearson correlation matrix between signaificant CTA and immune cells expression (n = 82)

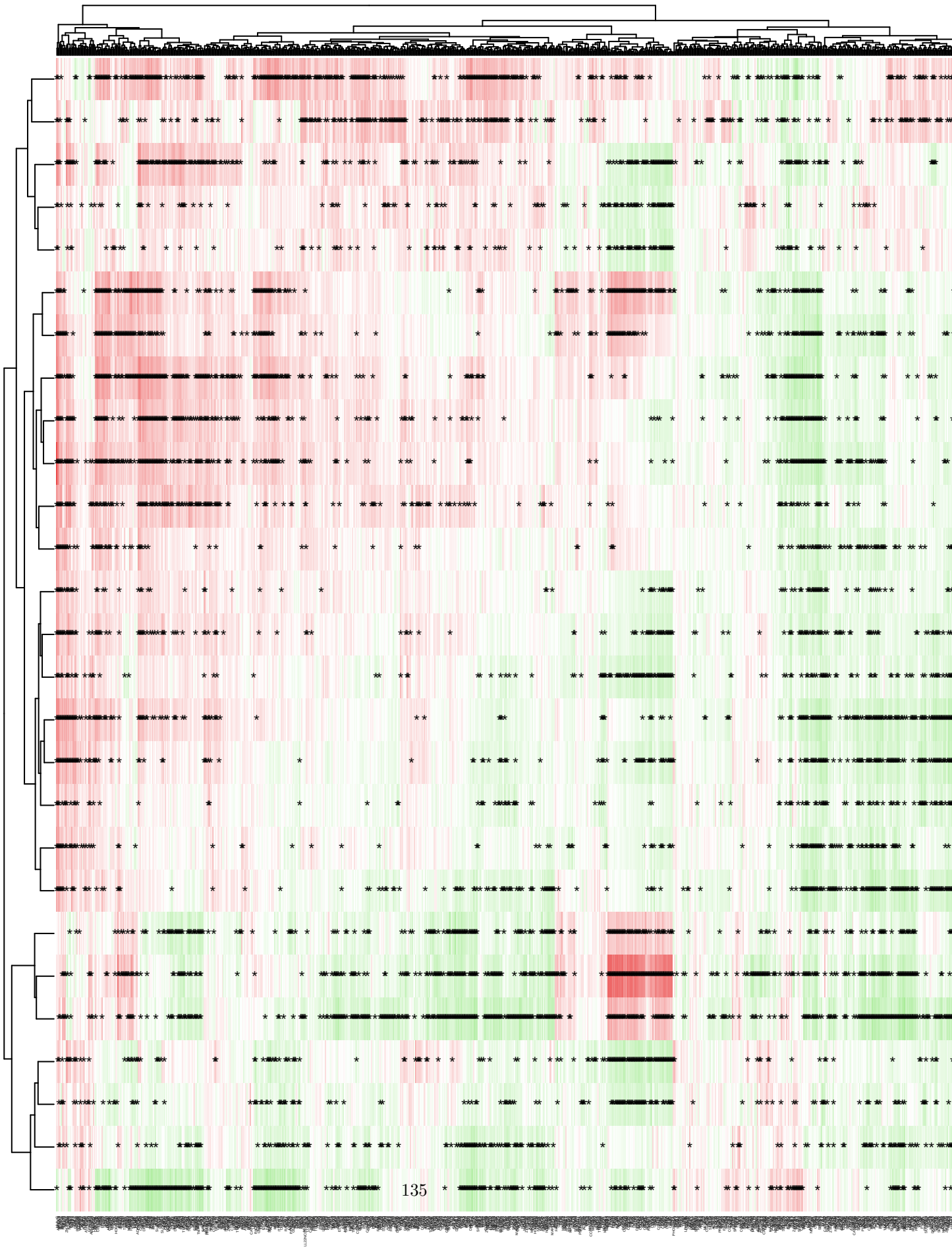
c- All CTA

```
data <- subset(df_z_scores_82, df_z_scores_82$CTA == "CTA")
data <- data[, -c(1, 2)]

# Pearson correlation and p-value compute
cor_matrix <- cor(t(df_imm_z_scores_82), t(data), method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, 82)

# Significant p-val
pval_signif <- cor_pval < 0.05

# Heatmap
# pdf('../results/figures/other_plots/correlation_matrix_all_CTA_82.pdf',
# height = 6, width = 25)
heatmap_all_cta_corr_82 <- 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_82 <- draw(heatmap_all_cta_corr_82)
```

