

Chondrosarcoma expression analysis

Léa ROGUE

2025-02-04

List of Figures

1	Heatmap of relative expression of all CTAs (n = 102)	9
2	Heatmap of CTAs that impact survival (n = 82)	11
3	Heatmap of CTAs that impact survival (n = 63)	13
4	Heatmap and hierarchical clustering of relative immune cells expression (n = 102)	14
5	Elbow plot for k-means clustering	15
6	Heatmap with k-means clustering (k = 2) of immune cells expression (n = 102)	17
7	Volcano plot of DEG between C1 and C2 (from fig. 6)	19
8	Heatmap with k-means clustering (k = 4) of immune cells expression (n = 102)	20
9	Volcano plot of DEG between C1 and C4 (from fig. 8)	23
10	Volcano plot of differentially expressed CTA between C1 and C4 (from fig. 8)	24
11	Intensities boxplot for SDK2 in C1 and C4 (from fig. 8)	25
12	Histogram of the average intensities for all genes	26
13	Heatmap of MHC genes expression (n = 102)	27
14	Heatmap with k-means clustering (k = 4) of immune cells expression (+ MHC) (n = 102) . .	29
15	Volcano plot of DEG between C1 and C4 (from fig. 14)	31
16	Volcano plot of differentially expressed CTA between C1 and C4 (from fig. 14)	33
17	Heatmap with k-means clustering (k = 2) of immune cells expression (+ MHC) (n = 82) . .	35
18	Volcano plot of DEG between C1 and C2 (from fig. 17)	37
19	Volcano plot of differentially expressed CTA between C1 and C2 (from fig. 17)	38
20	Heatmap with k-means clustering (k = 4) of immune cells expression (+ MHC) (n = 82) . .	39
21	Heatmap with k-means clustering (k = 3) of immune cells expression (+ MHC) (n = 82) . .	40
22	Volcano plot of DEG between C1 and C2 (from fig. 21)	42
23	Volcano plot of DEG between C2 and C3 (from fig. 21)	43
24	Volcano plot of DEG between C1 and C3 (from fig. 21)	44
25	Heatmap and hierarchical clustering on relative immune cells expression (+ MHC) (n = 82) .	45
26	Heatmap with k-means clustering (k = 2) on relative immune cells expression (+ MHC) (n = 82)	46
27	Heatmap with k-means clustering (k = 3) on relative immune cells expression (+ MHC) (n = 82)	47
28	Heatmap with k-means clustering (k = 4) on relative immune cells expression (+ MHC) (n = 82)	48
29	Heatmap and hierarchical clustering of relative immune cells expression (+ MHC) (n = 82) .	49
30	Heatmap of relative immune cells expression without immunosuppressive cells (Treg and Immune checkpoints) (n = 102)	51
31	Heatmap of Treg and Immune checkpoints expression	52
32	Heatmap (from fig. 14) with metadata (n = 102)	54

33	Heatmap (from fig. 17) with metadata (n = 82)	56
34	Heatmap (from fig. 21) with metadata (n = 82)	58
35	PCA plot with histology type (n = 102)	59
36	Volcano plot of DEG between benign and malignant tumors	60
37	Volcano plot of differentially expressed CTA between benign and malignant tumors	61
38	Volcano plot of DEG between benign and malignant tumors	63
39	Volcano plot of DEG between benign and G1	64
40	Volcano plot of DEG between benign and G2	65
41	Volcano plot of DEG between benign and G3	66
42	Heatmap of CTA that impact survival analysis and immune cells expression (n = 63)	68
43	Heatmap with kmeans clustering of CTA that impact survival analysis and immune cells expression (n = 63)	70
44	Dendrogram of conventionnal patients (from fig. 3)	72
45	Corrected dendrogram of conventionnal patients (from fig. 3)	72
46	Scale independence and mean connectivity plots	74
47	Dendrogram of gene clustering based on topological overlap matrix dissimilarity (TOM)	75
48	Dendrogram with module detection	76
49	Pearson correlation matrix between modules and immune cells expression	77

Contents

Load libraries	6
Load data and function	6
I. Relative expression of CTAs	8
1) Global relative expression of CTA	8
2) Relative expression of CTA associated with survival probability across all individuals (with survival metadata)	10
3) Relative expression of CTA associated with survival probability in conventional chondrosarcoma patients (with survival metadata)	12
II. Relative immune cells expression	14
1) Hierarchical clustering	14
2) K-means clustering with k = 2 on patients	15
Differently expressed genes analysis	16
3) Clustering k-means with k = 4	19
a- Differently expressed genes analysis	21
b- Verification of the intensities differences	22
i) SDK2	25
ii) Histogram of intensities for each genes	25
III. MHC genes integration	26
1) MHC genes expression	26
2) MHC genes and immune cells signatures	27
DEG of all patients with MHC clustering with clustering k-means (k = 4)	28
IV. Relative immune cells expression without dedifferentiated and benign patients	34
1) Clustering with kmeans, k = 2 on columns	34
DEG analysis	34
2) Clustering with kmeans, k = 4	38
3) Clustering with k = 3	40
DEG analysis	41
4) Clustering on the rows	45
a- Hierarchical clustering	45
b- Kmeans k = 2	45
c- Kmeans k = 3	46
d- Kmeans k = 4	47
3) Hierarchical clustering	48

V. Relative immune cells expression without immunosuppressive cells	50
1) K-means clustering with k = 4	50
2) Expression of immunosuppressive cells	50
VI. Adding metadata	53
1) Adding metadata on heatmap with all patients	53
2) Adding metadata on heatmap with conventional patients	55
a- k = 2	55
b- k = 3	57
VII. Expression analysis benign vs others	58
1) PCA to see groups	58
2) DEG analysis between benign tumors and malignant tumors	58
3) DEG analysis between all the groups	62
VIII. Exploring the relationship between the expression of CTAs and immune cell infiltration	67
1) Visual exploration	67
a- Hierarchical clustering	67
b- Kmeans clustering	69
2) Weighted correlation network analysis (WGCNA)	71
1) Threshold power selection for WGCNA	71
2) Gene clustering and module detection with the topological similarity matrix (TOM)	73
3) Relationship between gene modules and immune cell types: Correlation heatmap	75

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

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

Load data and function

The data was generated using the previous script on a virtual machine (8 CPUs, 32 GB RAM) provided by IFB Biosphere to leverage additional computational resources, as the analysis involves processing 102 files.

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

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

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

```

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

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

# Matrix with avg on genes per signature in zscores
df_avg_immune_sign_z_scores <- 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)

```

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

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

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

# dev.off()
```

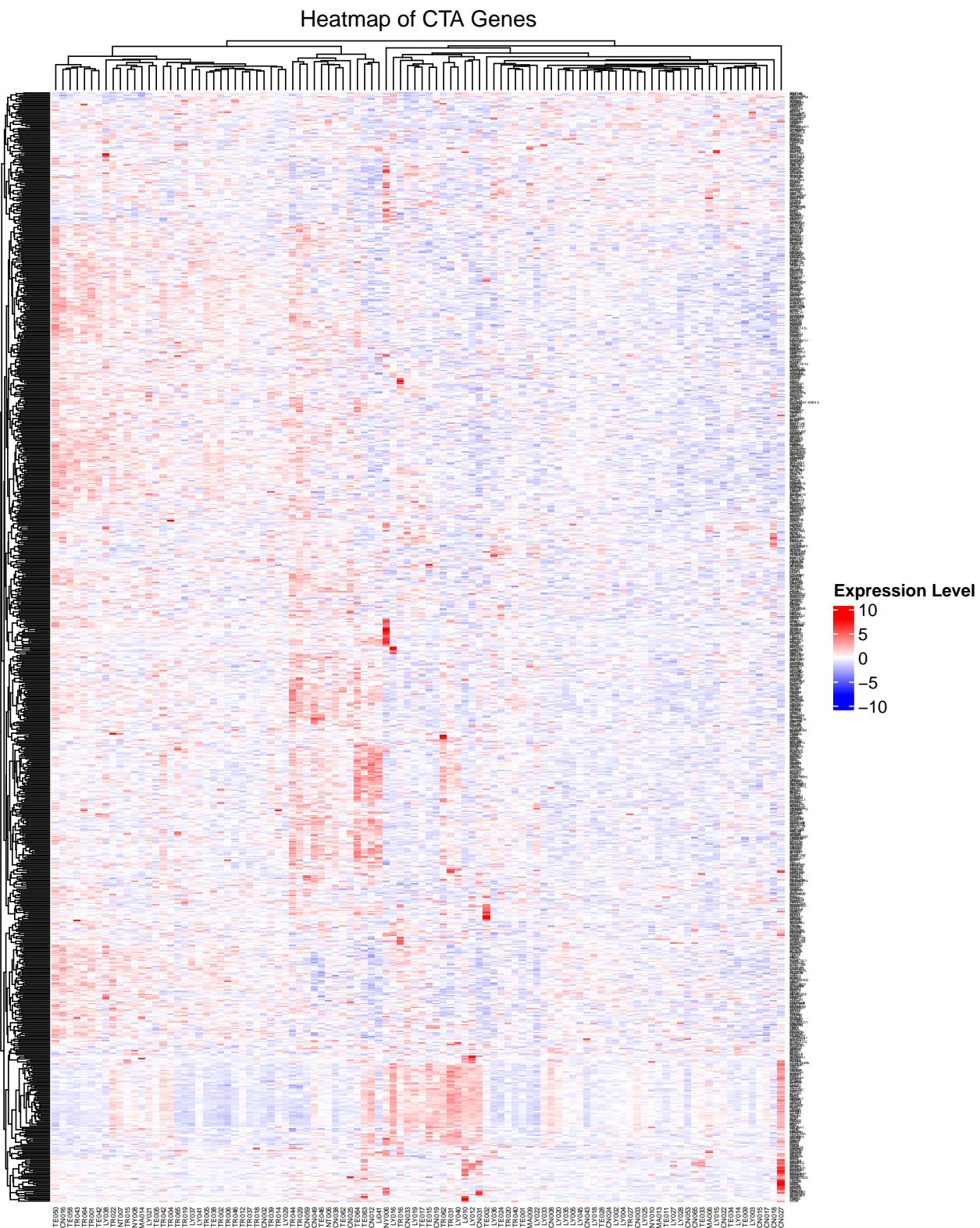


Figure 1: Heatmap of relative expression of all CTAs ($n = 102$)

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

2) 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.

```
# 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$V1
data <- matrix_expr_cta[rownames(matrix_expr_cta) %in% l_CTA_all,
  colnames(matrix_expr_cta) %in% df_metadata_surv_all$Patient]

# Create the heatmap
# pdf('../results/figures/heatmaps/heatmap_cta_coxph_signif_all_indiv.pdf',
# height = 8)
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 Genes",
  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)

# dev.off()
```

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)

# 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[19:52])), rep(3, length(indiv_clust[53:82]))),
  Patient = c(colnames(data)[indiv_clust[1:18]], colnames(data)[indiv_clust[19:52]],
  colnames(data)[indiv_clust[53:82]]))

# 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)
```

Heatmap of CTA Genes

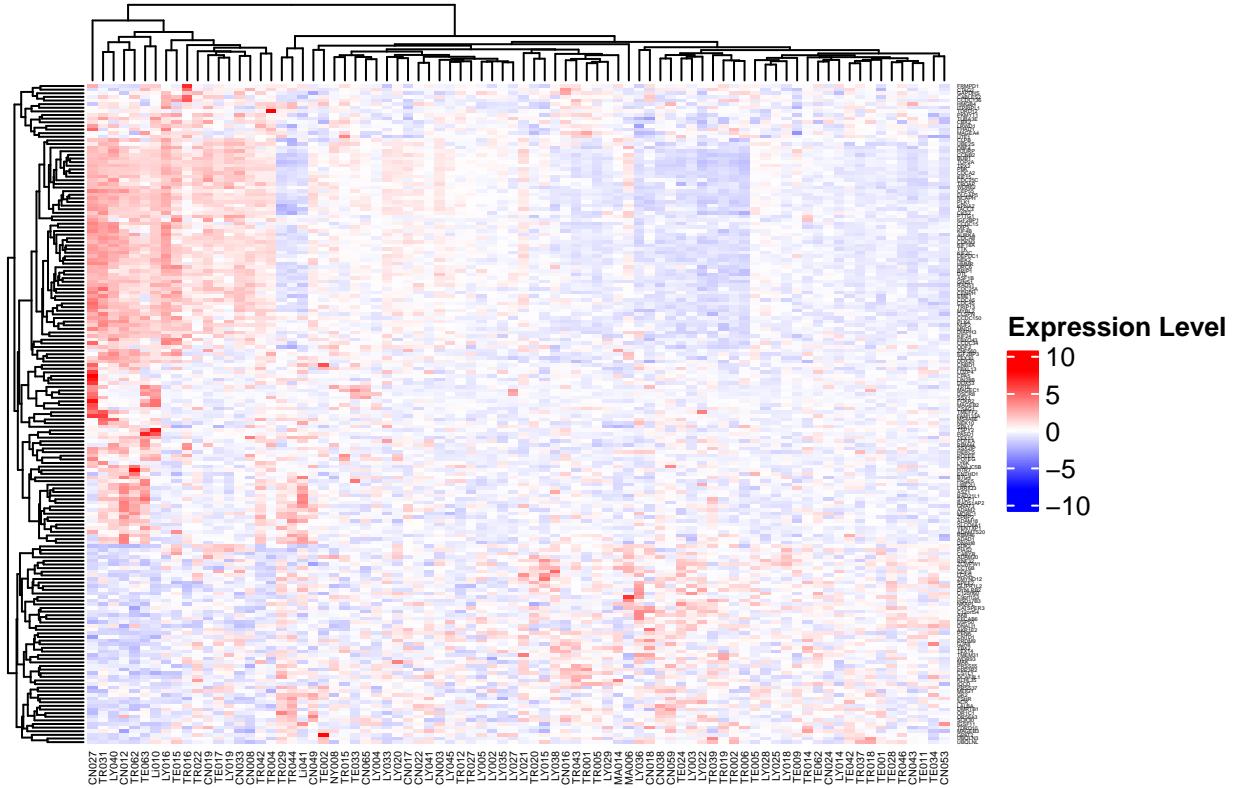


Figure 2: Heatmap of CTAs that impact survival (n = 82)

3) 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.

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

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

# dev.off()
```

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)

# 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[6:37])), rep(3, length(indiv_clust[38:63]))),
  Patient = c(colnames(data)[indiv_clust[1:5]], colnames(data)[indiv_clust[6:37]],
  colnames(data)[indiv_clust[38:63]]))

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

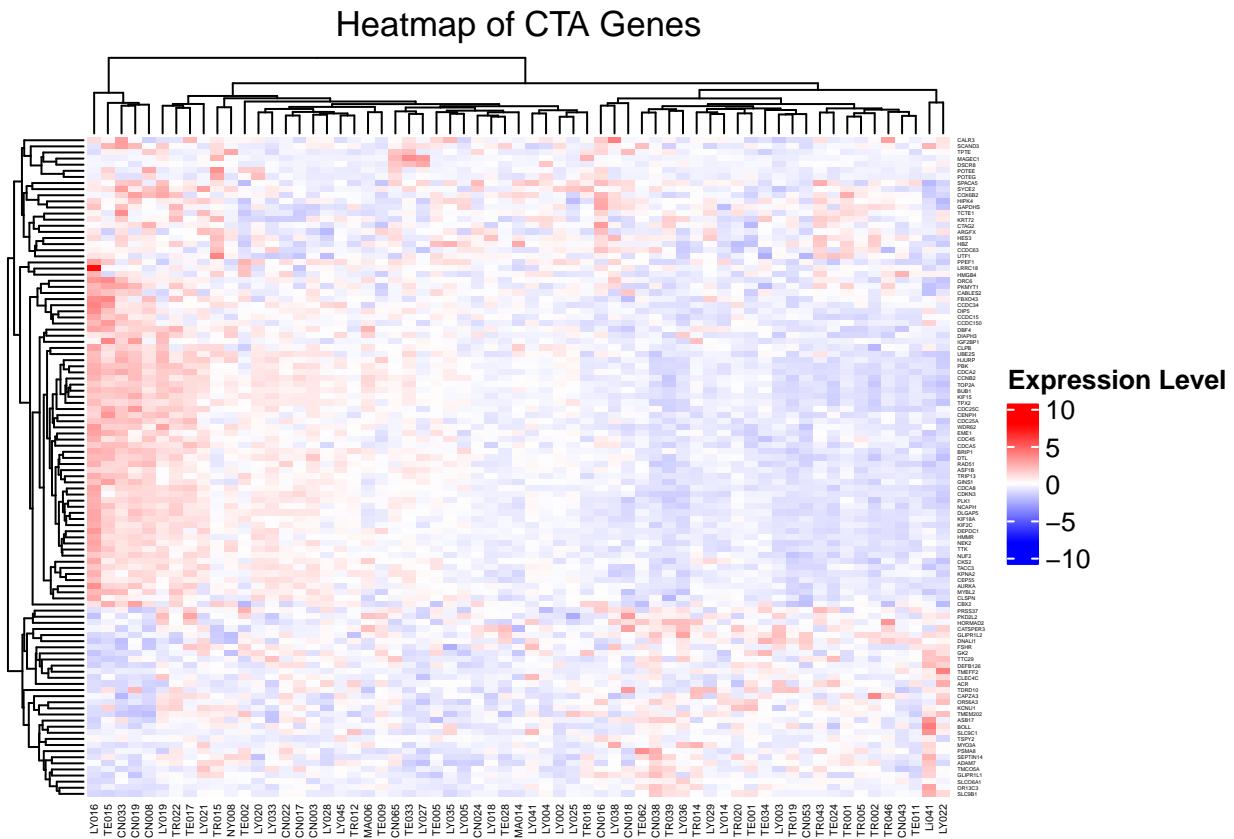


Figure 3: Heatmap of CTAs that impact survival (n = 63)

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)
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"))
```

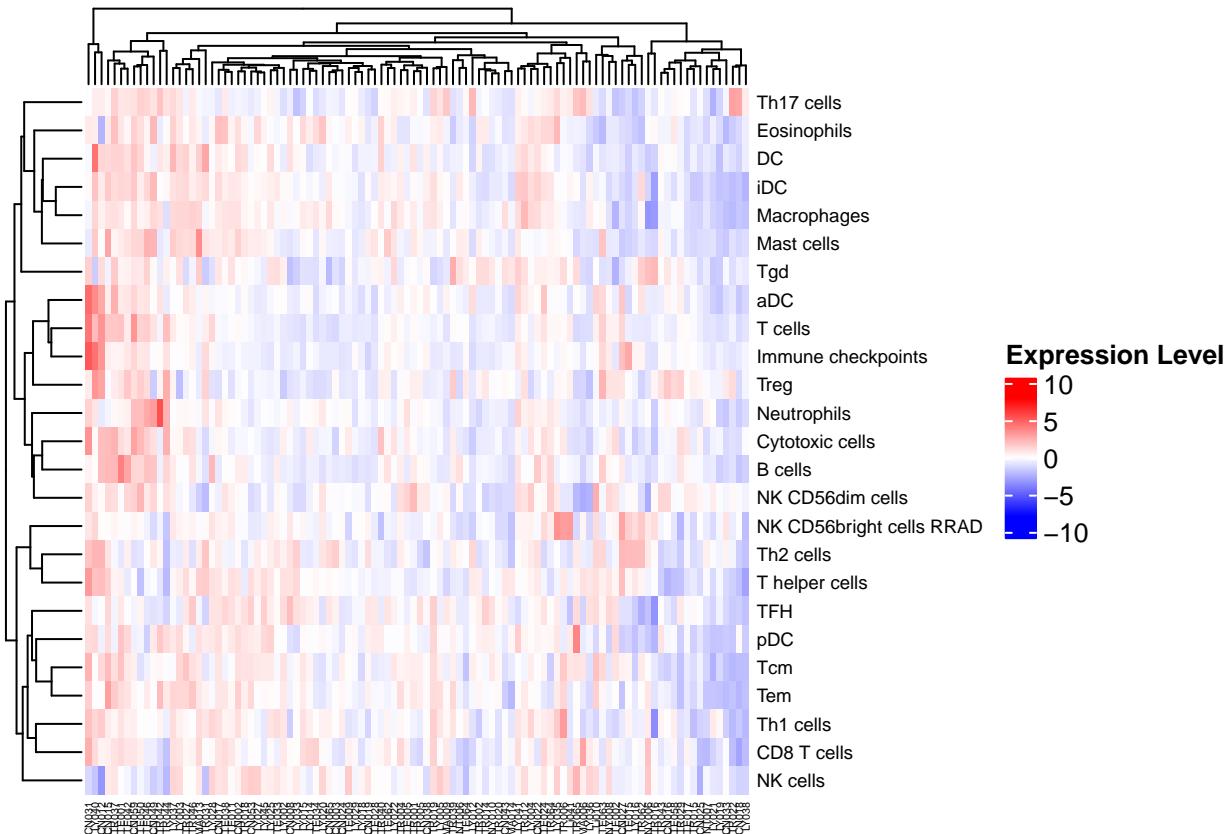


Figure 4: Heatmap and hierarchical clustering of relative immune cells expression (n = 102)

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

```
# 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")
```

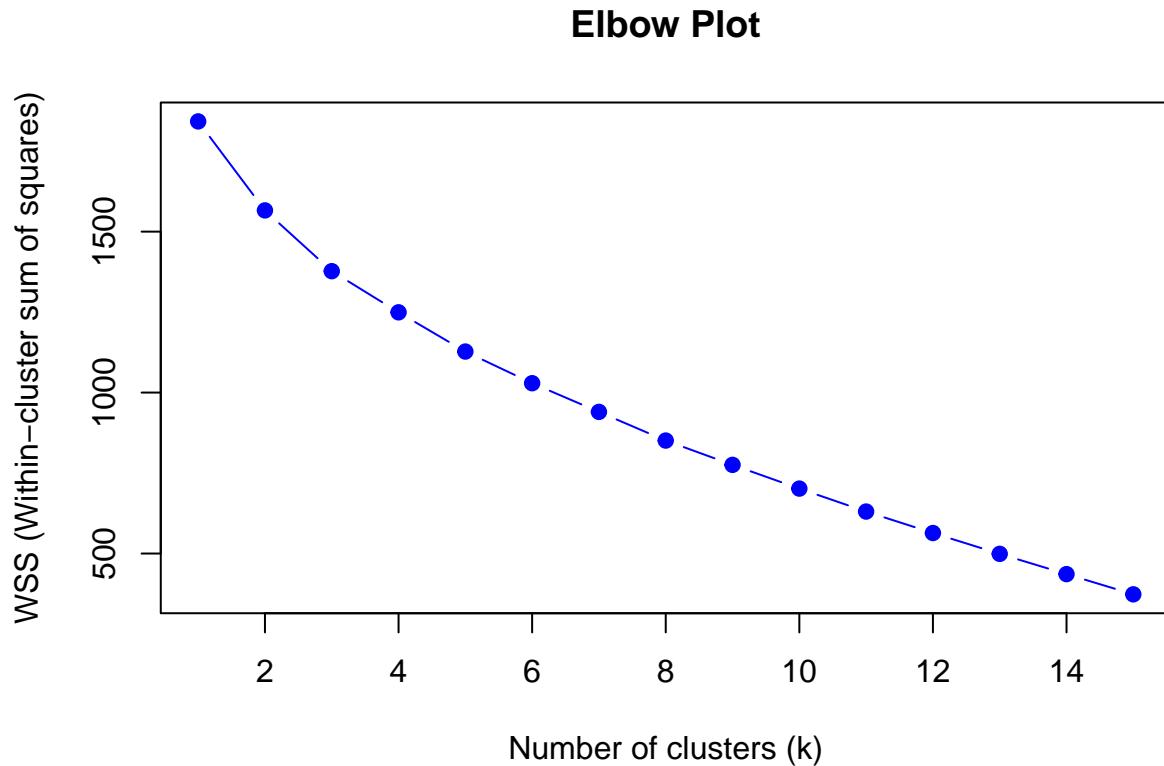


Figure 5: Elbow plot for k-means clustering

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.

```

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

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

# Create heatmap
heatmap <- Heatmap(
  as.matrix(heatmap_data),
  cluster_rows = 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)

# Close the pdf
#dev.off()

