

# Pseudo bulk chondrosarcoma analysis

Léa ROGUE

2025-03-25

## List of Figures

1	Relative immune cells expression heatmap . . . . .	6
2	Relative expression of CTAs . . . . .	8

# Contents

Load libraries	4
Read data	4
Prepare data	4
I. Relative immune cells expression	5
II. Relative expression of CTAs	6

This script analyze pseudo bulk scRNAseq the same manner tha microarry part to see if it is coherent and to see the ifiltrated tumors or not..

## Load libraries

```
library(dplyr)
library(ComplexHeatmap)
library(colorRamp2)
```

## Read data

```
df_tpm <- read.table("../results/matrix_pseudo_bulk_tpm_normalized.tsv",
  sep = "\t", header = TRUE, check.names = F)

# Read CTA file
df_CTA <- read.table("../data/CTA_list_clean.txt", header = FALSE)

# Read immune cells
df_immune_sign <- read.table("../data/immune_cells_mhc_genes.tsv",
  header = FALSE, sep = "\t")

# Read list of CTA that impact survival probabilities
l_CTA_conv <- read.table("../data/CTA_signif_coxph_conv_indiv.txt",
  sep = "\t", header = FALSE)$V1
```

## Prepare data

```
df_tpm$CTA <- NA

# Rename col
colnames(df_CTA) <- c("SYMBOL")

# Df with CTA and whole genes by adding CTA in col CTA for
# CTA genes
df_tpm_CTA <- df_tpm %>%
  left_join(df_CTA, by = "SYMBOL") %>%
  mutate(CTA = ifelse(SYMBOL %in% df_CTA$SYMBOL, "CTA", NA))

# Reorganize columns
df_tpm_CTA <- df_tpm_CTA %>%
  select(SYMBOL, CTA, everything())

# Read immune cells genes
colnames(df_immune_sign) <- c("Signature", "Gene")
```

```

# Merge df_CTA_whole and df_immune_sign to associate
# signatures
df_tpm_CTA_imm_cells <- merge(df_tpm_CTA, df_immune_sign, by.x = "SYMBOL",
  by.y = "Gene", all.x = TRUE)

# Reorganize the columns if needed
df_tpm_CTA_imm_cells <- df_tpm_CTA_imm_cells %>%
  select(SYMBOL, CTA, Signature, everything())

# Combine multiple signatures into one for each gene by
# concatenating with a comma
df_tpm_CTA_imm_cells_clean <- df_tpm_CTA_imm_cells %>%
  group_by(SYMBOL) %>%
  summarise(Signature = paste(unique(Signature), collapse = ", "),
    across(everything(), ~first()), .groups = "drop")

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

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

```

## I. Relative immune cells expression

```

# Create the heatmap
heatmap_data <- as.data.frame(df_avg_immune_sign_final)
rownames(heatmap_data) <- heatmap_data$Signature
heatmap_data <- heatmap_data[, -1] # Remove the Signature column
heatmap_data <- log2(heatmap_data + 1)

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

# Add columns
df_z_scores <- as.data.frame(z_scores_row)
colnames(df_z_scores) <- c("Low_L07", "Low_L28", "High_L31",
  "High_L44", "Ben_L49", "Med_L63", "Med_L80", "Low_L81", "Ded_L83")

colors <- colorRampPalette(c("blue", "white", "red"))(100)
Heatmap(as.matrix(df_z_scores), 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 = 7),
row_names_gp = gpar(fontsize = 7), heatmap_legend_param = list(title = "Expression Level"))
```