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



```r
# Store clusters HOT and COLD
cta_clust <- column_order(heatmap_all_cta_corr_82)

# Create table with cta
df_cta_clusters_hm <- data.frame(Cluster = c(rep("HOT", length(cta_clust[1:464])),
    rep("Immune suppressed", length(cta_clust[465:574])), rep("COLD",
        length(cta_clust[575:843]))), SYMBOL = c(rownames(data)[cta_clust[1:464]],
    rownames(data)[cta_clust[465:574]], rownames(data)[cta_clust[575:843]]))

# Save write.table(df_cta_clusters_hm, file =
# '../results/clusters_indiv/clusters_cta_pearson_82.tsv',
# sep = '\t', quote = FALSE, row.names = FALSE)
```

d- Not selected CTA

```r
# Pearson and p-value compute
cor_matrix <- cor(t(df_imm_z_scores_82), t(data_non_selected_CTA_82),
    method = "pearson")
cor_pval <- corPvalueStudent(cor_matrix, ncol(data_non_selected_CTA_82))

# Significant p-value
significant_mask <- cor_pval < 0.05

# Heatmap
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 = 4), 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 (significant_mask[i, j]) {
            grid.text("*", x, y, gp = gpar(fontsize = 8, col = "black"))
        }
    })
```

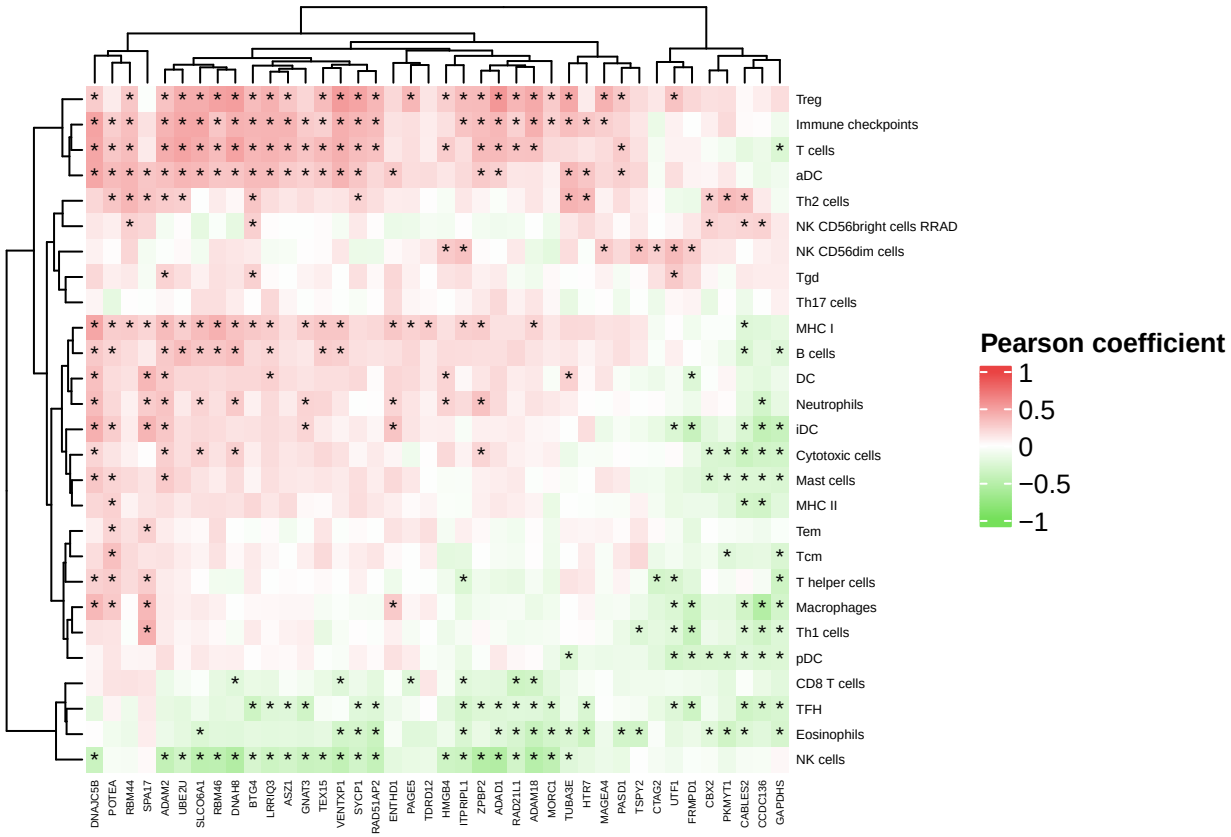


Figure 72: Pearson correlation matrix between non selected significant CTAs and patients with survival data (n = 82)