```

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)

```

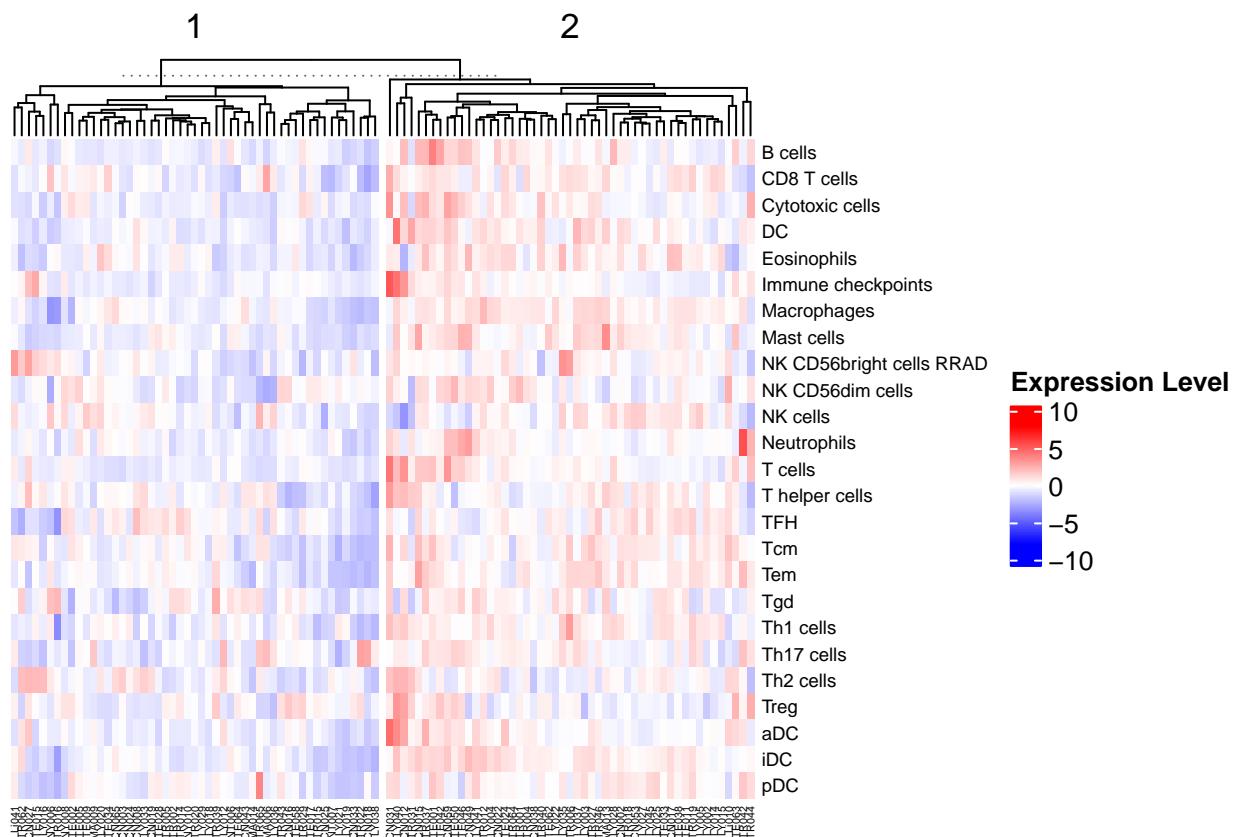


Figure 6: Heatmap with k-means clustering ($k = 2$) of immune cells expression ($n = 102$)

```

# 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)),
                      names(df_CTA_immune_whole_clean_avg))

# 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),
                         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)

# DEG analysis with limma
rownames(df_CTA_immune_whole_clean_avg) <- df_CTA_immune_whole_clean_avg$SYMBOL
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")

# Fit the linear model
data_fit <- lmFit(df_CTA_immune_whole_clean_avg[-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"
```

Volcano plot with all genes

Cluster 1 vs cluster 2 from heatmap k = 2, wo MHC

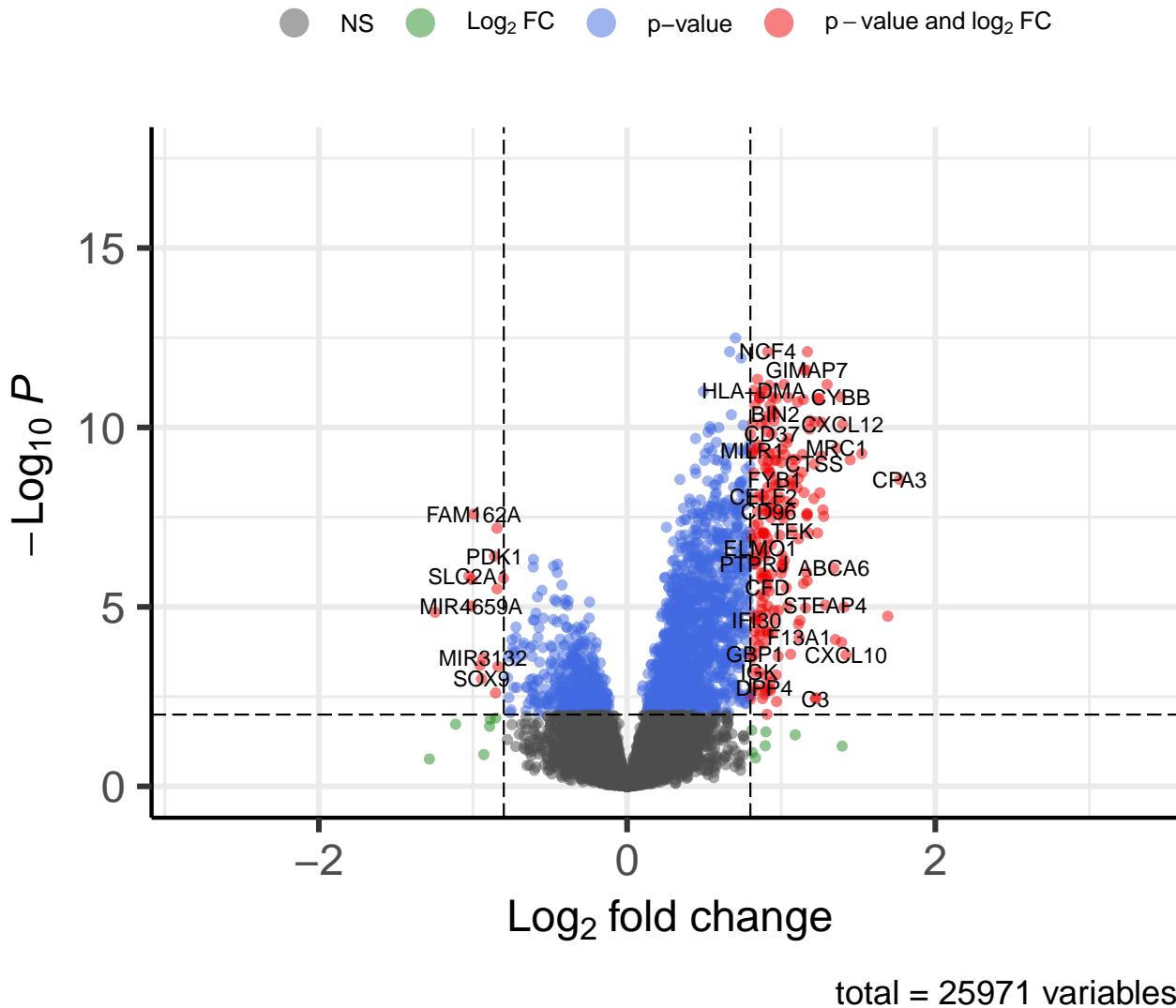


Figure 7: Volcano plot of DEG between C1 and C2 (from fig. 6)

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.

```

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

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

```

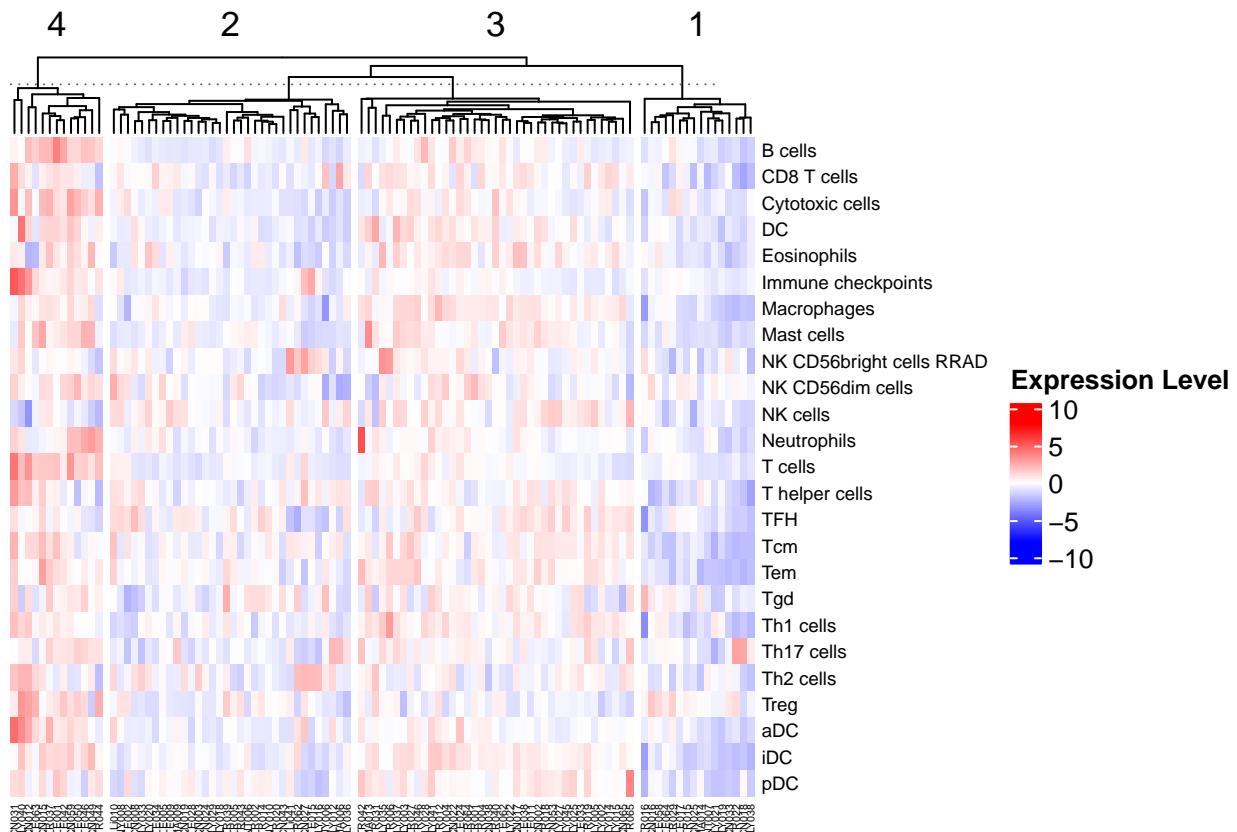


Figure 8: Heatmap with k-means clustering ($k = 4$) of immune cells expression ($n = 102$)

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

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[, 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
data_fit <- lmFit(df_extr, 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_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 %>%
  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 heatmap k=4, wo MHC")

# 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")

```

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 $\log_2\text{FC} > 2$ 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]))

```

Volcano plot with all genes

Cluster HOT vs cluster COLD from heatmap k = 4, wo MHC

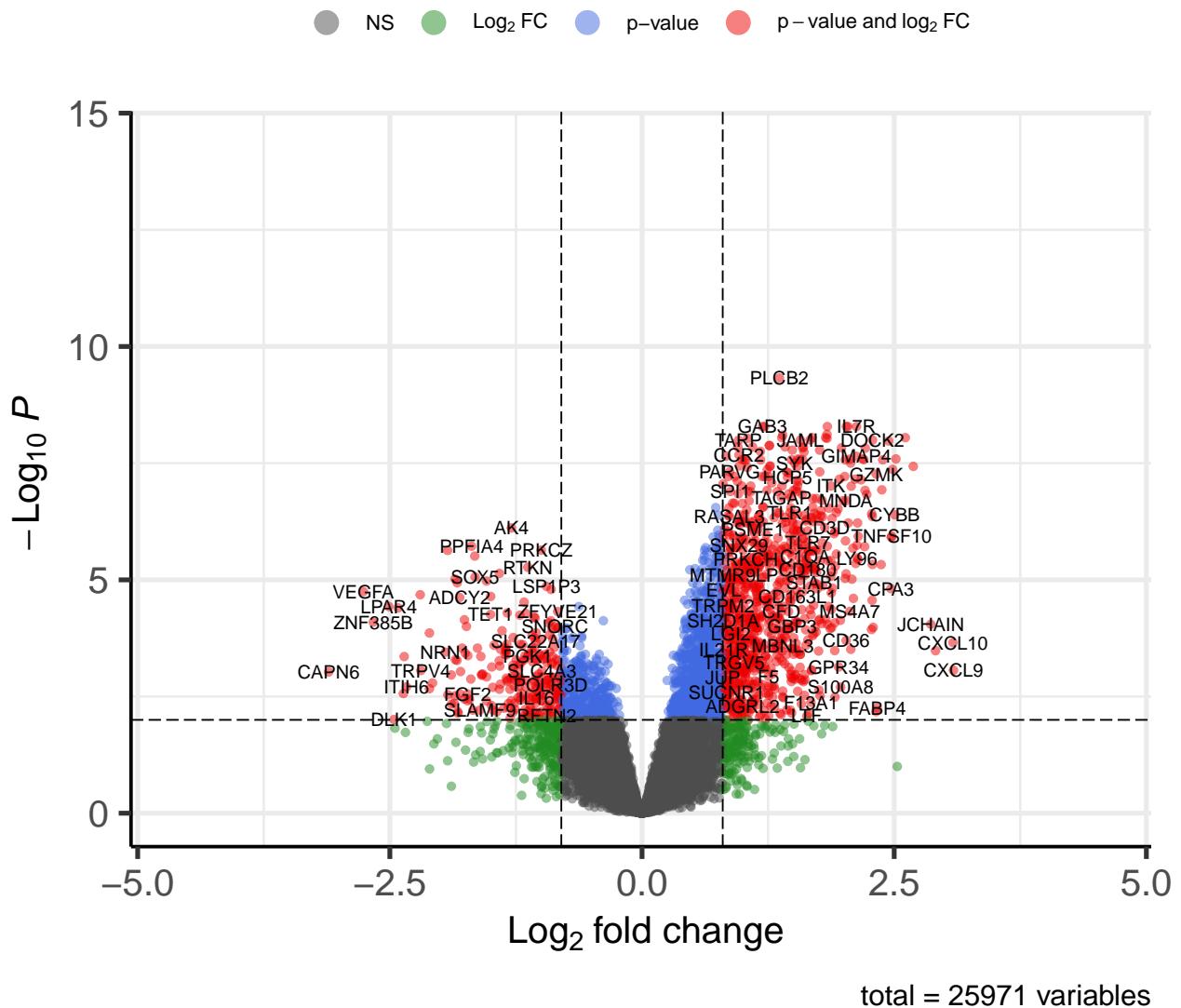


Figure 9: Volcano plot of DEG between C1 and C4 (from fig. 8)

Volcano plot with CTA genes

Cluster HOT vs cluster COLD from heatmap k = 4, wo MHC

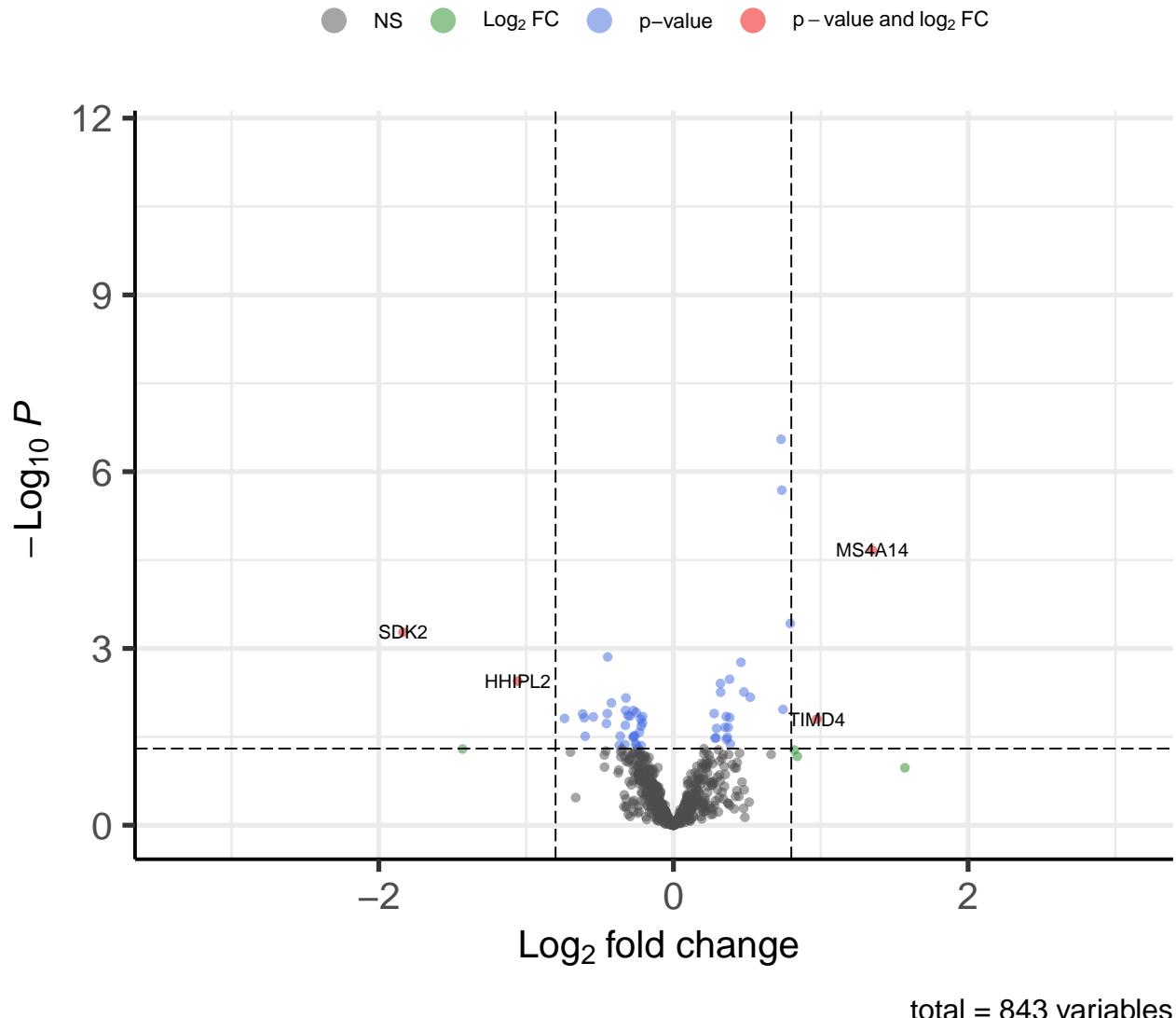


Figure 10: Volcano plot of differentially expressed CTA between C1 and C4 (from fig. 8)

```
# 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"))
```

Boxplot : HOT vs COLD for SDK2

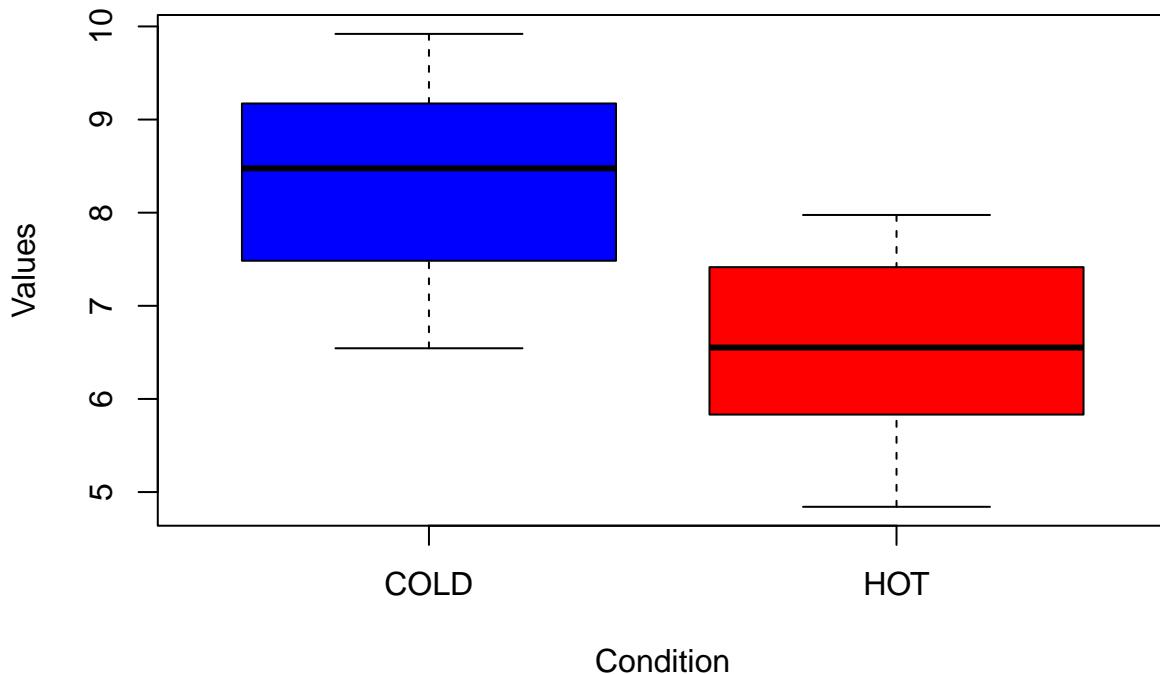


Figure 11: Intensities boxplot for SDK2 in C1 and C4 (from fig. 8)

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")
```

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.