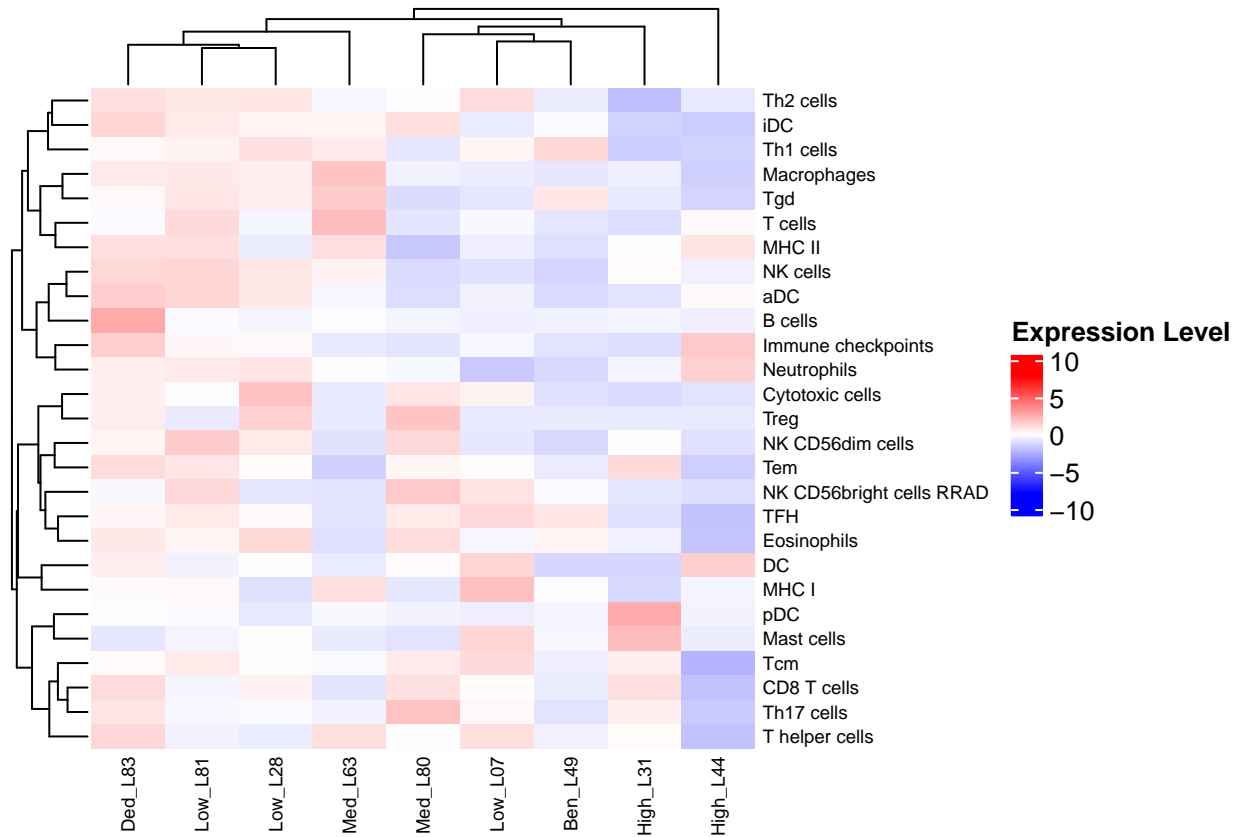


Figure 1: Relative immune cells expression heatmap

We see that the more infiltrated tumors are dedifferentiated and low tumors, contrary to the less infiltrated that are high grade.

## II. Relative expression of CTAs

```
# Create the heatmap
cta_data <- subset(df_tpm_CTA, df_tpm_CTA$CTA == "CTA")
rownames(cta_data) <- cta_data$SYMBOL
cta_data <- cta_data[, -c(1, 2)]
cta_data <- log2(cta_data + 1)

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

# Add columns
df_z_scores <- as.data.frame(z_scores_row)

Heatmap(as.matrix(df_z_scores), 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 = 7),
row_names_gp = gpar(fontsize = 4), heatmap_legend_param = list(title = "Expression Level"))
```