Histogram of the average intensities for all genes

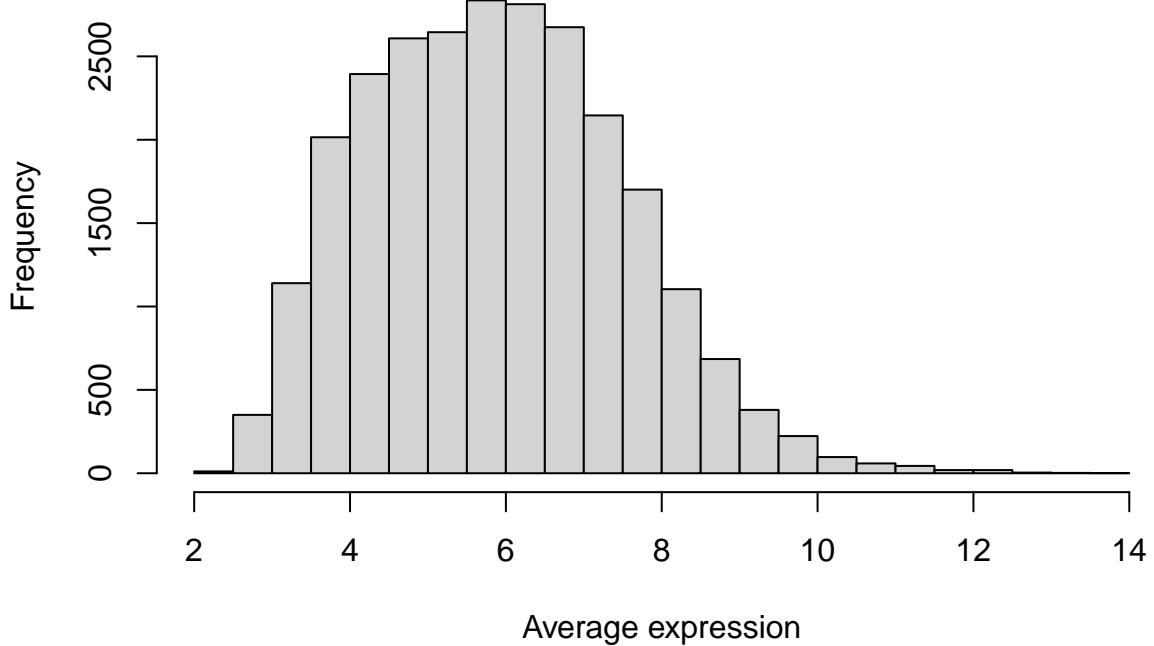


Figure 12: Histogram of the average intensities for all genes

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

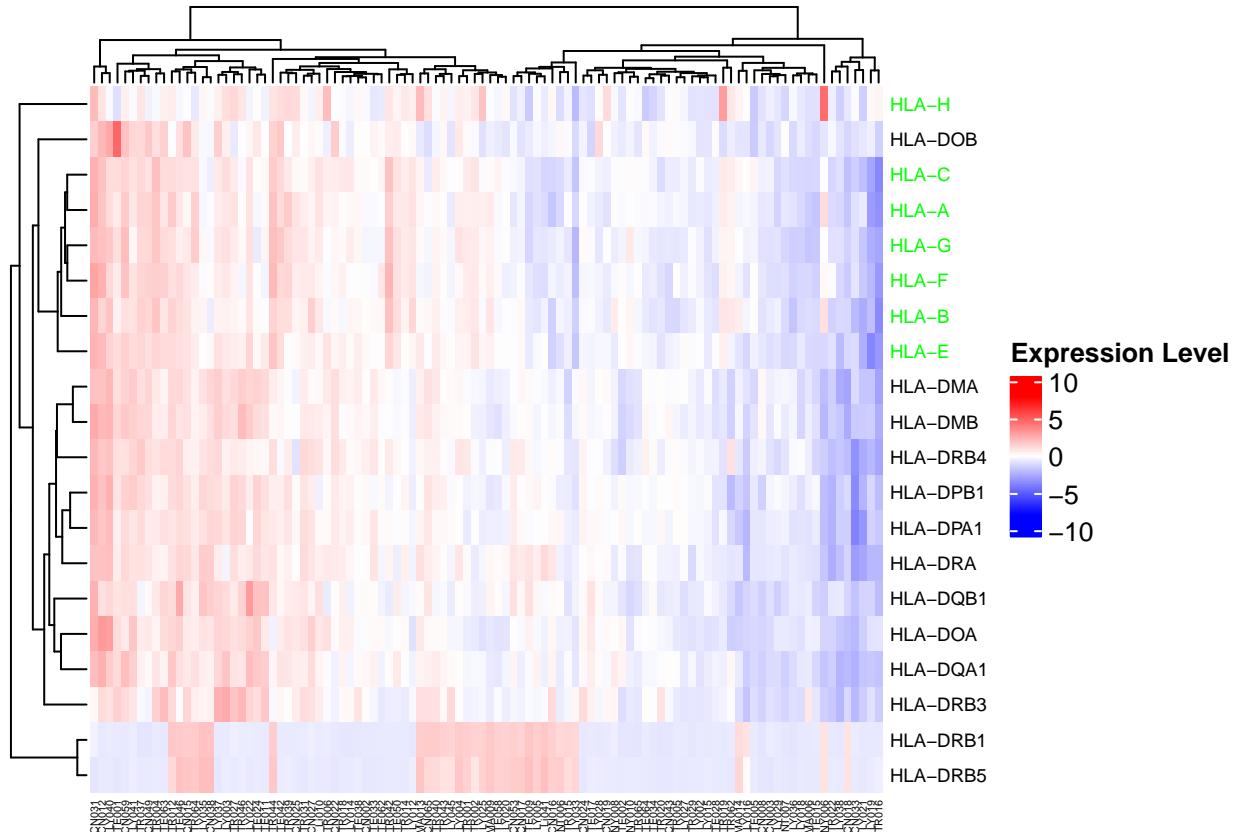


Figure 13: Heatmap of MHC genes expression (n = 102)

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

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 = TRUE)

# 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
#pdf("../results/figures/heatmaps/heatmap_imm_mhc_kmeans4.pdf")

# 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)

```

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

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_data$Cluster = i
  cluster_list[[i]] <- cluster_data
}
```

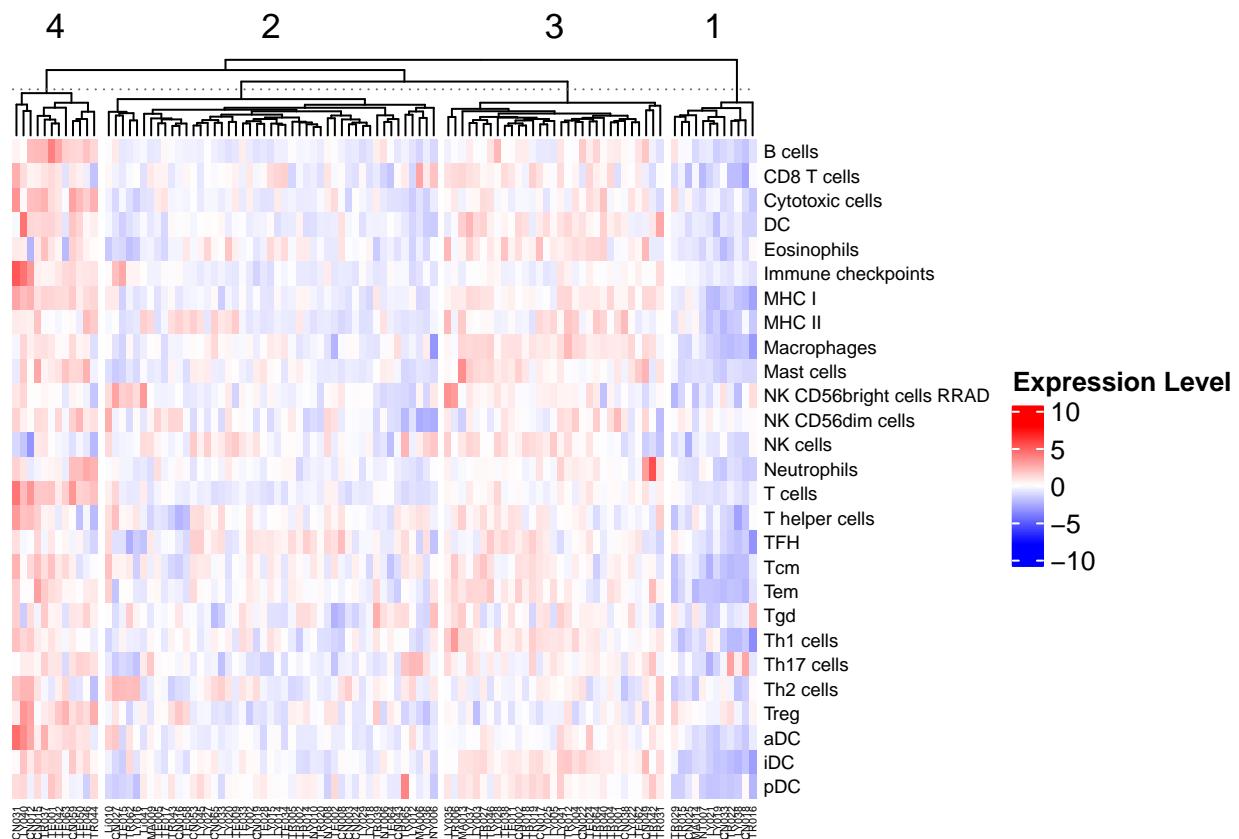


Figure 14: Heatmap with k-means clustering ($k = 4$) of immune cells expression (+ MHC) ($n = 102$)

```

}

# 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[, 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)

# 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 %>%
  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 heatmap k = 4, with MHC"

```

Volcano plot with all genes

Cluster HOT vs cluster COLD from heatmap k = 4, with MHC

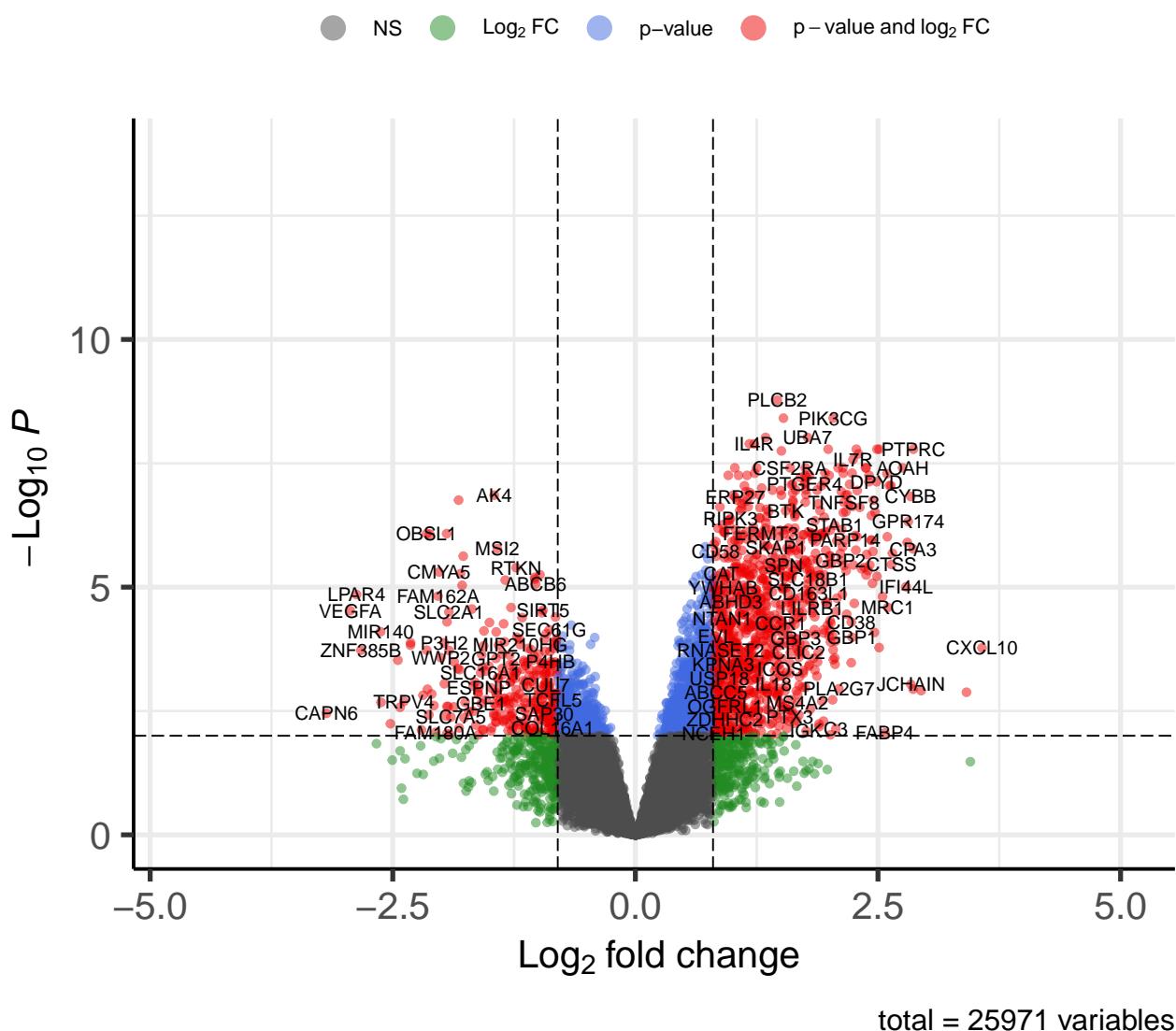


Figure 15: Volcano plot of DEG between C1 and C4 (from fig. 14)

```

# 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')

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

```

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.

Volcano plot with CTA genes

Cluster HOT vs cluster COLD from heatmap k = 4, with MHC

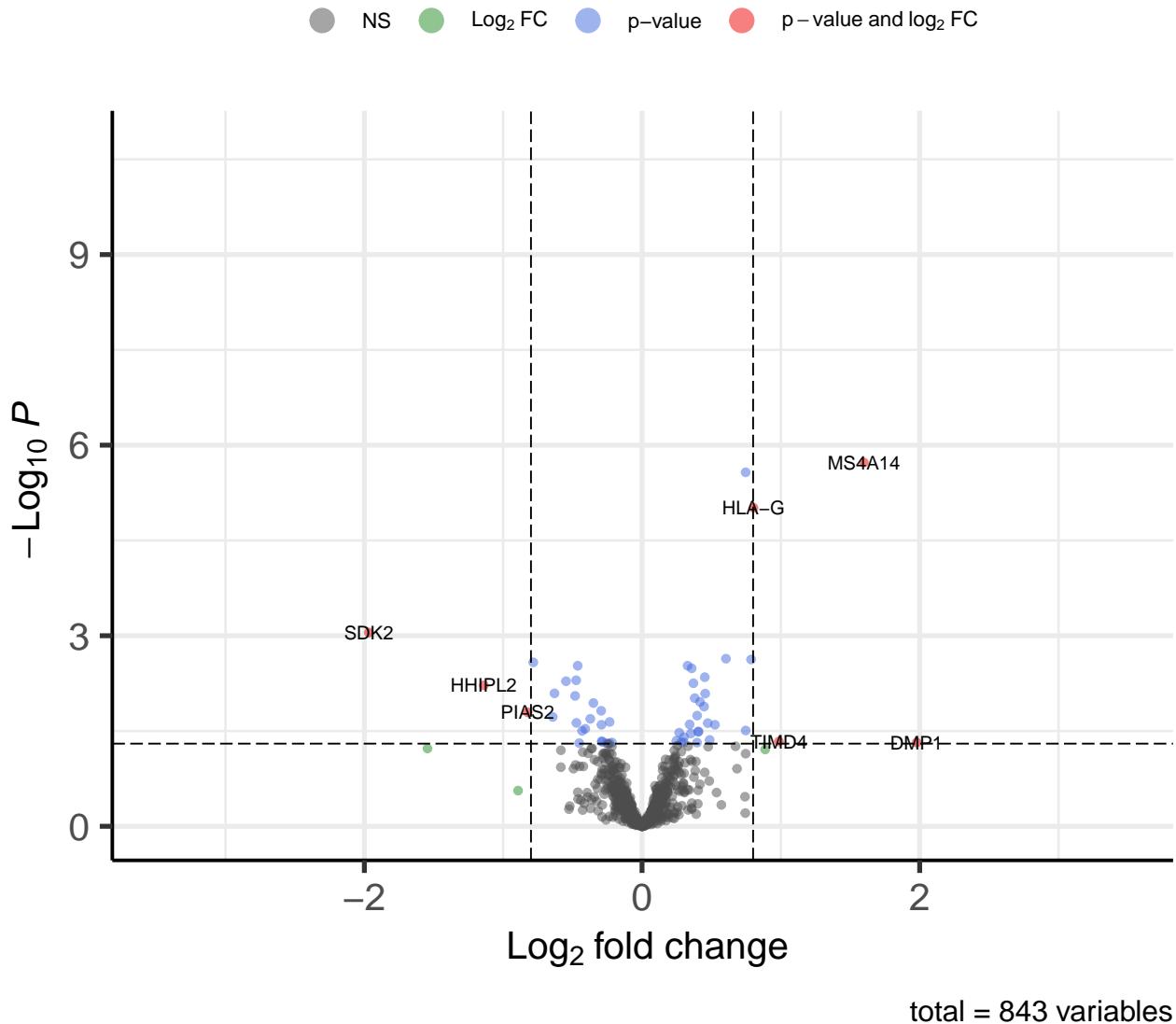


Figure 16: Volcano plot of differentially expressed CTA between C1 and C4 (from fig. 14)

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.

```
heatmap_data_mhc_all_conv <- heatmap_data_mhc_all[, patients_conv]  
df_expr_conv <- df_CTA_immune_whole_clean_avg[, patients_conv]
```

1) Clustering with kmeans, k = 2 on columns

```
set.seed(1)  
heatmap_km2_conv <- Heatmap(  
  as.matrix(heatmap_data_mhc_all_conv),  
  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)
```

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(heatmap_data_mhc_all_conv))[column_clusters[i]],  
    Cluster = i)  
  cluster_list[[i]] <- cluster_data  
}  
  
# Combine all clusters into a single data frame
```

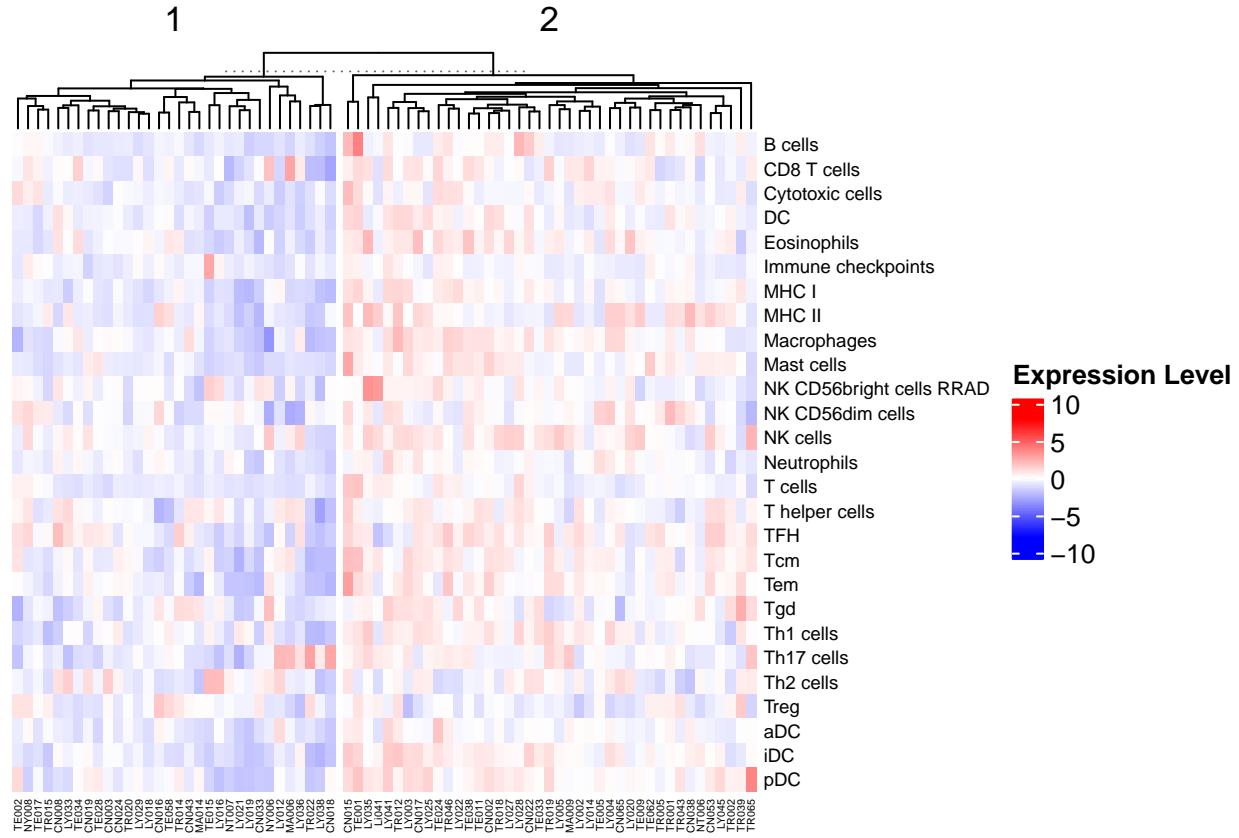


Figure 17: Heatmap with k-means clustering ($k = 2$) of immune cells expression (+ MHC) ($n = 82$)

```

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_expr_conv)), names(df_expr_conv))

# 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_expr_conv), value = TRUE)
  new_row[matching_cols] <- cluster_value
}

# Add the new column
df_whole_cold_hot_km2_conv <- rbind(new_row, df_expr_conv)

# 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")

# Fit the linear model
data_fit <- lmFit(df_expr_conv, 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

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",

```

Volcano plot with all genes

Cluster 1 vs cluster 2 from heatmap k = 2, with MHC

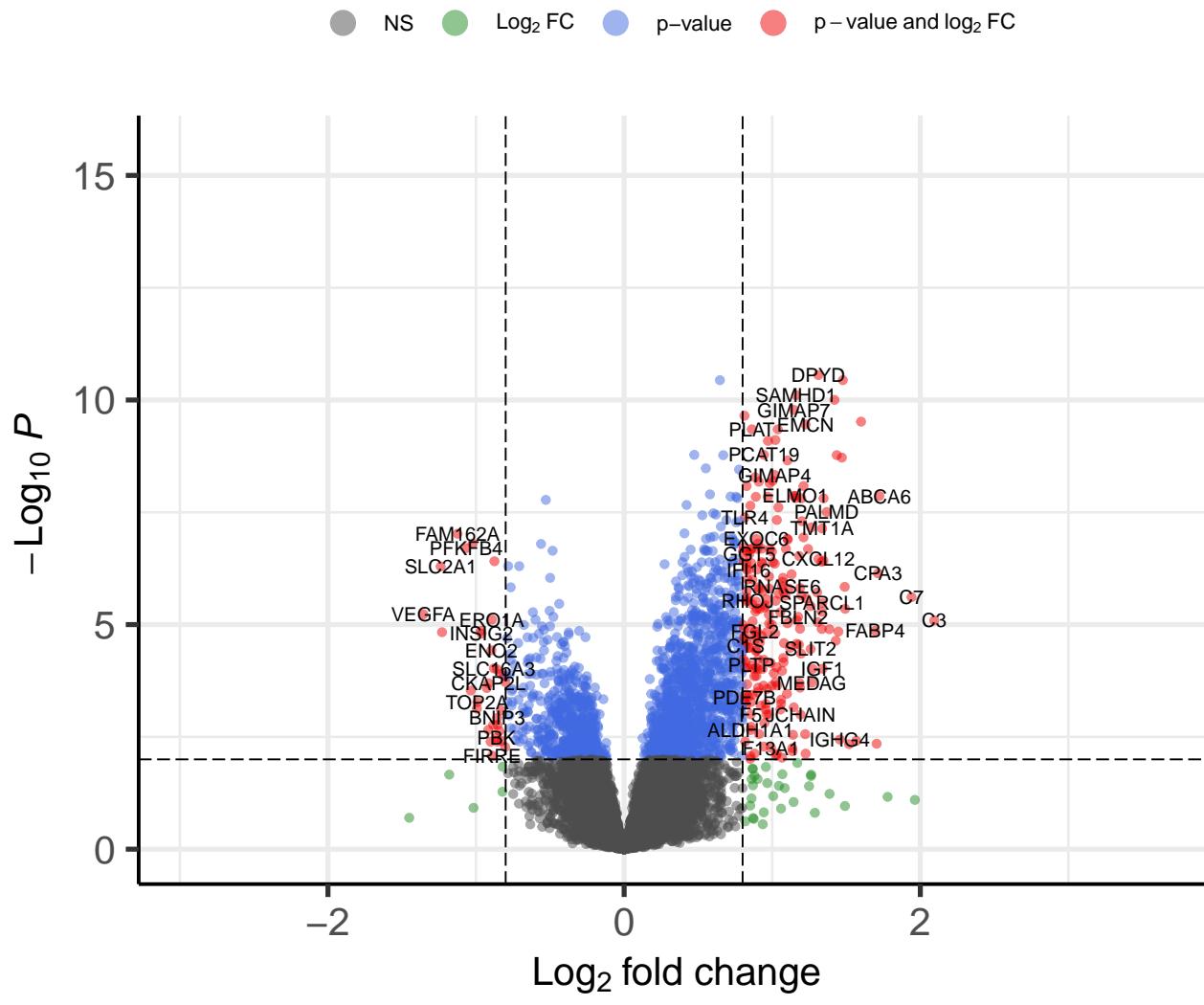


Figure 18: Volcano plot of DEG between C1 and C2 (from fig. 17)

```
subtitle = "Cluster 1 vs cluster 2 from heatmap k = 2, wth MHC")
```

Volcano plot with CTA genes

Cluster 1 vs cluster 2 from heatmap k = 2, wth MHC

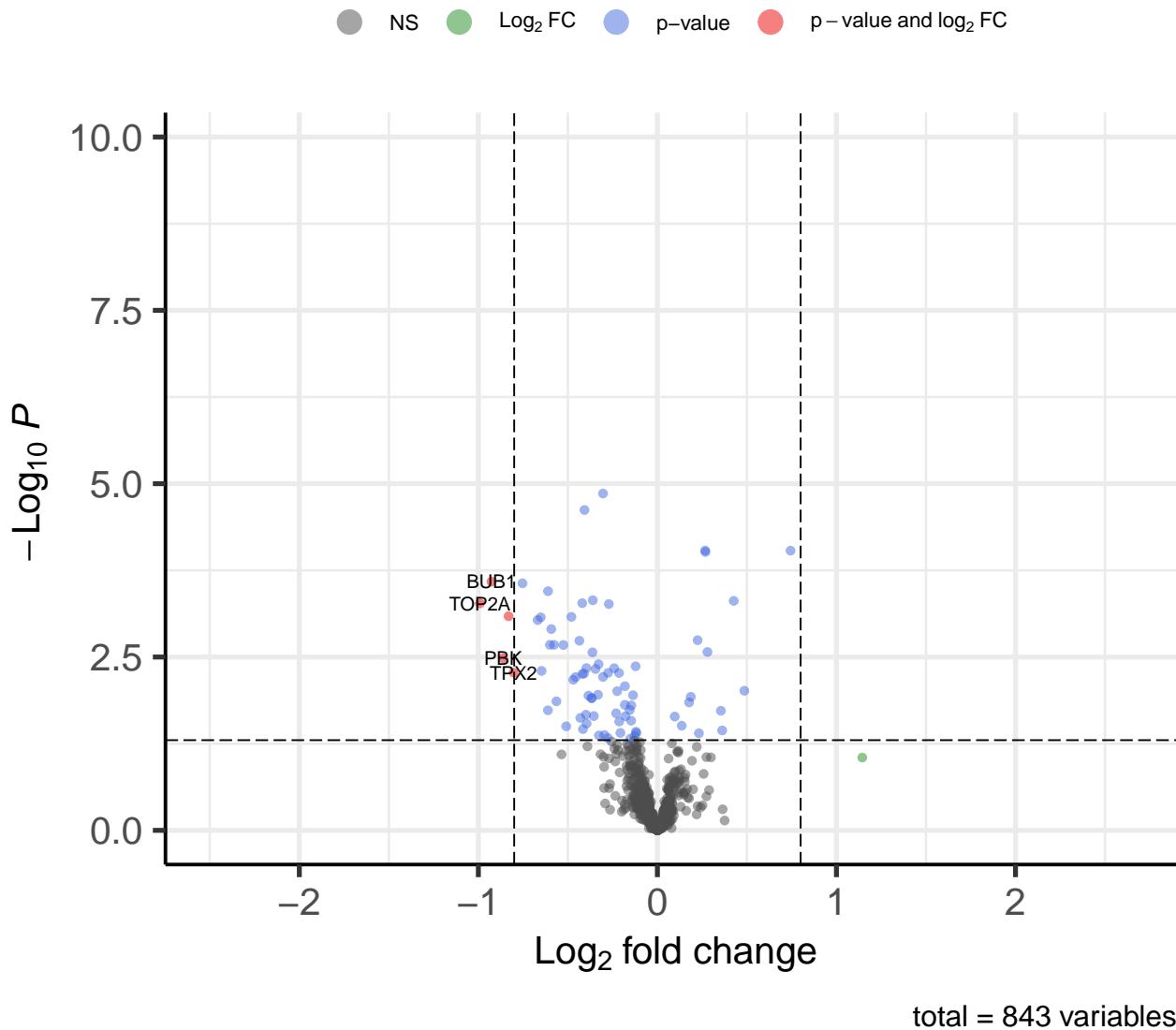


Figure 19: Volcano plot of differentially expressed CTA between C1 and C2 (from fig. 17)

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(heatmap_data_mhc_all_conv),
  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")
)

```

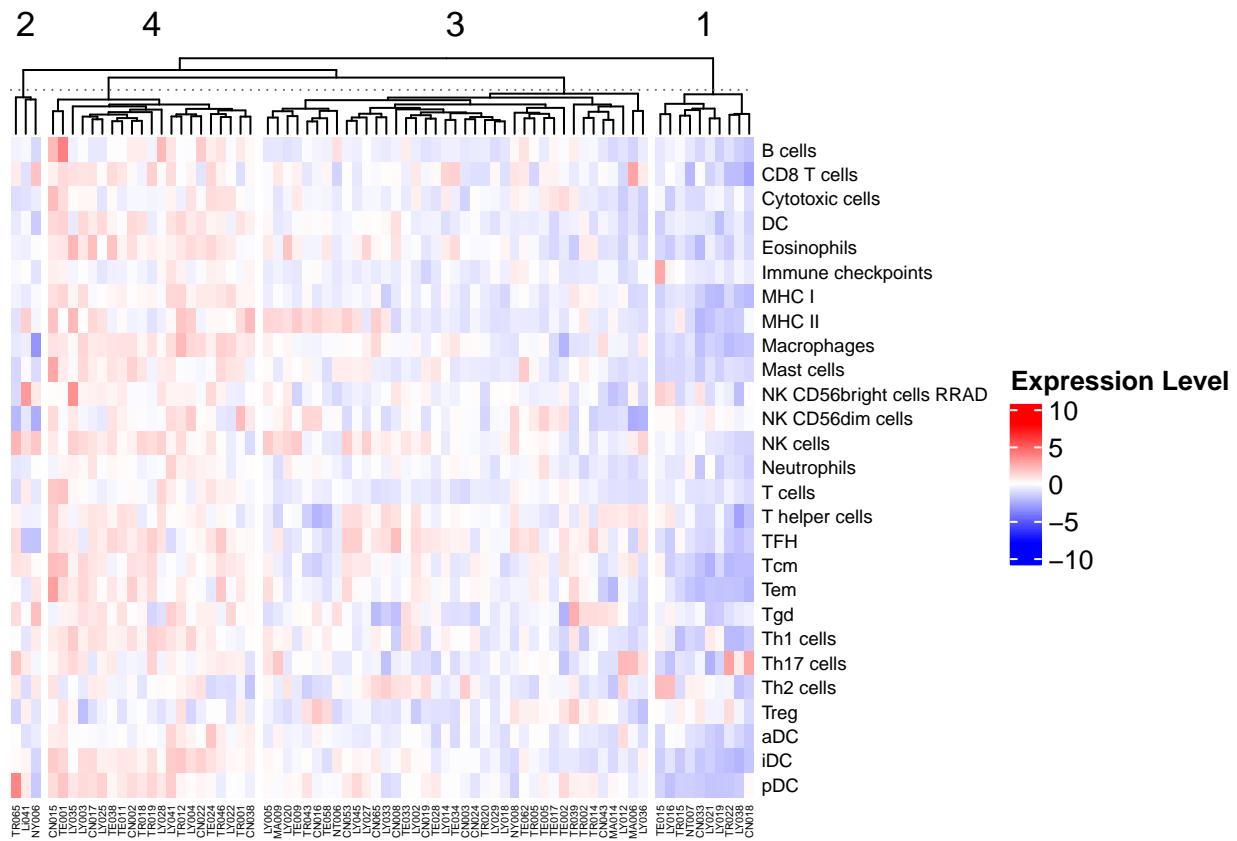


Figure 20: Heatmap with k-means clustering ($k = 4$) of immune cells expression (+ MHC) ($n = 82$)

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(heatmap_data_mhc_all_conv),
  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)

```

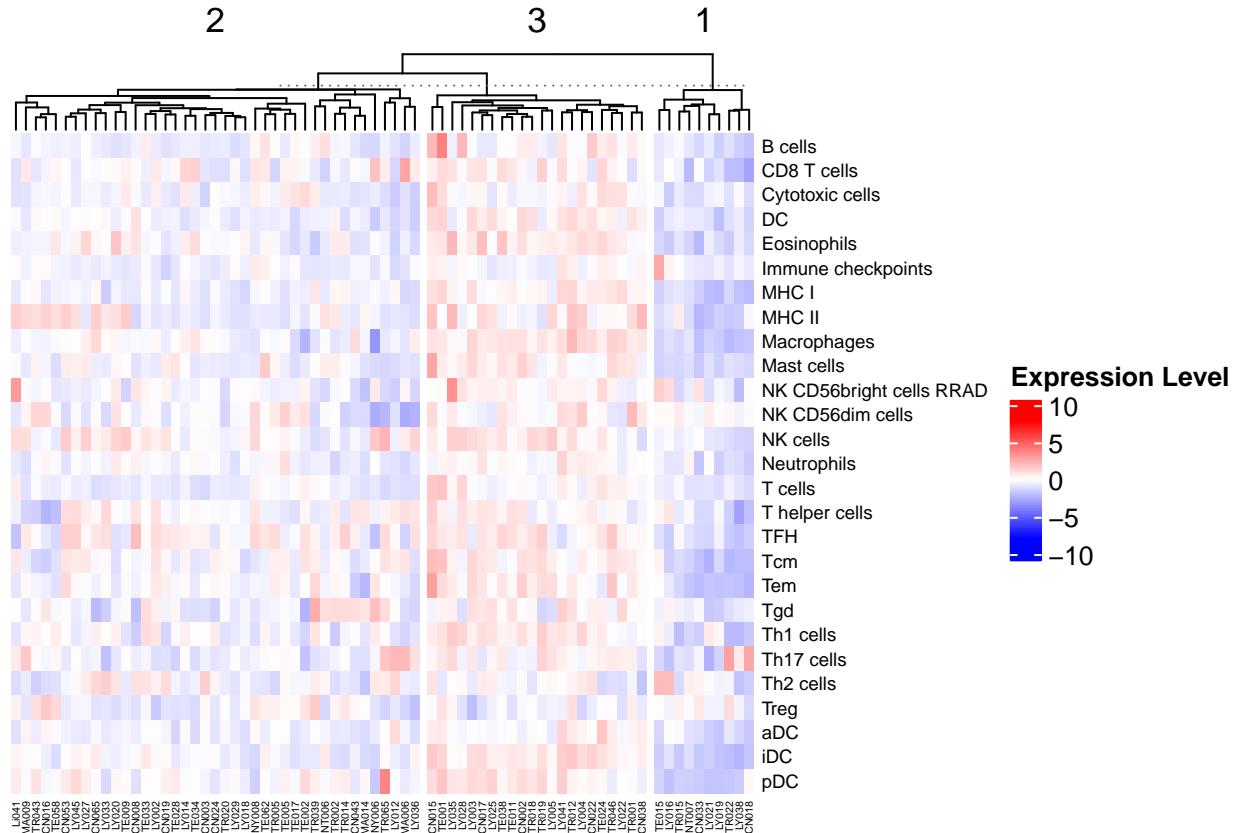


Figure 21: Heatmap with k-means clustering (k = 3) of immune cells expression (+ MHC) (n = 82)

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

between each clusters

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_expr_conv, 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, FCCcutoff = 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")
```

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, FCCcutoff = 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")
```

Volcano plot with all genes

C1 vs C2

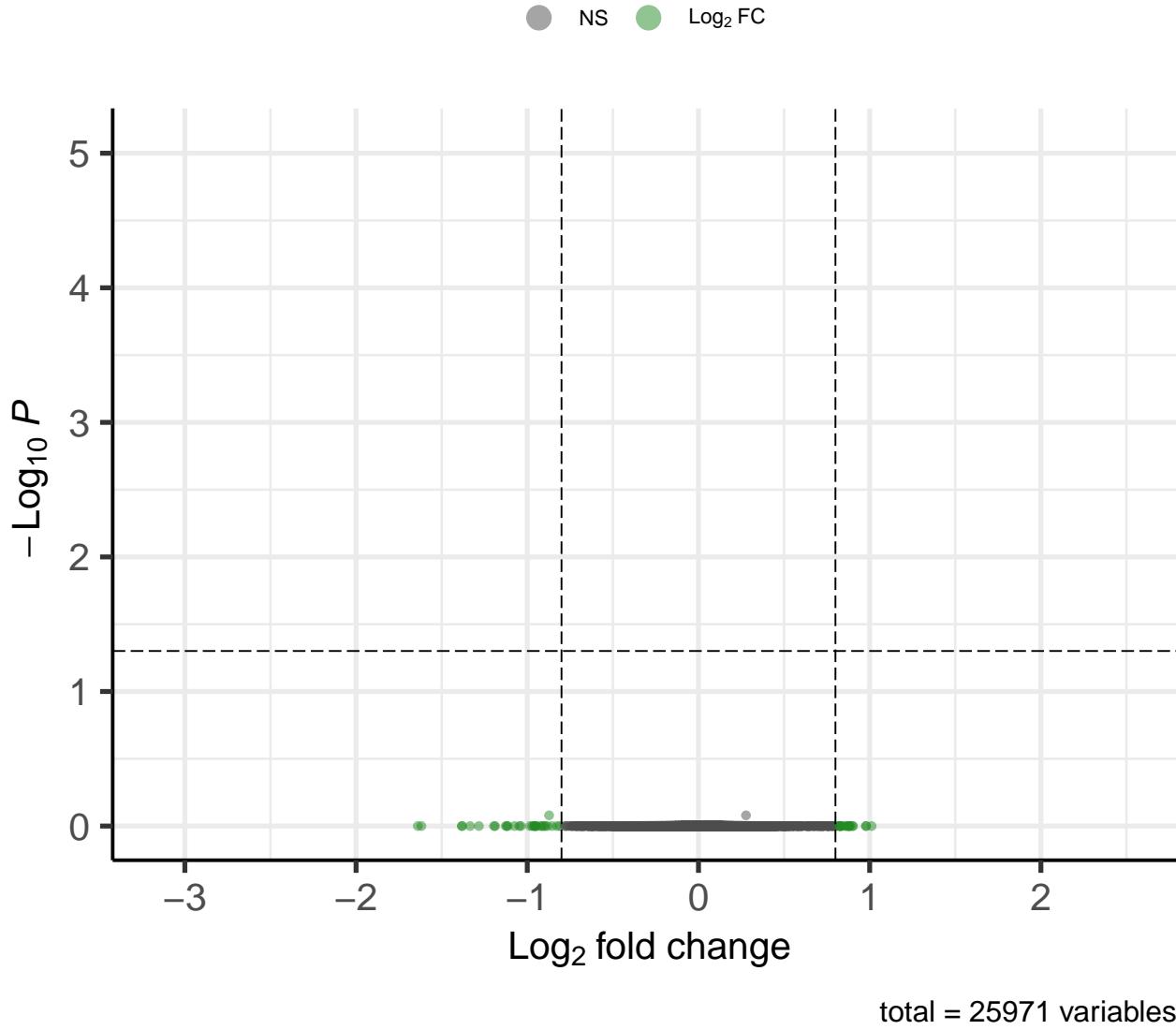


Figure 22: Volcano plot of DEG between C1 and C2 (from fig. 21)

Volcano plot with all genes

C2 vs C3

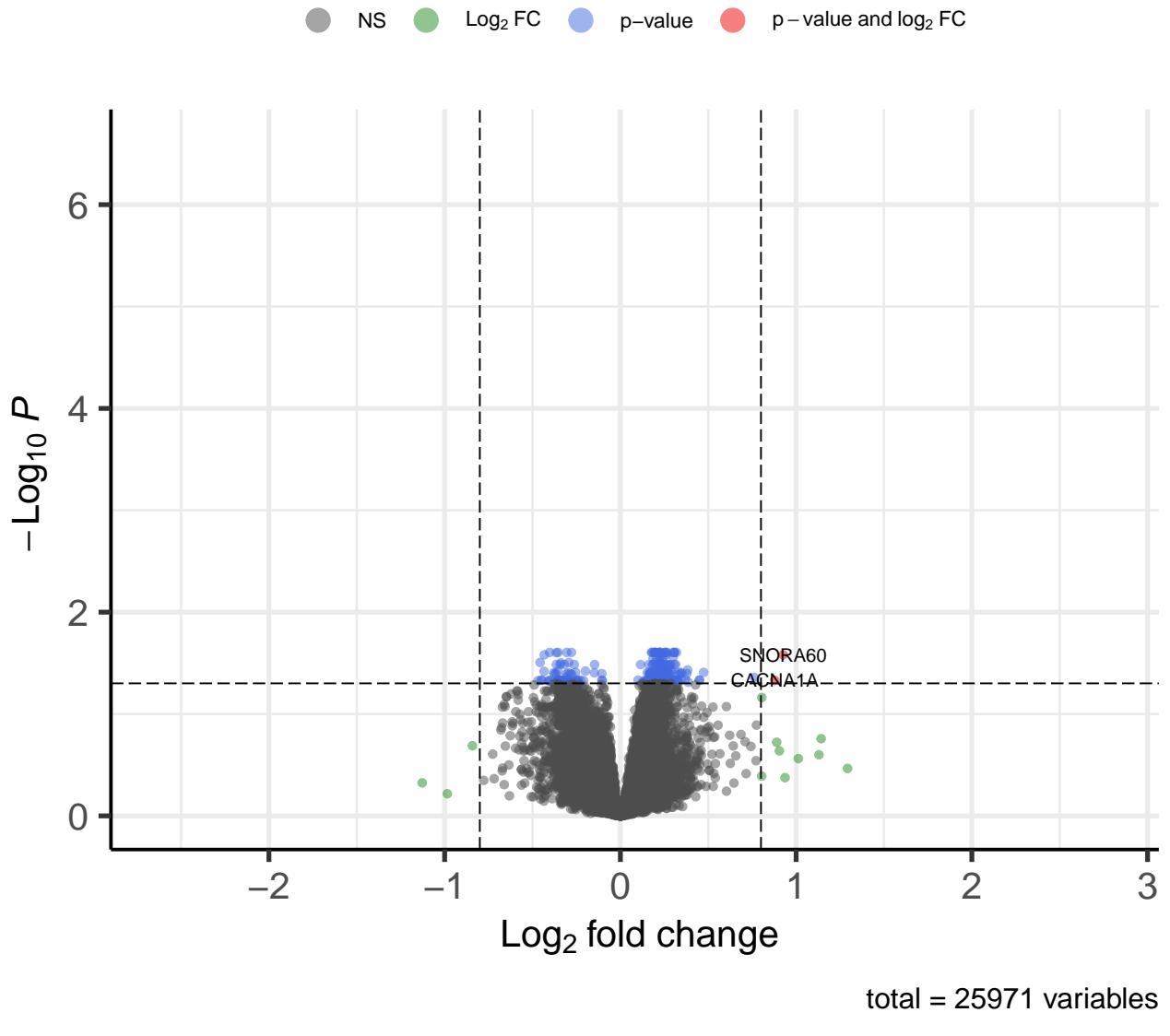


Figure 23: Volcano plot of DEG between C2 and C3 (from fig. 21)