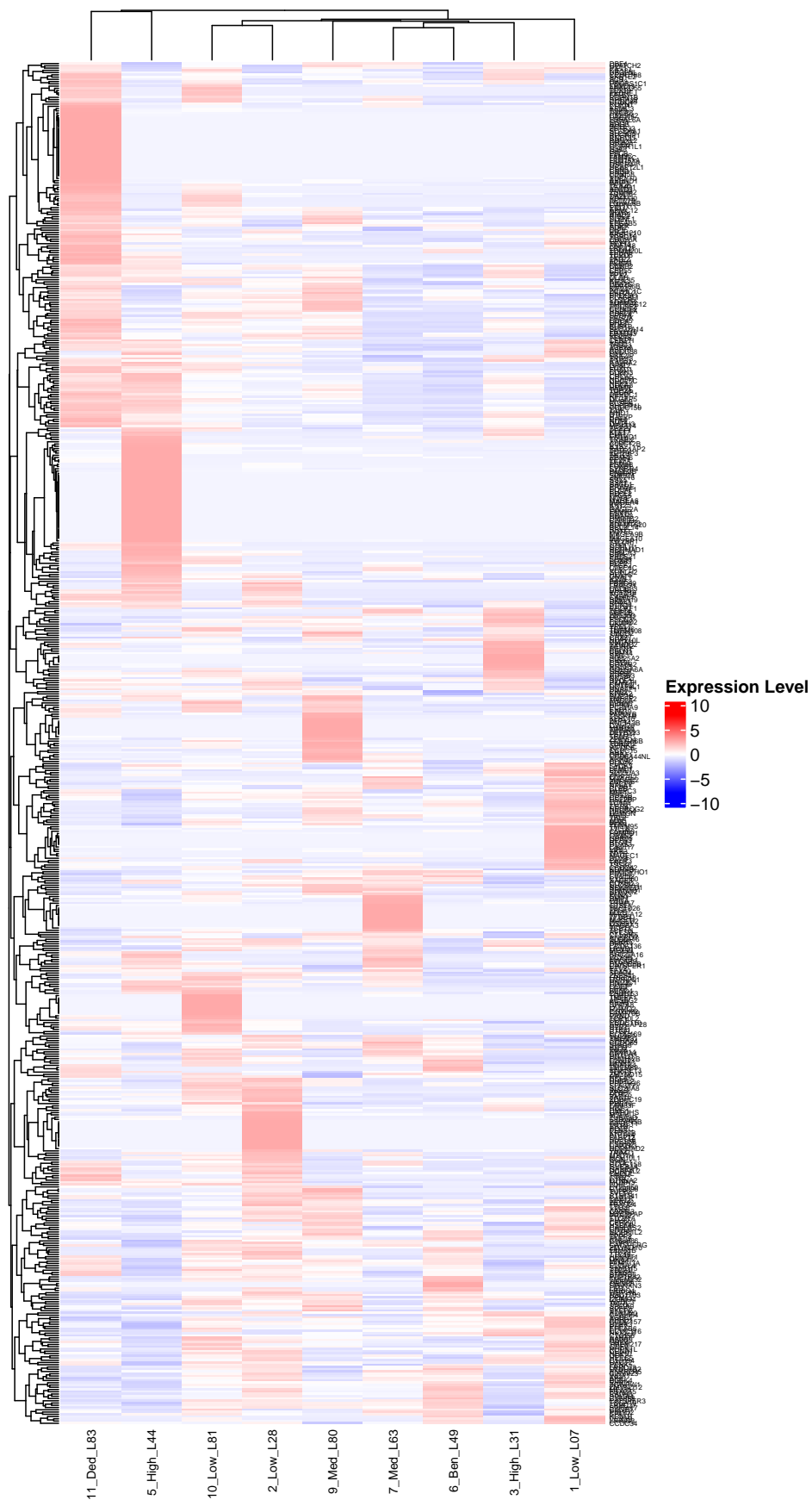


Figure 2: Relative expression of CTAs