```

EnhancedVolcano(resultats$C1_vs_C3, lab = rownames(resultats$C1_vs_C3),
  pCutoff = 0.05, FCCcutoff = 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")

```

Volcano plot with all genes

C3 vs C1

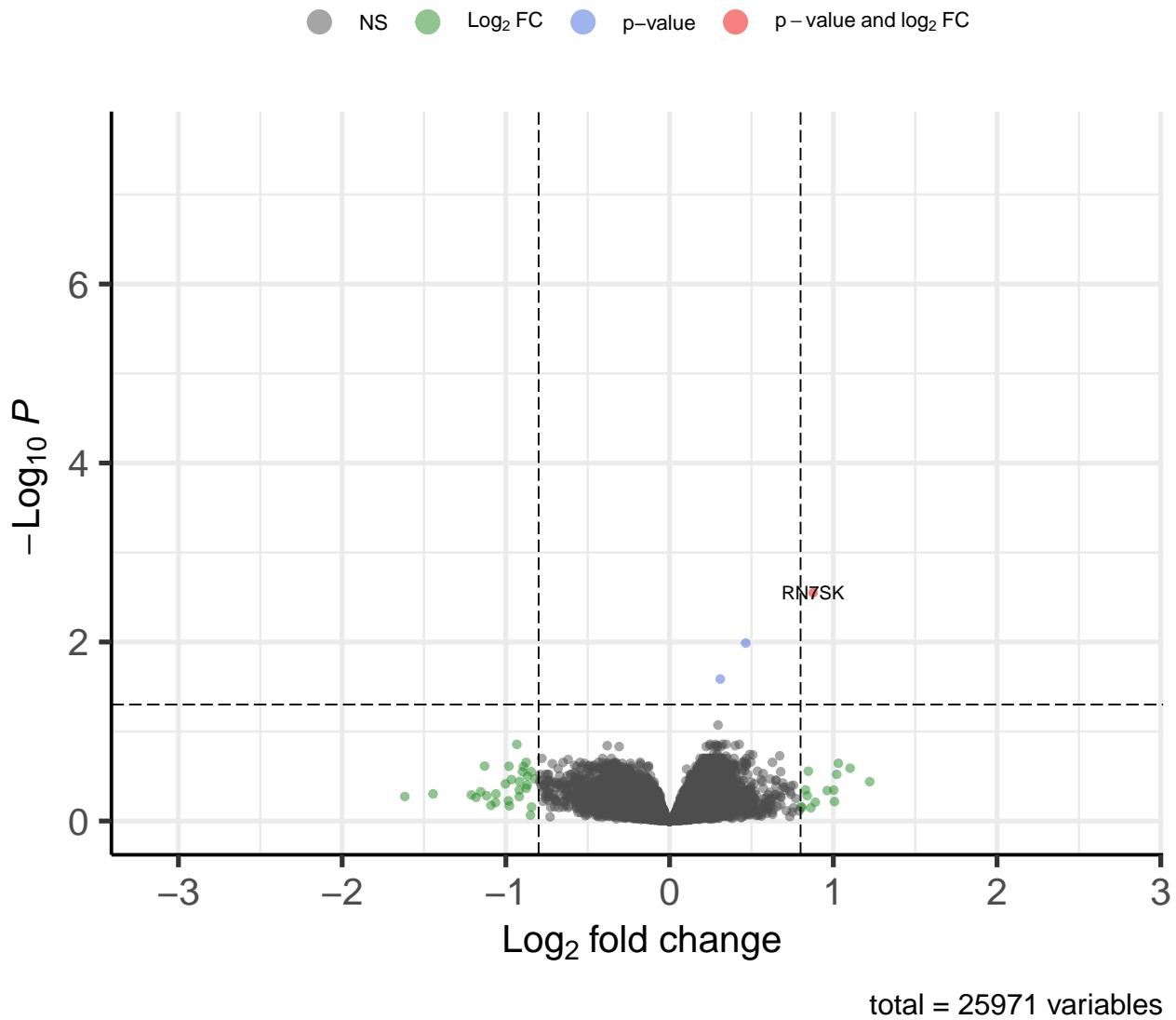


Figure 24: Volcano plot of DEG between C1 and C3 (from fig. 21)

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

```
set.seed(1)
Heatmap(as.matrix(heatmap_data_mhc_all_conv), 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"))
```

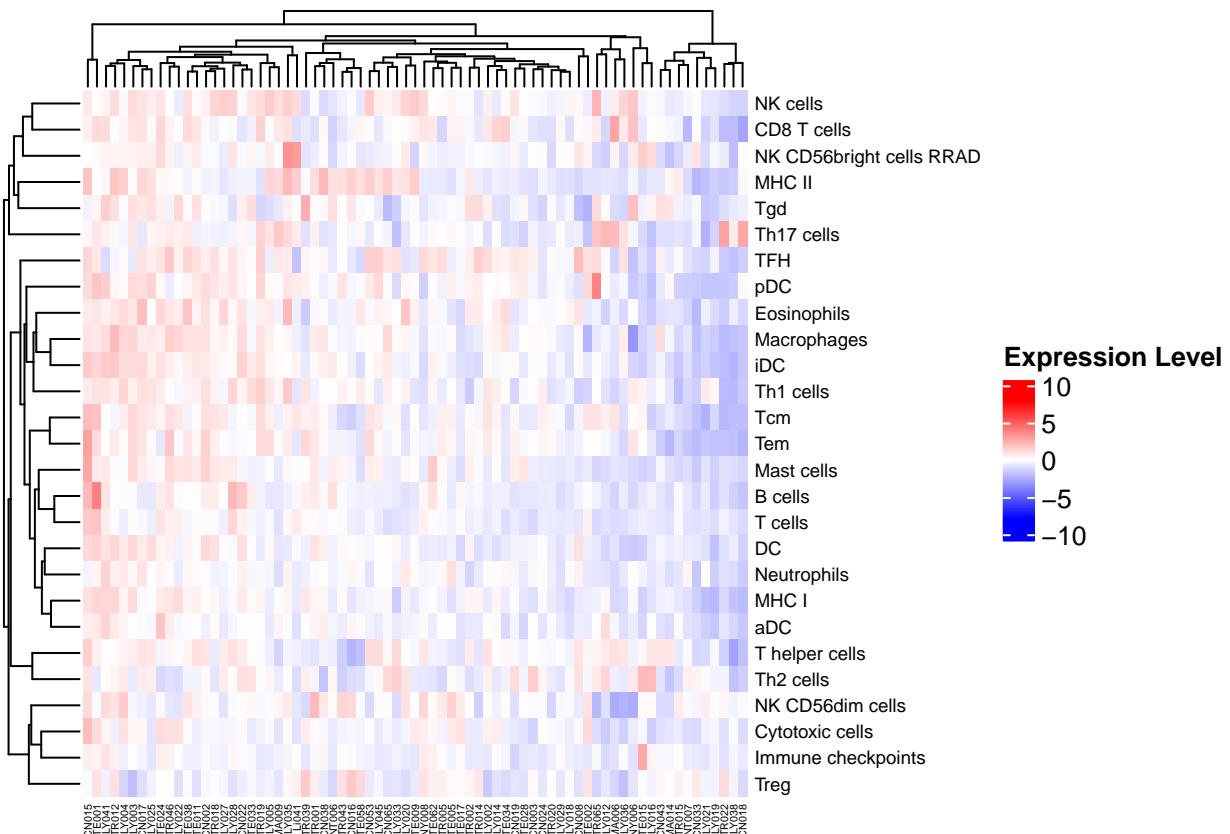


Figure 25: Heatmap and hierarchical clustering on relative immune cells expression (+ MHC) ($n = 82$)

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(heatmap_data_mhc_all_conv),
  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")
)

```

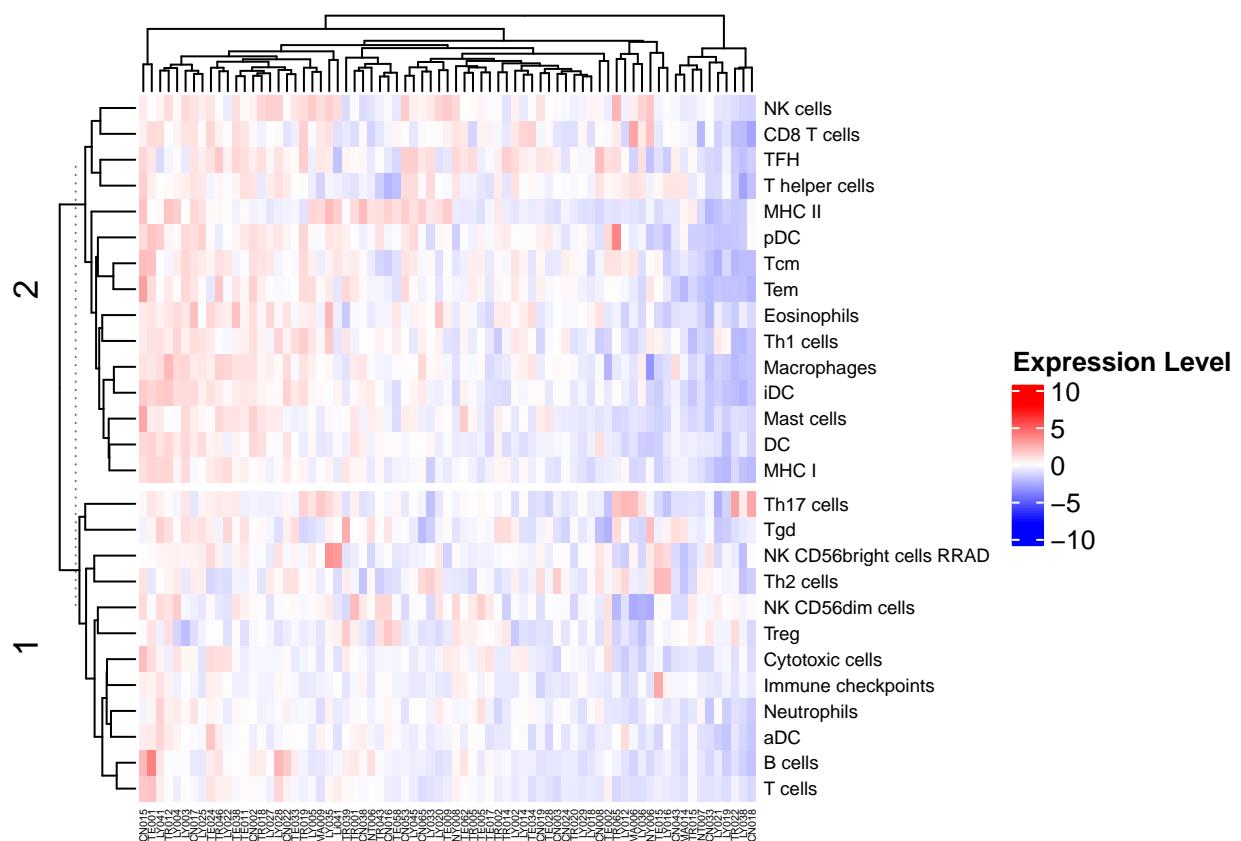


Figure 26: Heatmap with k-means clustering ($k = 2$) on relative immune cells expression (+ MHC) ($n = 82$)

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

c- Kmeans $k = 3$

```

set.seed(1)
Heatmap(
  as.matrix(heatmap_data_mhc_all_conv),
  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")
)

```

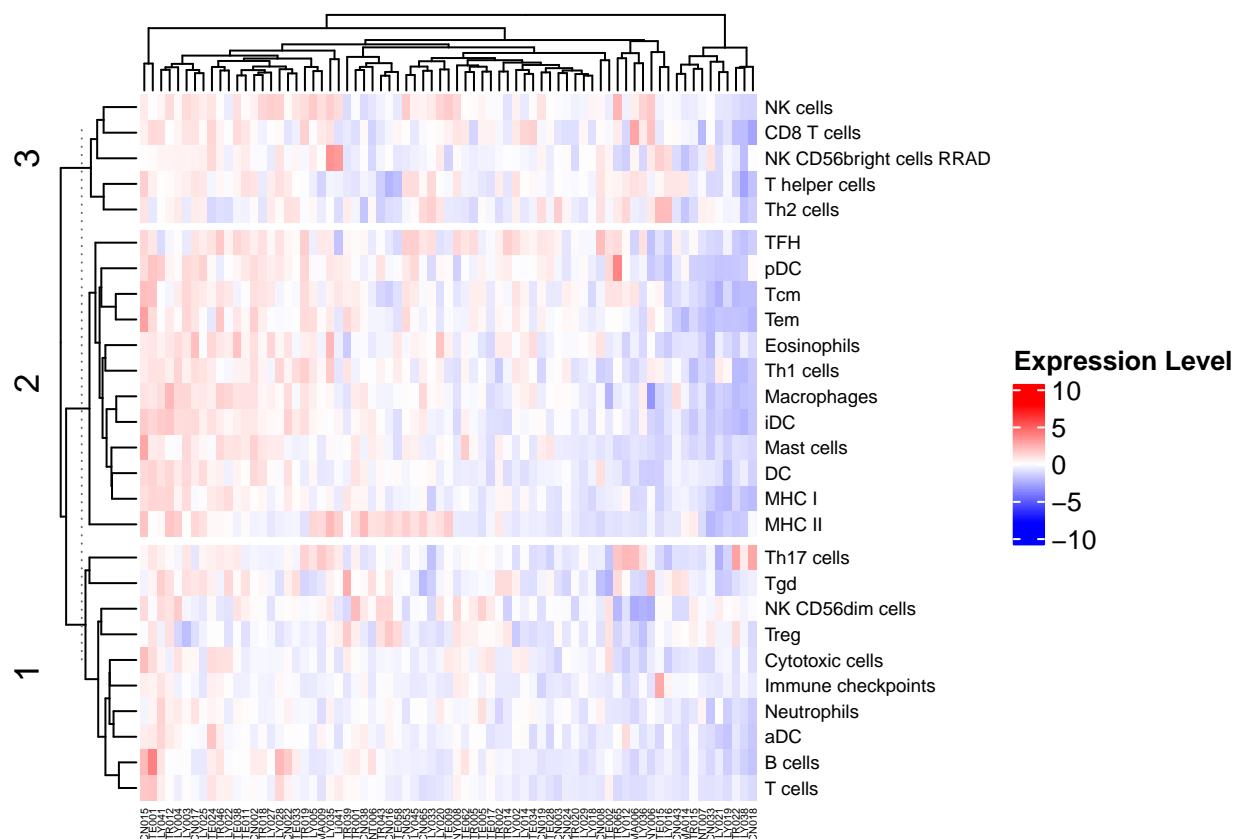


Figure 27: Heatmap with k-means clustering ($k = 3$) on relative immune cells expression (+ MHC) ($n = 82$)

Here, it is not very clear.

d- Kmeans $k = 4$

```

set.seed(1)
Heatmap(
  as.matrix(heatmap_data_mhc_all_conv),

```

```

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")
)

```

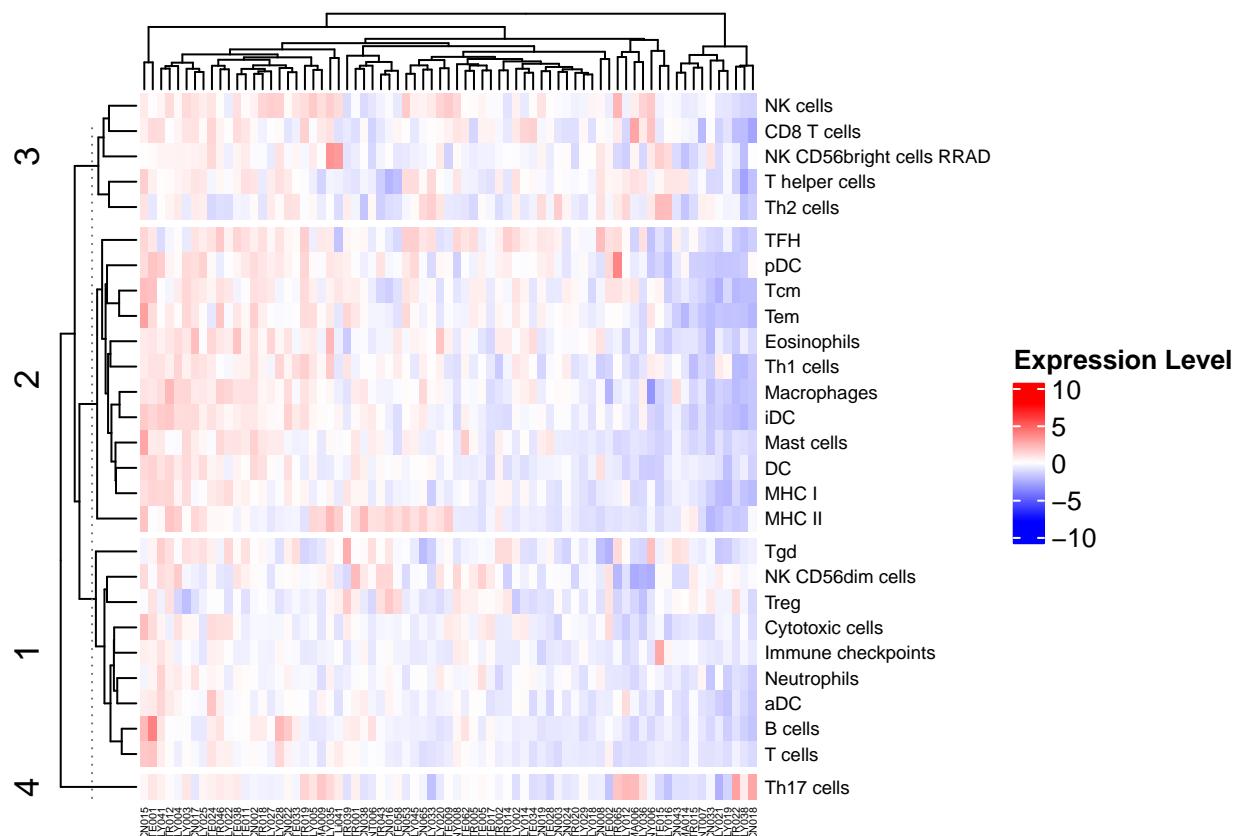


Figure 28: Heatmap with k-means clustering ($k = 4$) on relative immune cells expression (+ MHC) ($n = 82$)

3) Hierarchical clustering

```

set.seed(1)
Heatmap(as.matrix(heatmap_data_mhc_all_conv), 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"))

```

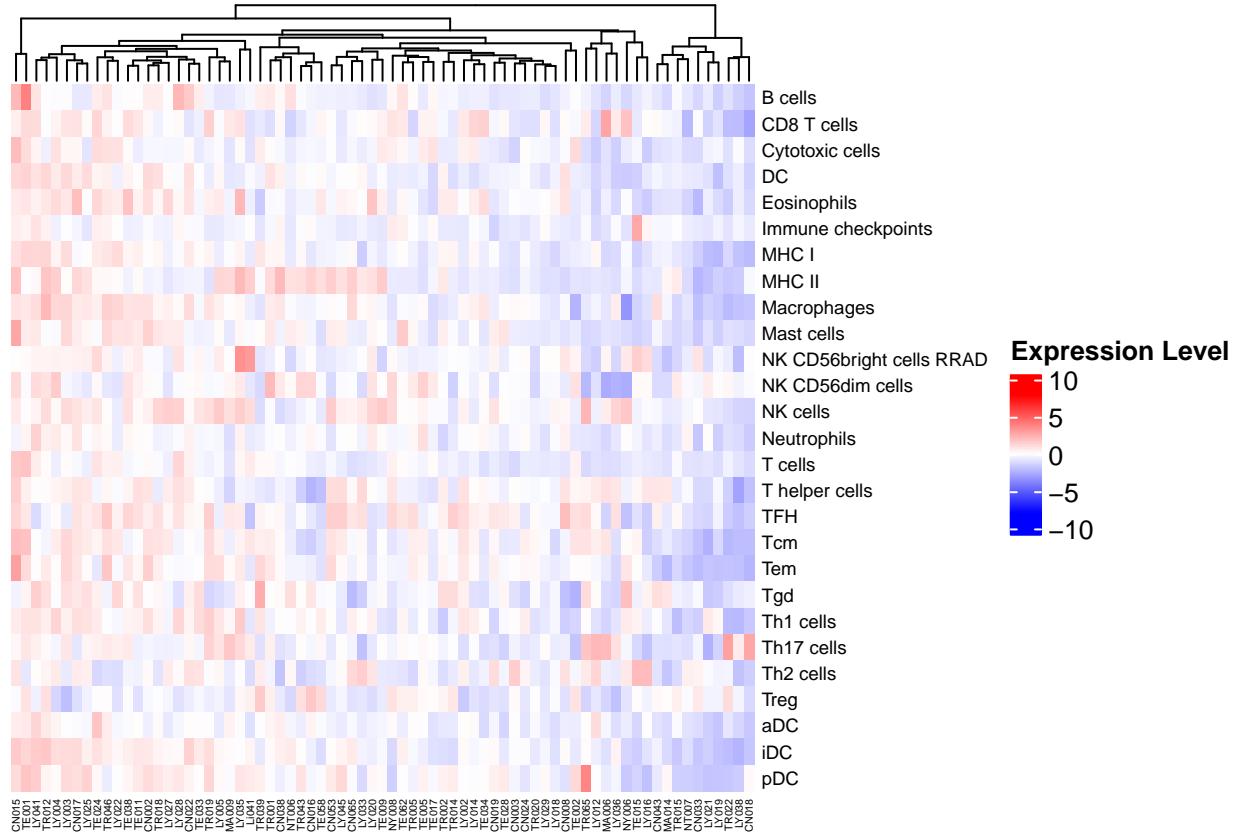


Figure 29: Heatmap and hierarchical clustering of relative immune cells expression (+ MHC) (n = 82)

This clustering shows 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 immunosuppressive cells to see “real” hot and cold tumors.

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 express 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", "Immune Checkpoint", "Regulatory T cell", "Mast cell", "Neutrophil", "Plasmacytoid dendritic cell", "T helper 17 cell", "T regulatory cell", "Tumour-associated macrophage", "Vascular endothelial cell")], row.names(heatmap_data_wo_immunosupp) = NULL

# 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")
)
```

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 Checkpoint", "Regulatory T cell", "Mast cell", "Neutrophil", "Plasmacytoid dendritic cell", "T helper 17 cell", "T regulatory cell", "Tumour-associated macrophage", "Vascular endothelial cell")], row.names(heatmap_data_immunosupp) = NULL

# Heatmap
set.seed(1)
Heatmap(
  as.matrix(heatmap_data_immunosupp),
  cluster_rows = FALSE,
  cluster_columns = TRUE,
  cluster_column_slices = TRUE,
  clustering_distance_columns = "euclidean",
```

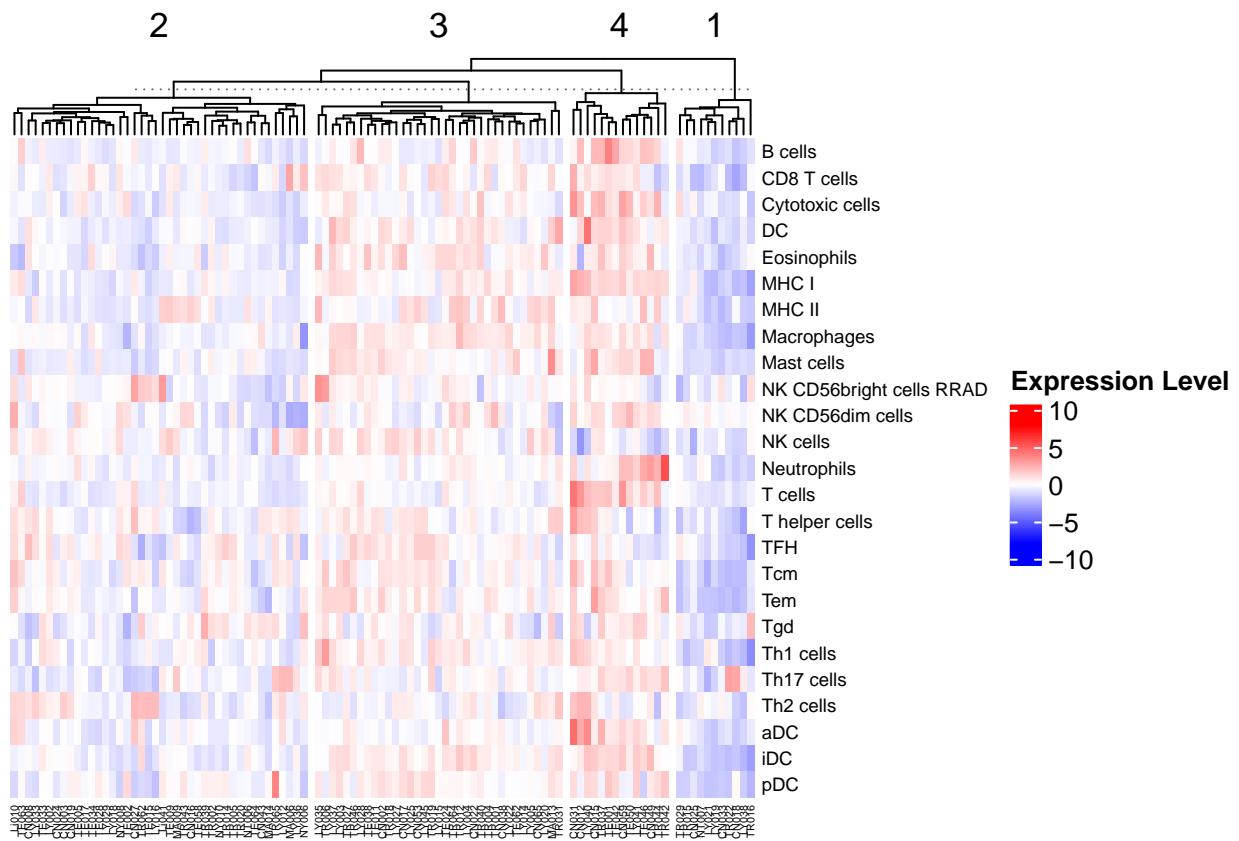


Figure 30: Heatmap of relative immune cells expression without immunosuppressive cells (Treg and Immune checkpoints) (n = 102)

```

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")
)

```

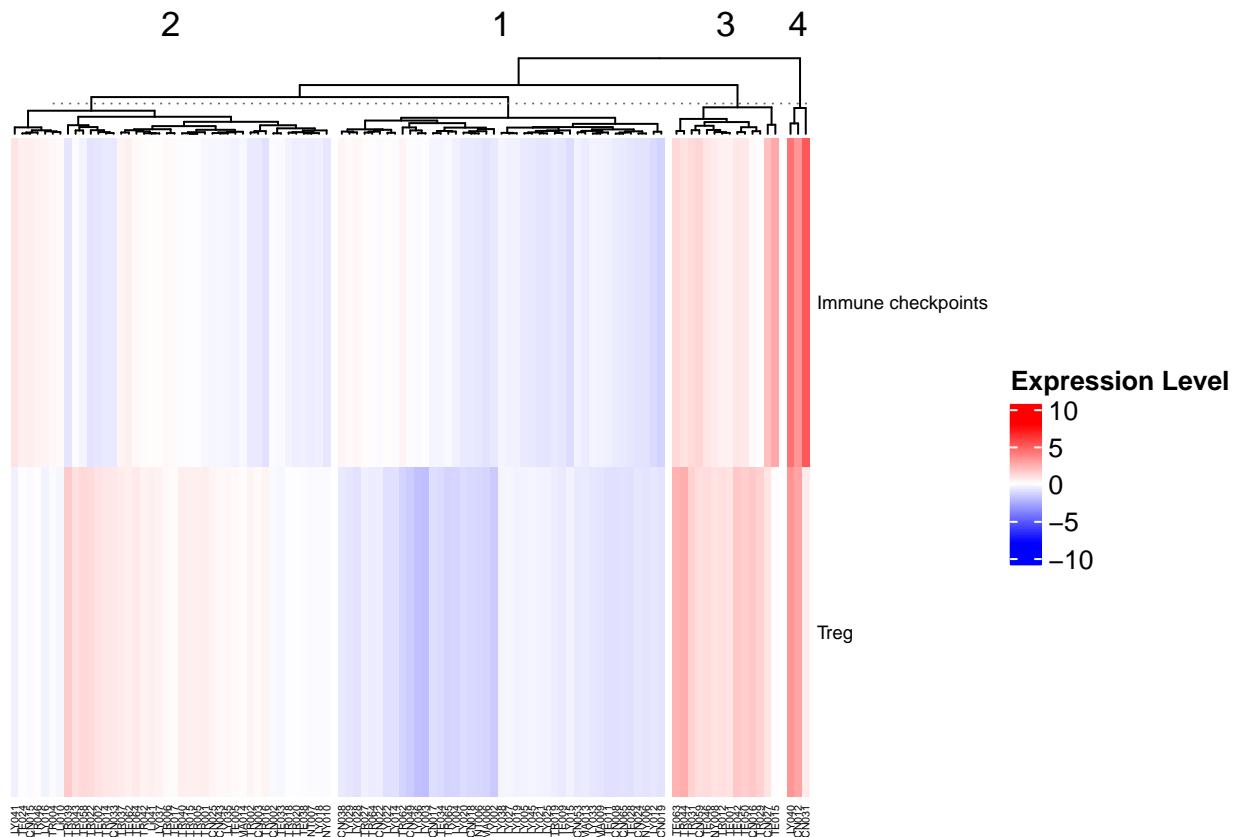


Figure 31: Heatmap of Treg and Immune checkpoints expression

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).

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
#pdf("../results/figures/heatmaps/heatmap_complete_annotated.pdf", height = 10, width = 15)
set.seed(1)
heatmap_anno_all <- Heatmap(
  as.matrix(heatmap_data_mhc_all),
  cluster_rows = FALSE,
  cluster_columns = TRUE,
  cluster_column_slices = TRUE,
  show_column_dend = FALSE,
  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)

```

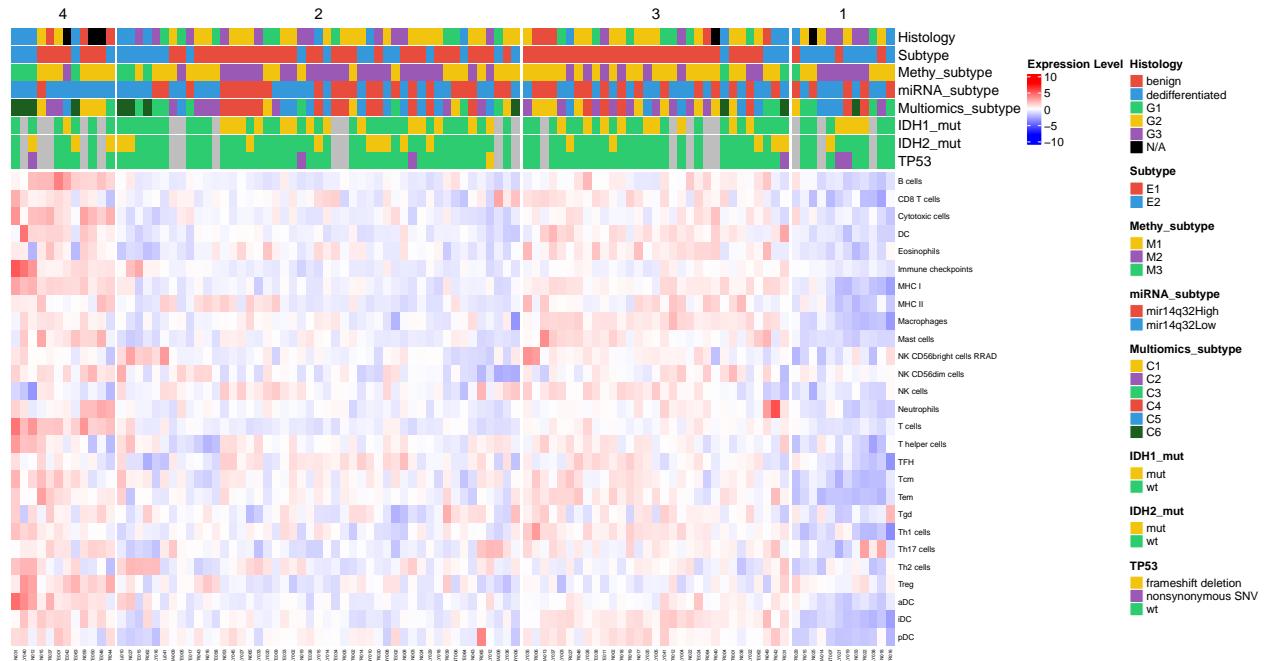


Figure 32: Heatmap (from fig. 14) with metadata (n = 102)

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

```

# 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", ...

```

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 know that dediff are generally bad diagnosis but here we see that they are infiltrated. So this could be induce errors.

2) Adding metadata on heatmap with conventional patients

a- $k = 2$

```

# Heatmap
#pdf("../results/figures/heatmaps/heatmap_complete_annotated_conv_patients.pdf", height = 10, width = 10)
set.seed(1)
heatmap_anno_conv <- Heatmap(
  as.matrix(heatmap_data_mhc_all_conv),
  cluster_rows = FALSE,
  cluster_columns = TRUE,
  cluster_column_slices = TRUE,
  show_column_dend = FALSE,
  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,
           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)

```

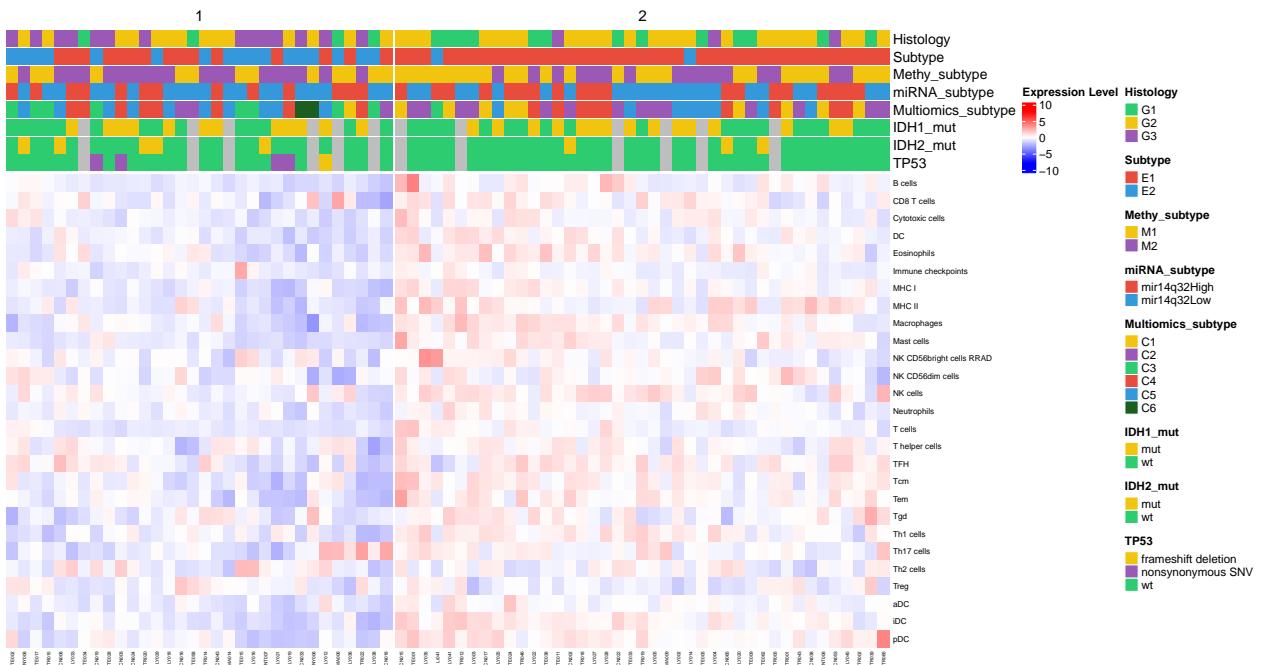


Figure 33: Heatmap (from fig. 17) with metadata (n = 82)

```

#dev.off()

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

# 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`))),
  Patient = c(colnames(heatmap_data_mhc_all_conv)[indiv_clust$`1`],
              colnames(heatmap_data_mhc_all_conv)[indiv_clust$`2`])
)

```

```
# Save
#write.table(df_indiv_clusters_hm_anno, file = "../results/clusters_conv_indiv_mhc.tsv", sep = "\t", qu
```

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

b- k = 3

```
# Heatmap
#pdf("../results/figures/heatmaps/heatmap_complete_annotated_conv_patients.pdf", height = 10, width = 10)
set.seed(1)
heatmap_anno_conv <- Heatmap(
  as.matrix(heatmap_data_mhc_all_conv),
  cluster_rows = FALSE,
  cluster_columns = TRUE,
  cluster_column_slices = TRUE,
  show_column_dend = FALSE,
  column_km = 3, # Number of clusters
  column_km_repeats = 20,
  col = colorRamp2(seq(-8, 8, length.out = 100), colors),
  border = NA,
  show_column_names = TRUE,
  column_names_gp = gpar(fontsize = 4),
  row_names_gp = gpar(fontsize = 7),
  heatmap_legend_param = list(title = "Expression Level"),
  top_annotation = columnAnnotation(
    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)

#dev.off()
```

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.

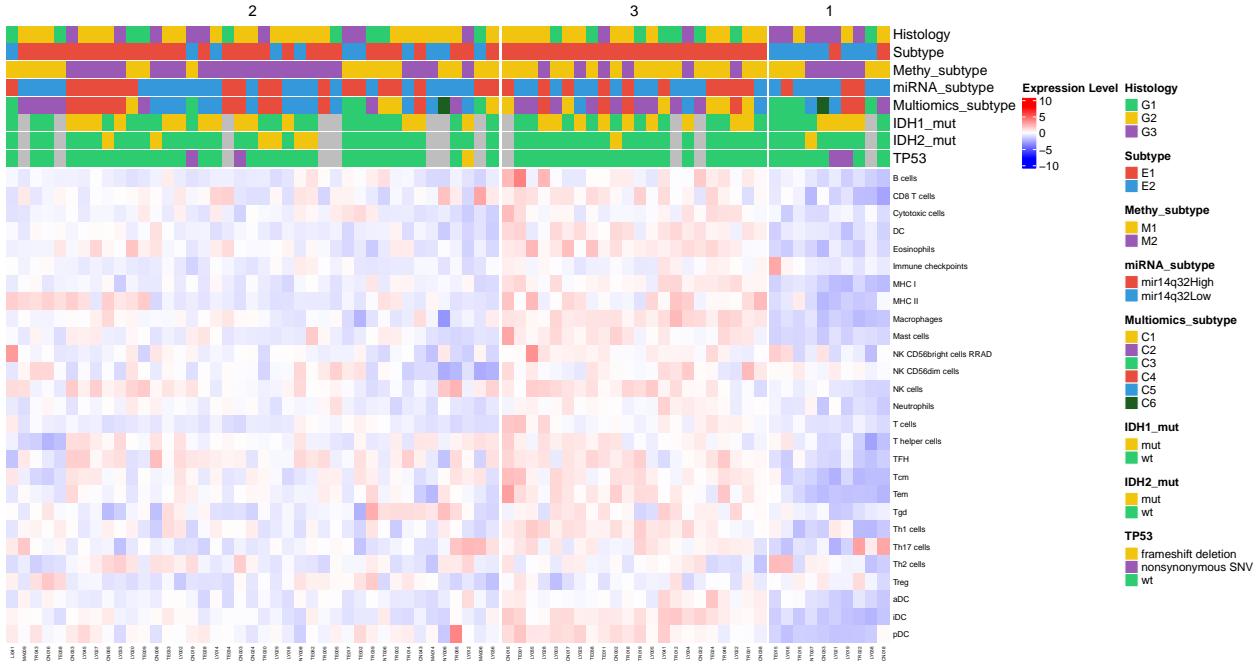


Figure 34: Heatmap (from fig. 21) with metadata ($n = 82$)

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

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))
```

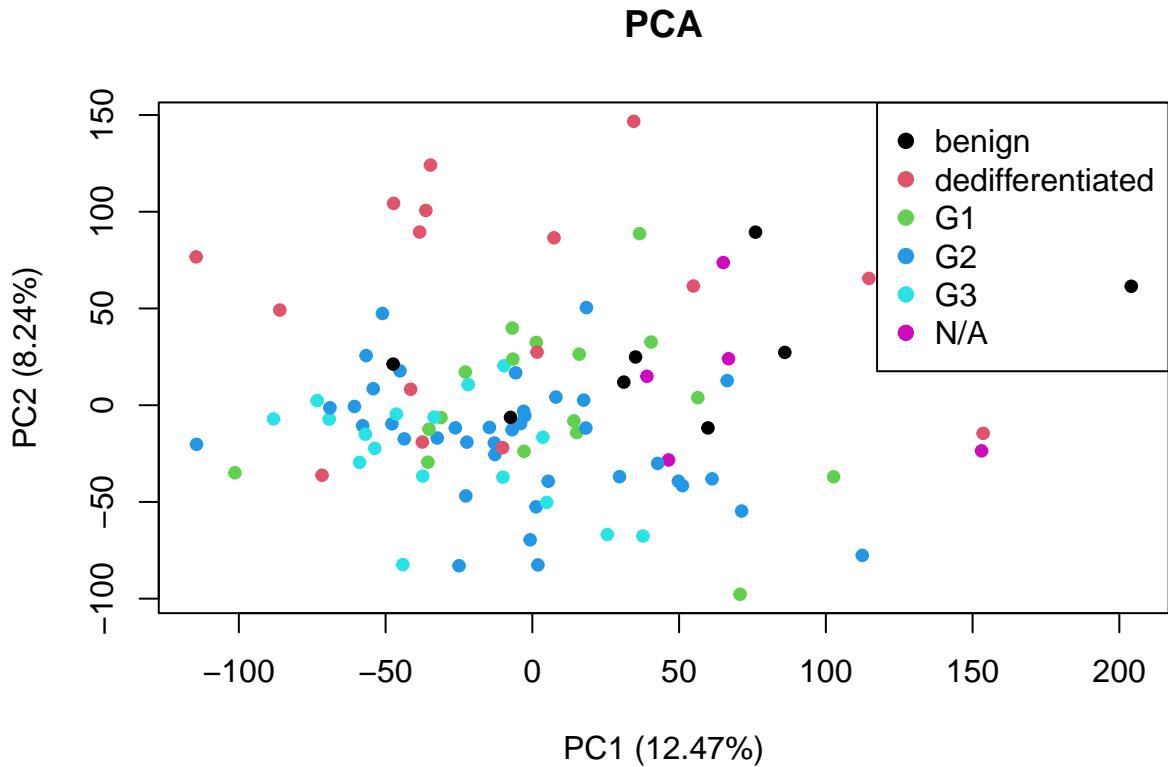


Figure 35: PCA plot with histology type ($n = 102$)

```

# Fit the linear model
data_fit <- lmFit(df_CTA_immune_whole_clean_avg[, -c(1:3)], design)

# Define contrasts
contrast_matrix <- makeContrasts(benign - malignant, 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 = "benign vs malignant")

# 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")

```

Volcano plot with all genes

benign vs malignant

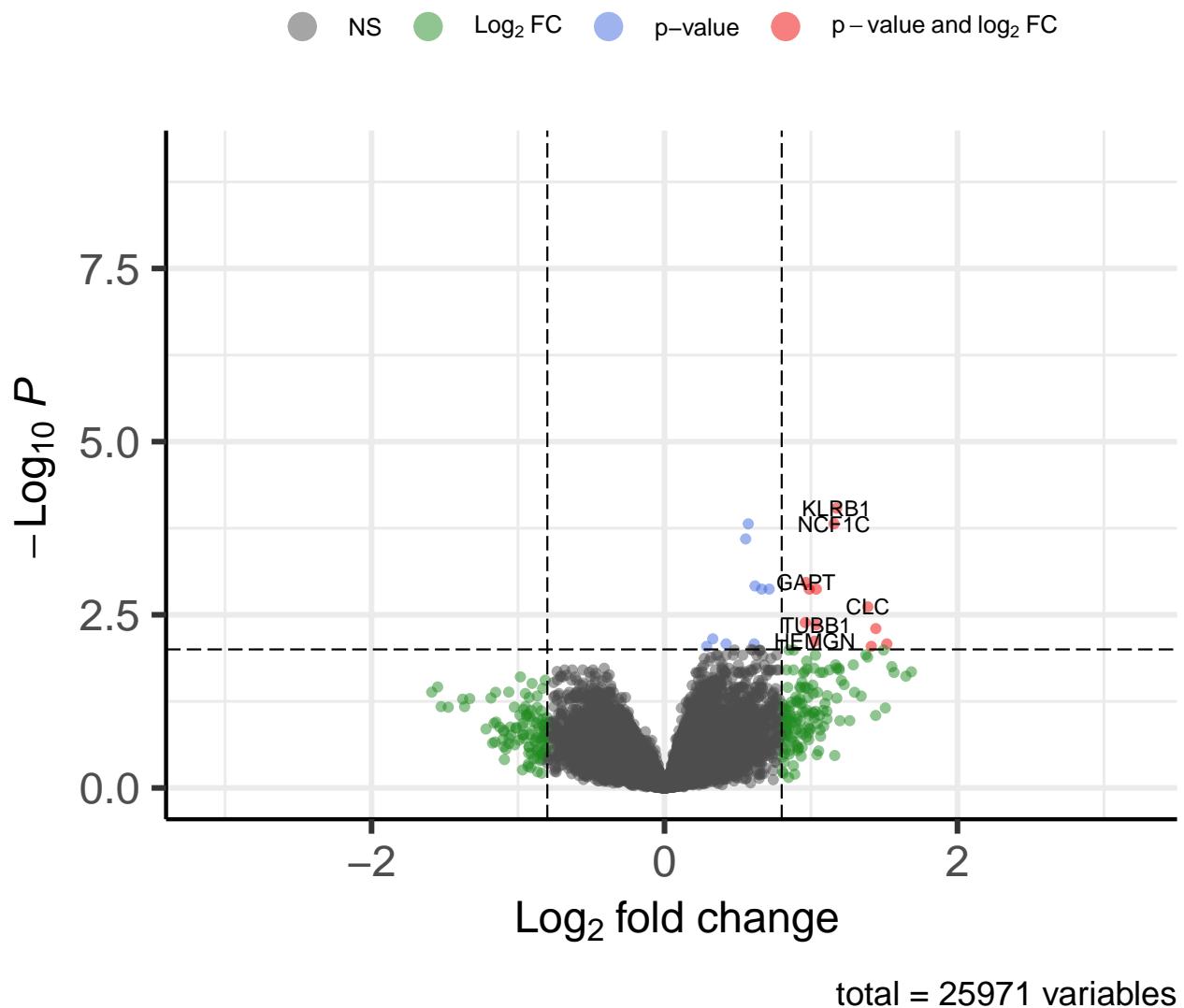


Figure 36: Volcano plot of DEG between benign and malignant tumors

Volcano plot with CTA genes

benign vs malignant

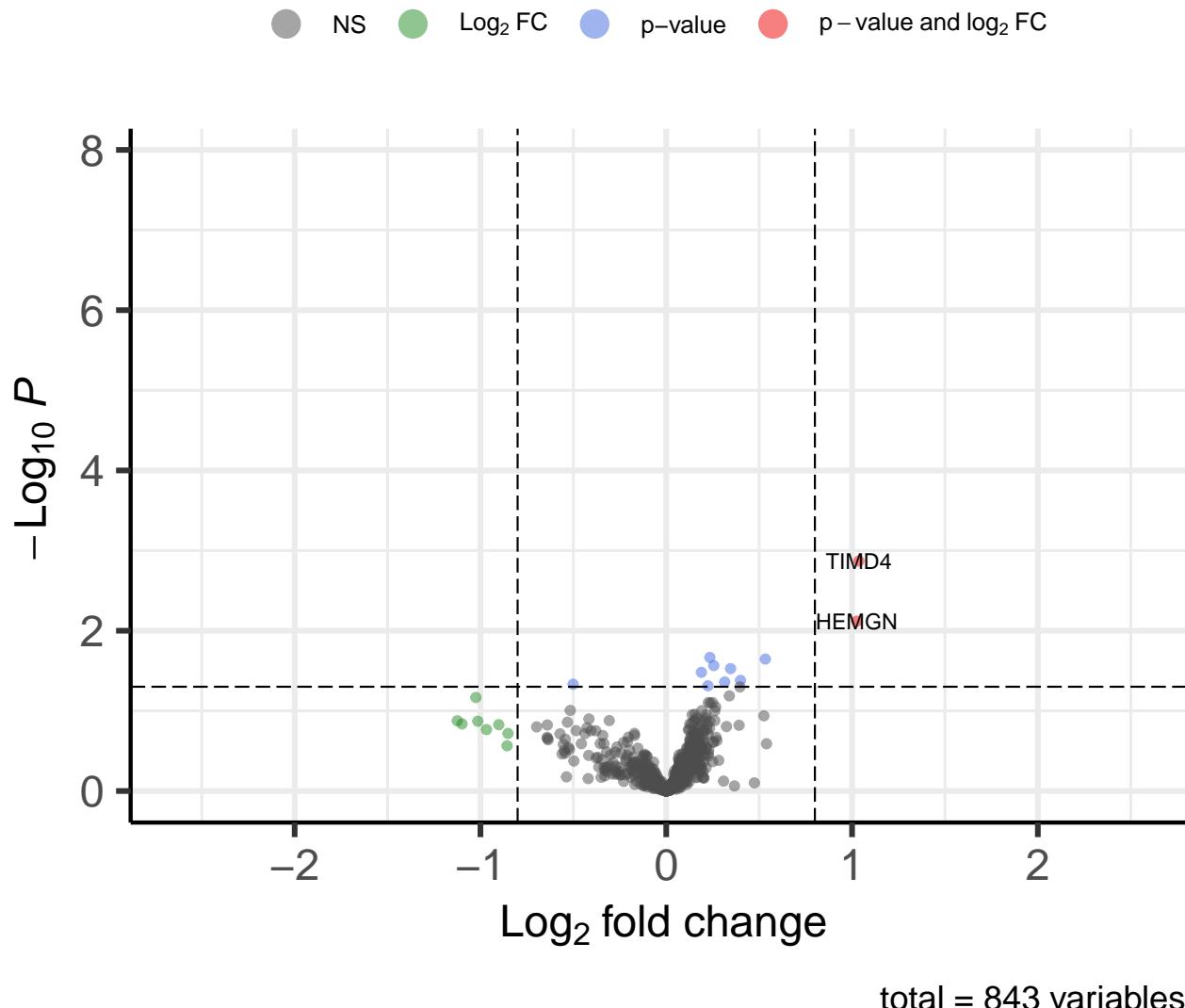


Figure 37: Volcano plot of differentially expressed CTA between benign and malignant tumors

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[, -c(1:3)], 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")

# 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")
```

Volcano plot with all genes

benign vs dediff

● NS ● Log₂ FC ● p-value ● p-value and log₂ FC

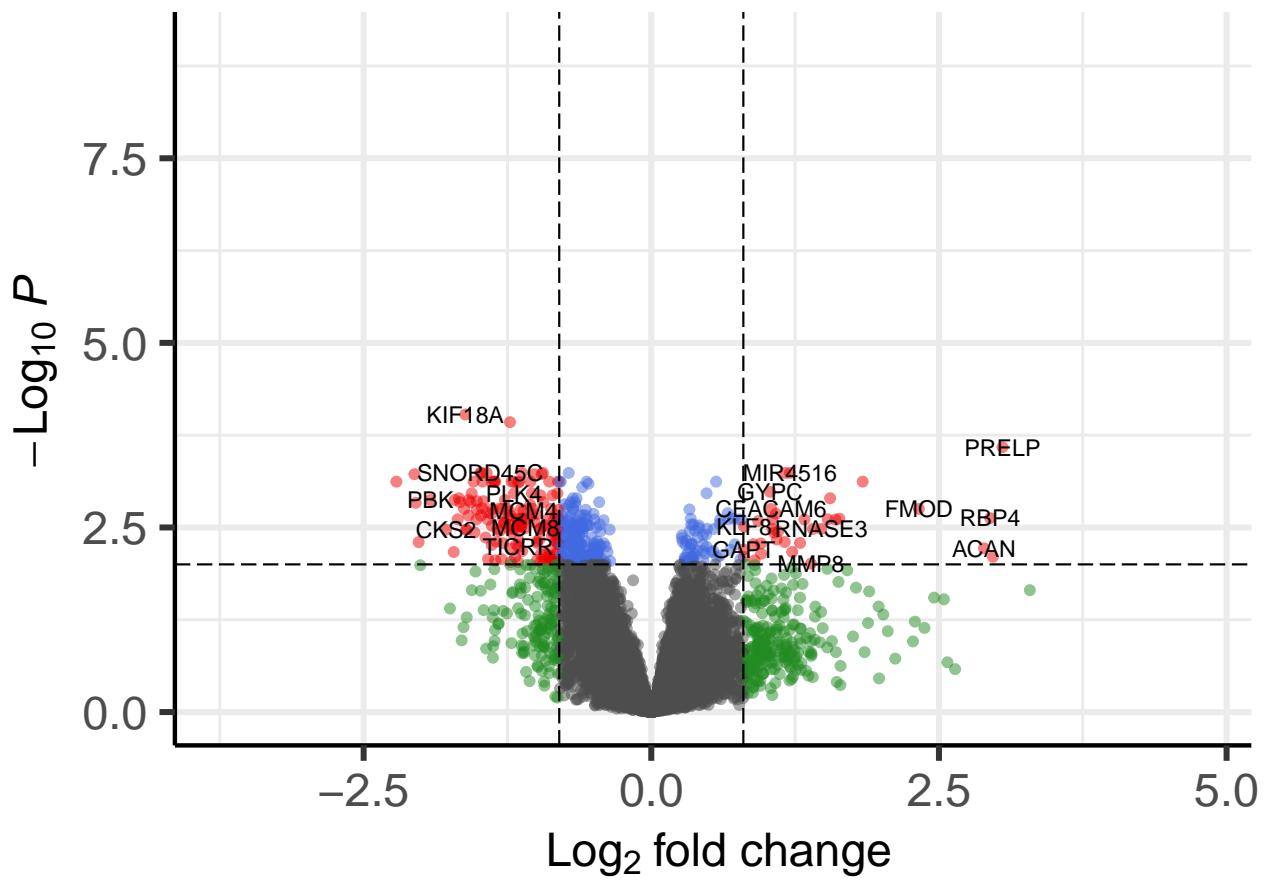
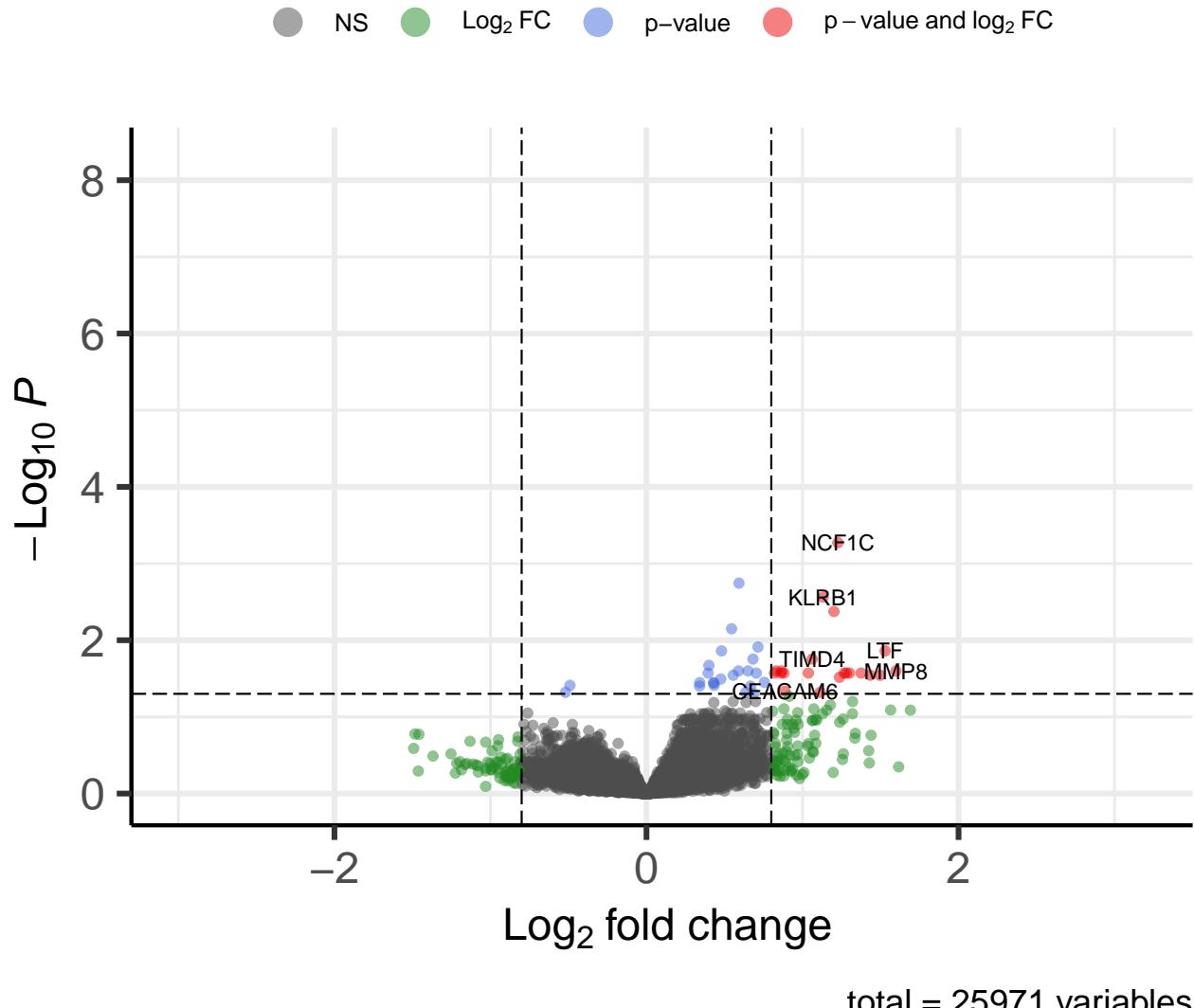


Figure 38: Volcano plot of DEG between benign and malignant tumors

Volcano plot with all genes

benign vs G1



```
# 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")
```

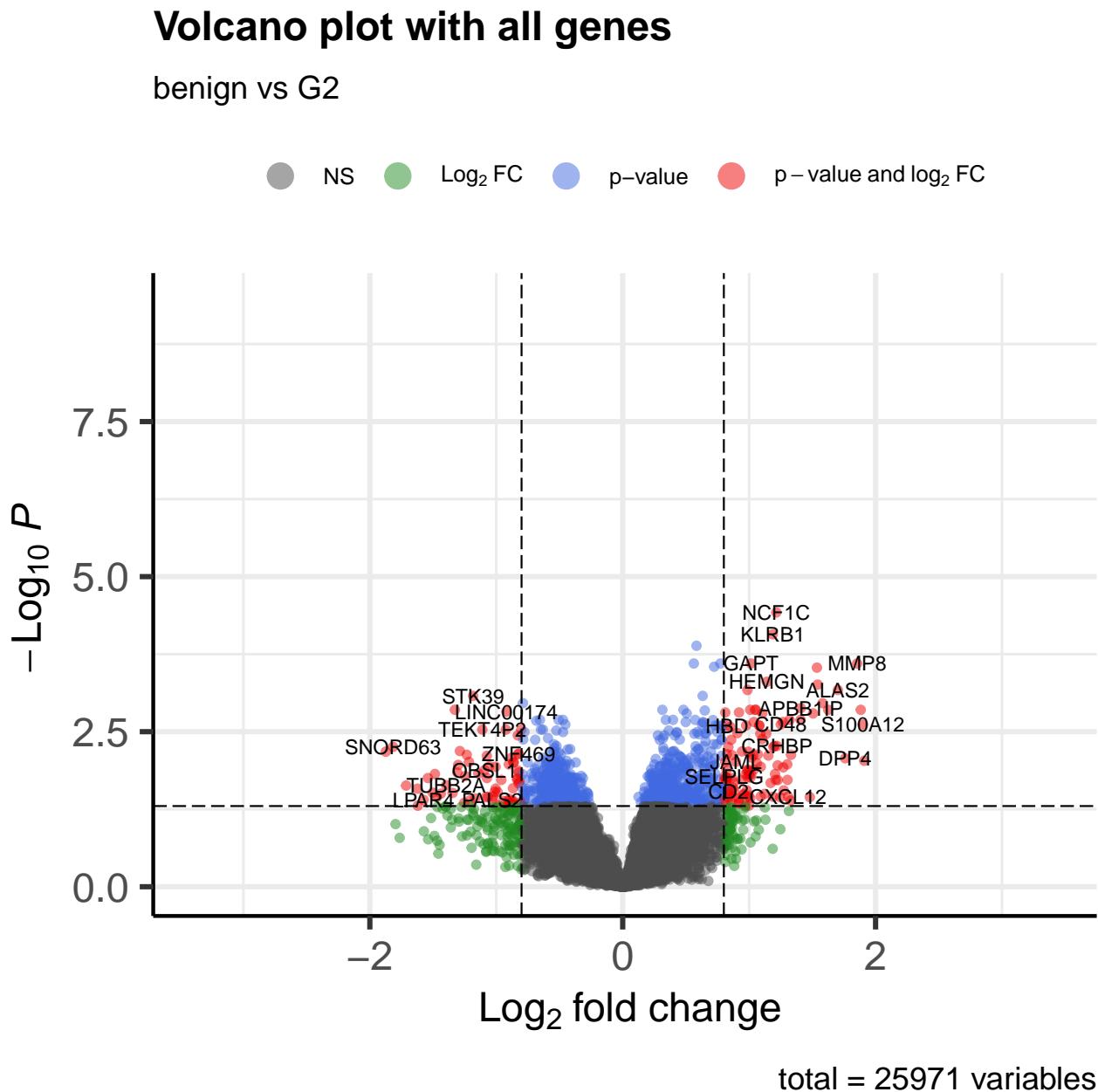


Figure 40: Volcano plot of DEG between benign and G2

```
# 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")
```

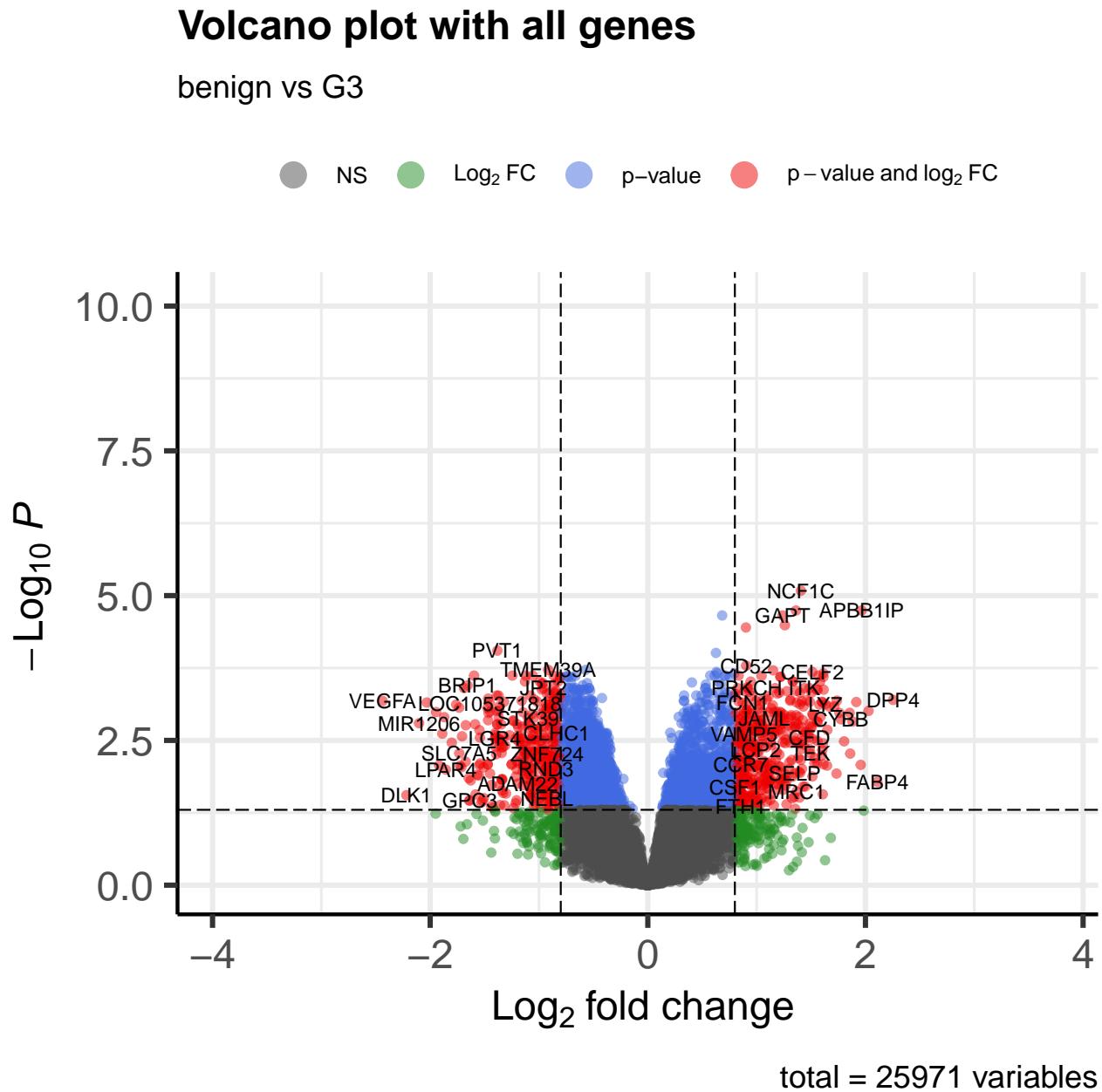


Figure 41: Volcano plot of DEG between benign and G3

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

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

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

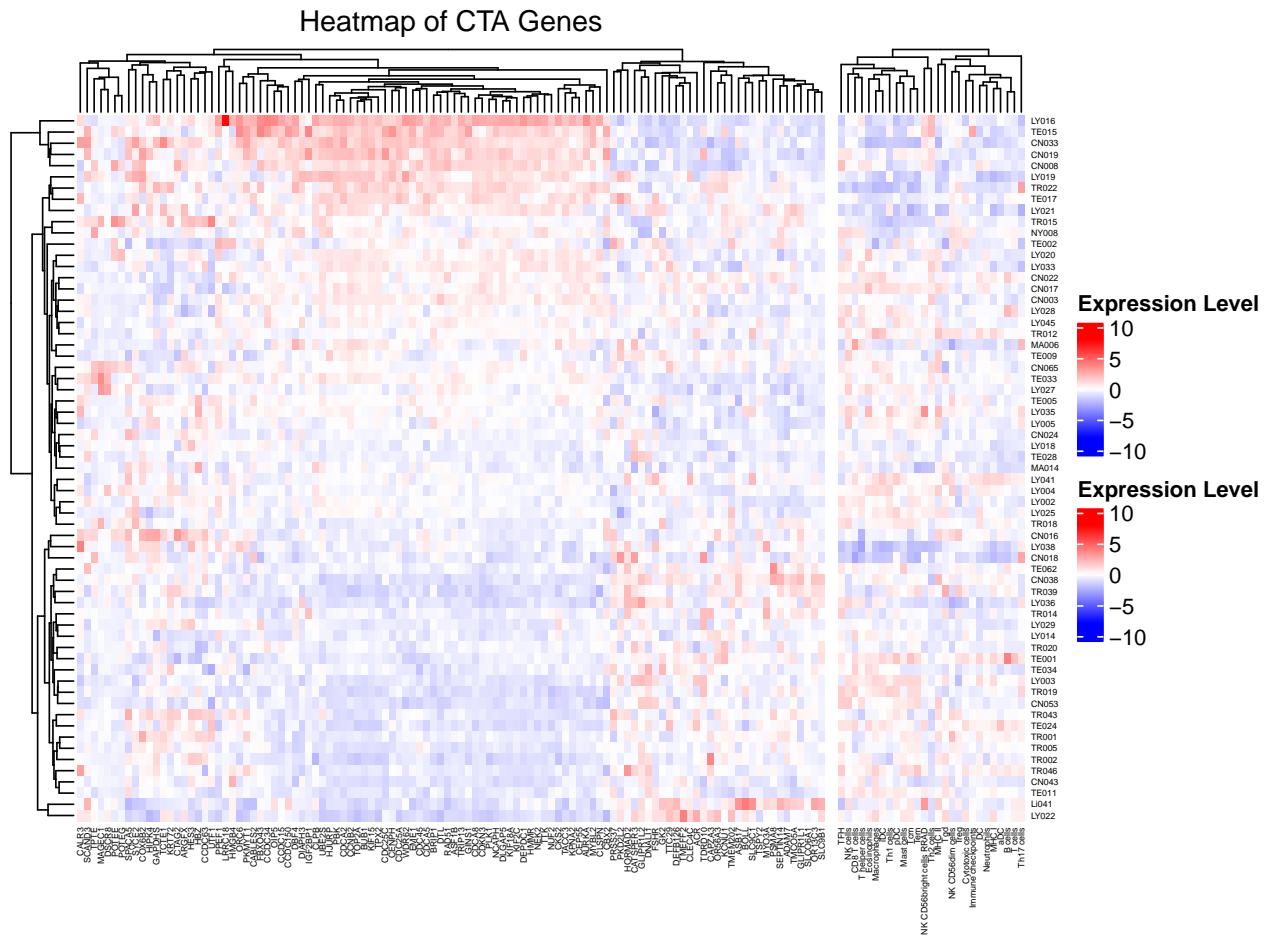


Figure 42: Heatmap of CTA that impact survival analysis and immune cells expression (n = 63)

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(heatmap_data_mhc_all_conv[, colnames(heatmap_data_mhc_all_conv) %in% df_metadata_surv
    ]), 2)

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

# kmeans heatmap
ht_k2_conv <- Heatmap(
    as.matrix(t(heatmap_data_mhc_all_conv[, colnames(heatmap_data_mhc_all_conv) %in% df_metadata_surv
        ])),
    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

```

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.

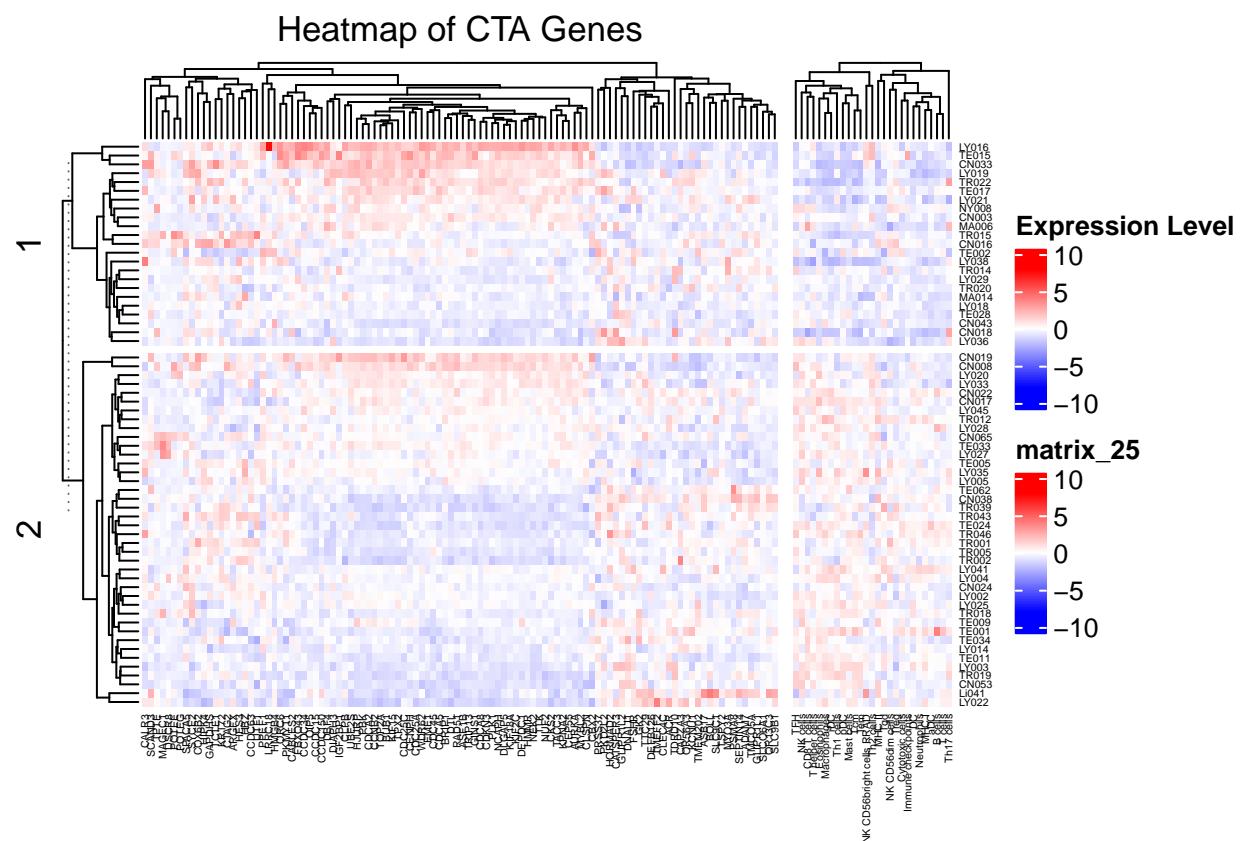


Figure 43: Heatmap with kmeans clustering of CTA that impact survival analysis and immune cells expression ($n = 63$)

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(subset(df_CTA_immune_whole_clean_z_scores[, colnames(df_CTA_immune_whole_clean_z_scores) %in% df_metadata_surv_conv$Patient], rownames(df_CTA_immune_whole_clean_z_scores) %in% l_CTA_conv)))

# Immune cells data
rownames(df_indiv_clusters_hm_anno) <- df_indiv_clusters_hm_anno$Patient
df_imm_data <- as.data.frame(t(heatmap_data_mhc_all_conv[, colnames(heatmap_data_mhc_all_conv) %in% df_metadata_surv_conv$Patient]))

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

# 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 between 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 = "average")

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

1) 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)

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

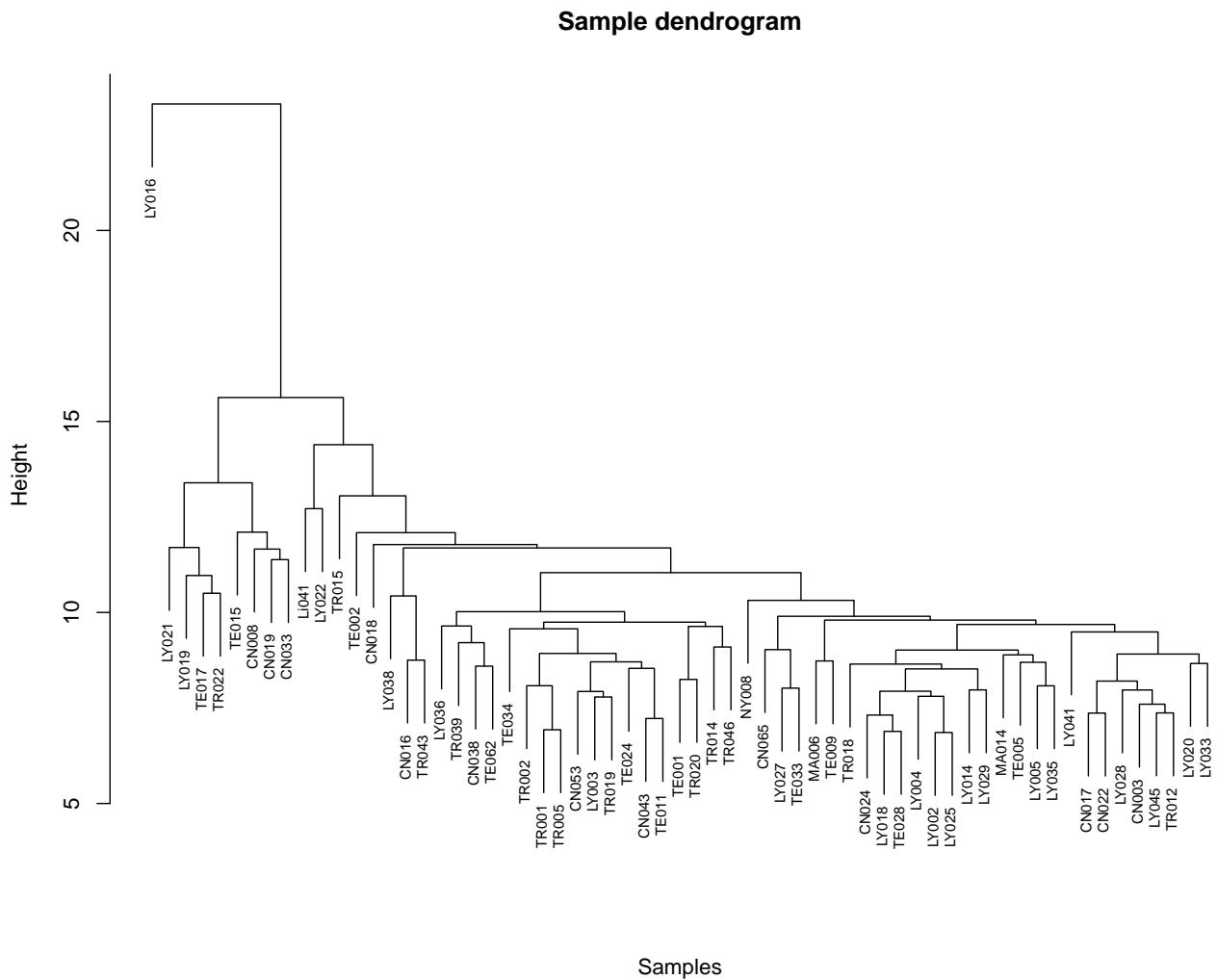


Figure 44: Dendrogram of conventional patients (from fig. 3)

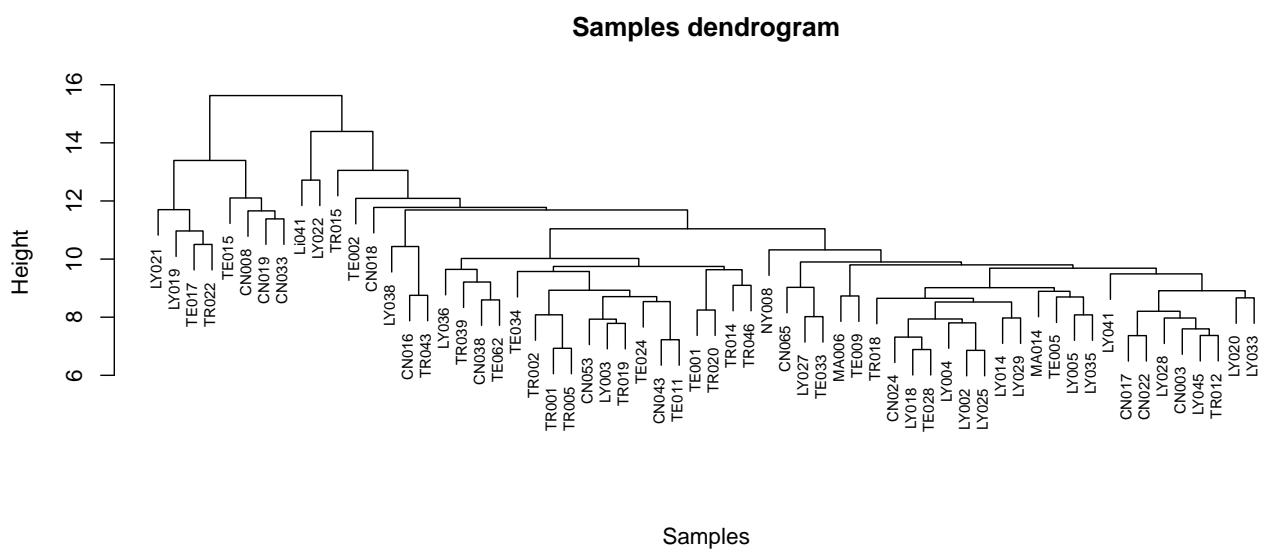


Figure 45: Corrected dendrogram of conventional patients (from fig. 3)

```

##      Power SFT.R.sq    slope truncated.R.sq mean.k. median.k. max.k.
## 1     0.000761  0.0257       0.0860   30.500  3.13e+01  49.60
## 2     0.218000 -0.4410       0.5100   14.600  1.16e+01  33.30
## 3     0.351000 -0.5070       0.6550   8.720  4.84e+00  25.20
## 4     0.450000 -0.5570       0.5560   5.810  2.15e+00  20.00
## 5     0.734000 -0.5990       0.7800   4.110  9.93e-01  16.30
## 6     0.773000 -0.6390       0.7450   3.030  4.73e-01  13.50
## 7     0.819000 -0.6670       0.7880   2.290  2.87e-01  11.30
## 8     0.105000 -1.3800      -0.1490   1.770  2.35e-01  9.51
## 9     0.917000 -0.7310       0.8940   1.380  1.31e-01  8.08
## 10    0.972000 -0.7330       0.9680   1.100  7.38e-02  6.91
## 11    0.810000 -0.7880       0.7560   0.885  4.18e-02  5.93
## 12    0.103000 -1.2800      -0.1440   0.720  2.38e-02  5.12
## 13    0.103000 -1.2800      -0.1440   0.720  2.38e-02  5.12
## 14    0.920000 -0.8750       0.8990   0.488  7.84e-03  3.85
## 15    0.191000 -1.7900       0.0637   0.341  2.70e-03  2.94
## 16    0.193000 -1.8600       0.0611   0.244  1.22e-03  2.26
## 17    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
plot(soft_threshold$fitIndices[, 1], -sign(soft_threshold$fitIndices[, 3]) * soft_threshold$fitIndices[, 2], xlab = "Soft Threshold (power)",
      ylab = "Scale Free Topology Model Fit,signed R^2", type = "n",
      main = paste("Scale independence"))
text(soft_threshold$fitIndices[, 1], -sign(soft_threshold$fitIndices[, 3]) * soft_threshold$fitIndices[, 2], labels = powers, cex = 0.9,
      col = "red")

# this line corresponds to using an R^2 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")

```

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.

2) 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.

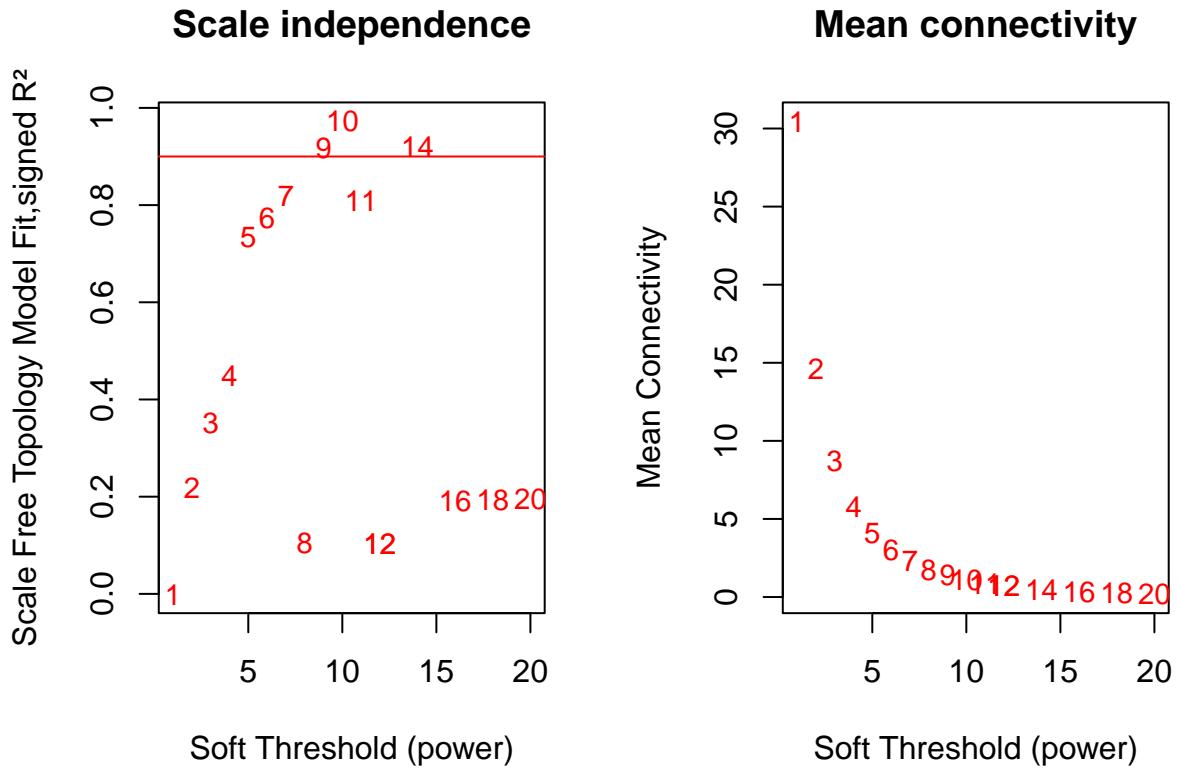


Figure 46: Scale independence and mean connectivity plots

```

# 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 (dendrogram)
plot(dendro, xlab = "", sub = "", main = "Gene Clustering on TOM-based dissimilarity",
      labels = FALSE, hang = 0.04)

```

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

Gene Clustering on TOM-based dissimilarity

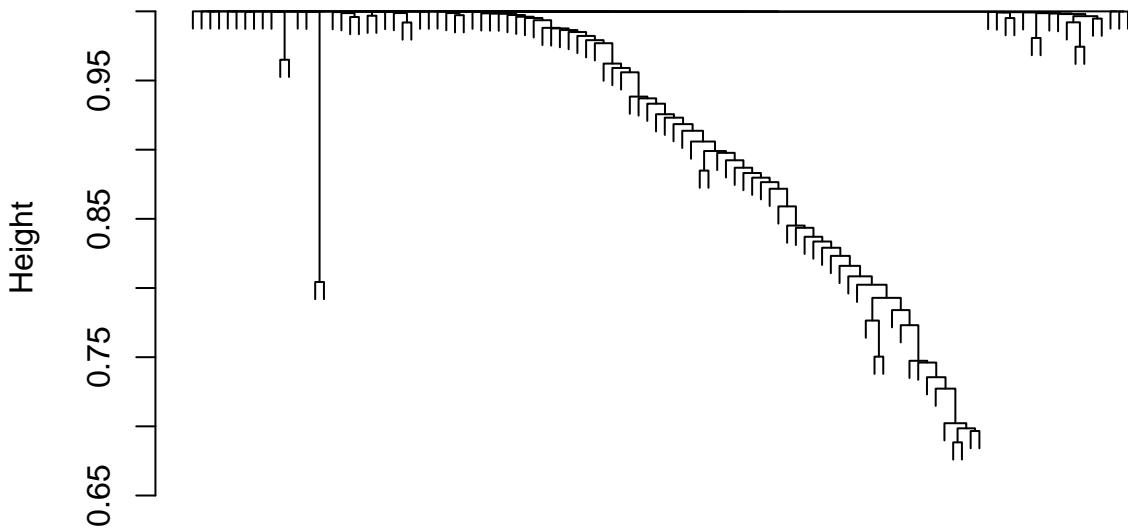


Figure 47: Dendrogram of gene clustering based on topological overlap matrix dissimilarity (TOM)

```
# Detect the modules
min_module_size <- 10
dynamic_modules <- cutreeDynamic(dendro = dendro, distM = dissimilarity_TOM_adj,
  deepSplit = 2, pamRespectsDendro = FALSE, minClusterSize = min_module_size,
  verbose = 0)

# Convert numeric labels 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")
```

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

3) 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),
```

Gene dendrogram and module colors

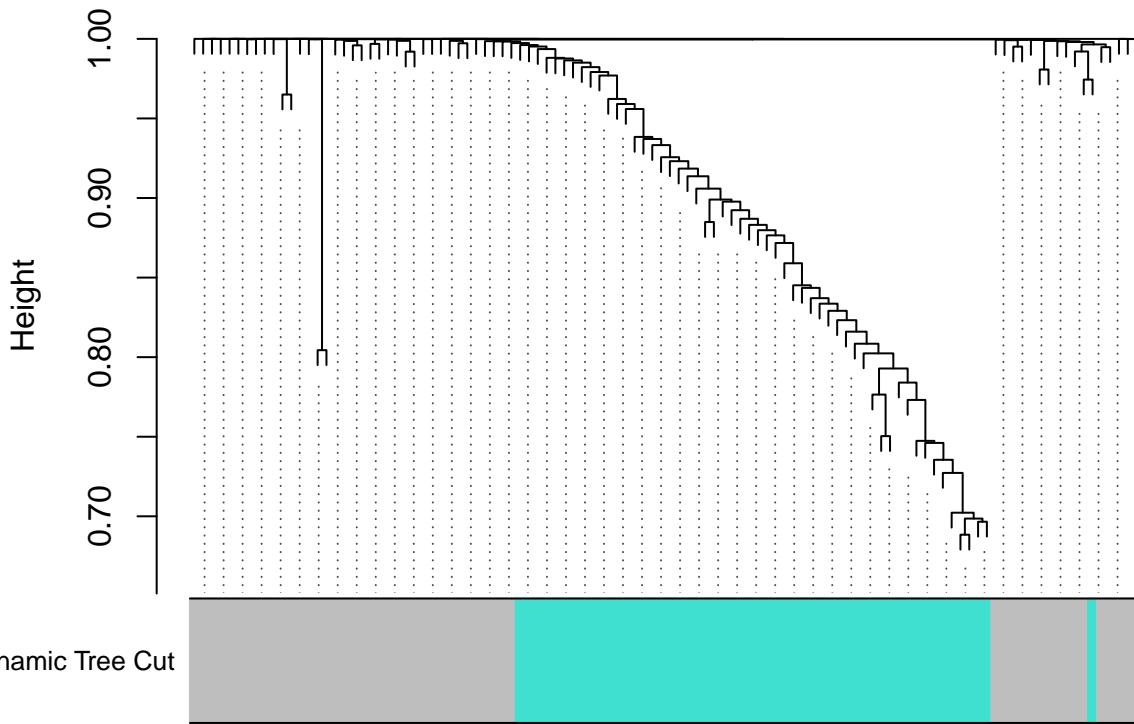


Figure 48: Dendrogram with module detection

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)

# pdf('../results/figures/heatmaps/heatmap_corr_cta_signif_conv.pdf',
# height = 8, width = 15) 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"))

# dev.off()
```

This pearson correlation matrix shows that there is a positive correlation between th2 cells and blue group,

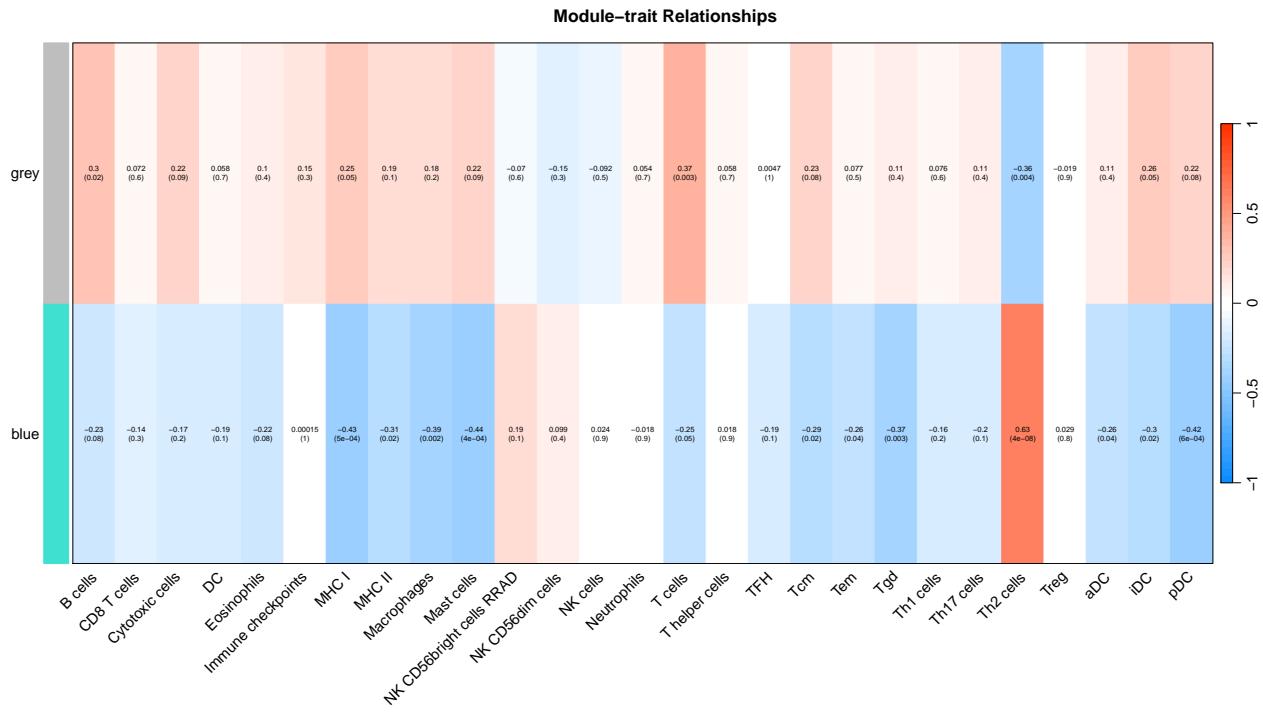


Figure 49: Pearson correlation matrix between modules and immune cells expression

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)
```