RNA Clustering

Michael Callahan

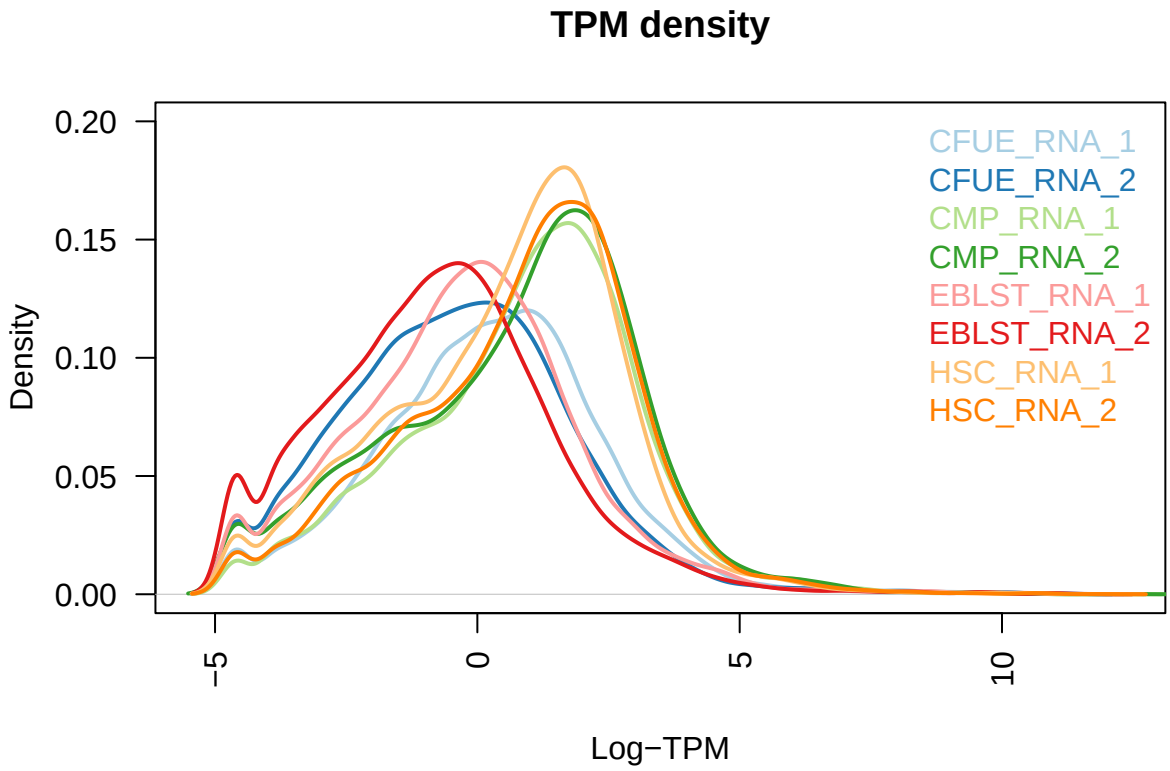
2024-04-03

Load data

```
library(here)
library(dplyr)

current_project_path <- here()

cols_dir_path <- file.path(current_project_path, "Data", "colData.Rdata")
counts_dir_path <- file.path(current_project_path, "Data", "countData.Rdata")

tpm_dir_path <- file.path(current_project_path, "Data", "tpm.Rdata")

load(cols_dir_path)
load(tpm_dir_path)
load(counts_dir_path)

tpm <- tpm %>% select(-length)
```

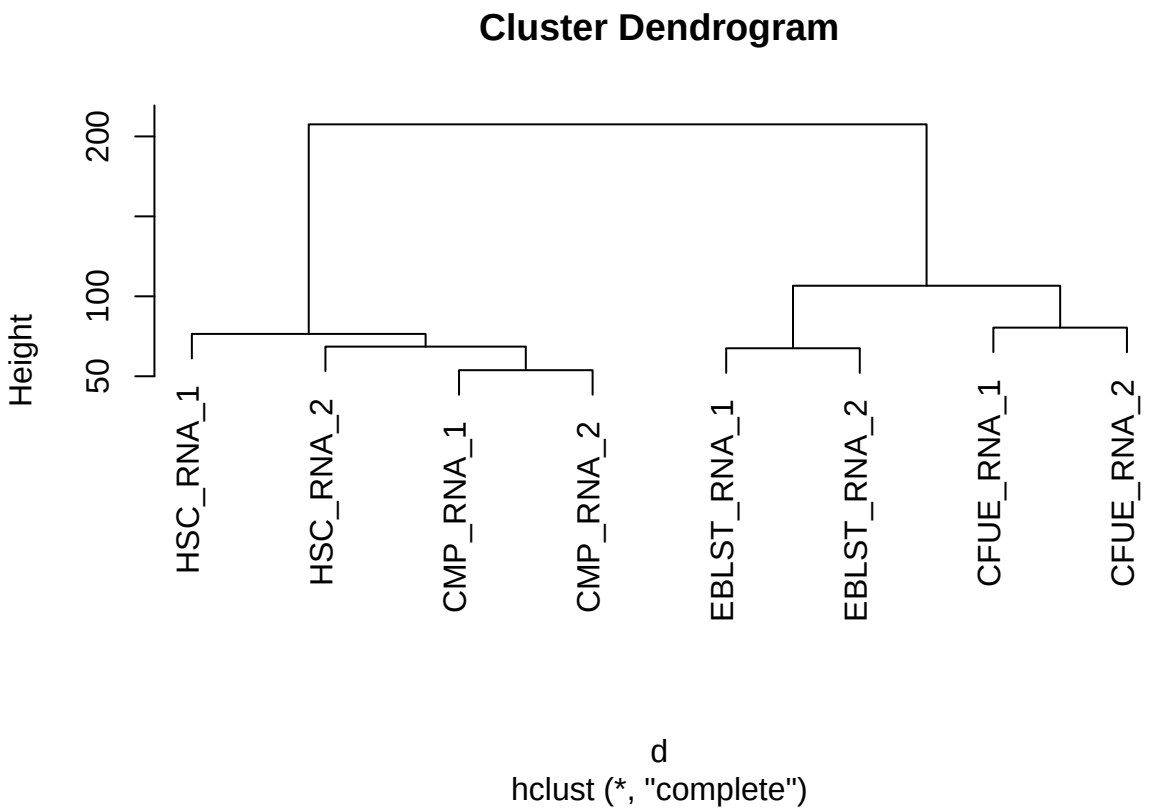
##Histogram overlay Here, we can see a density plot of the 4 samples being compared.

```
library(RColorBrewer)
tpm_filtered <- tpm[rowSums(tpm != 0) > 0, ]
samplenames <- colnames(tpm_filtered)
tpm.cutoff <- log2(0.1)
nsamples <- ncol(tpm_filtered)
col <- brewer.pal(nsamples, "Paired")
par(mfrow=c(1,1))
plot(density(log(tpm_filtered[,1])), col=col[1], lwd=2, ylim=c(0,0.2), las=2, main="", xlab="")
title(main="TPM density", xlab="Log-TPM")
for (i in 2:nsamples){
den <- density(log(tpm_filtered[,i]))
lines(den$x, den$y, col=col[i], lwd=2)
}
legend("topright", samplenames, text.col=col, bty="n")
```



Clustering

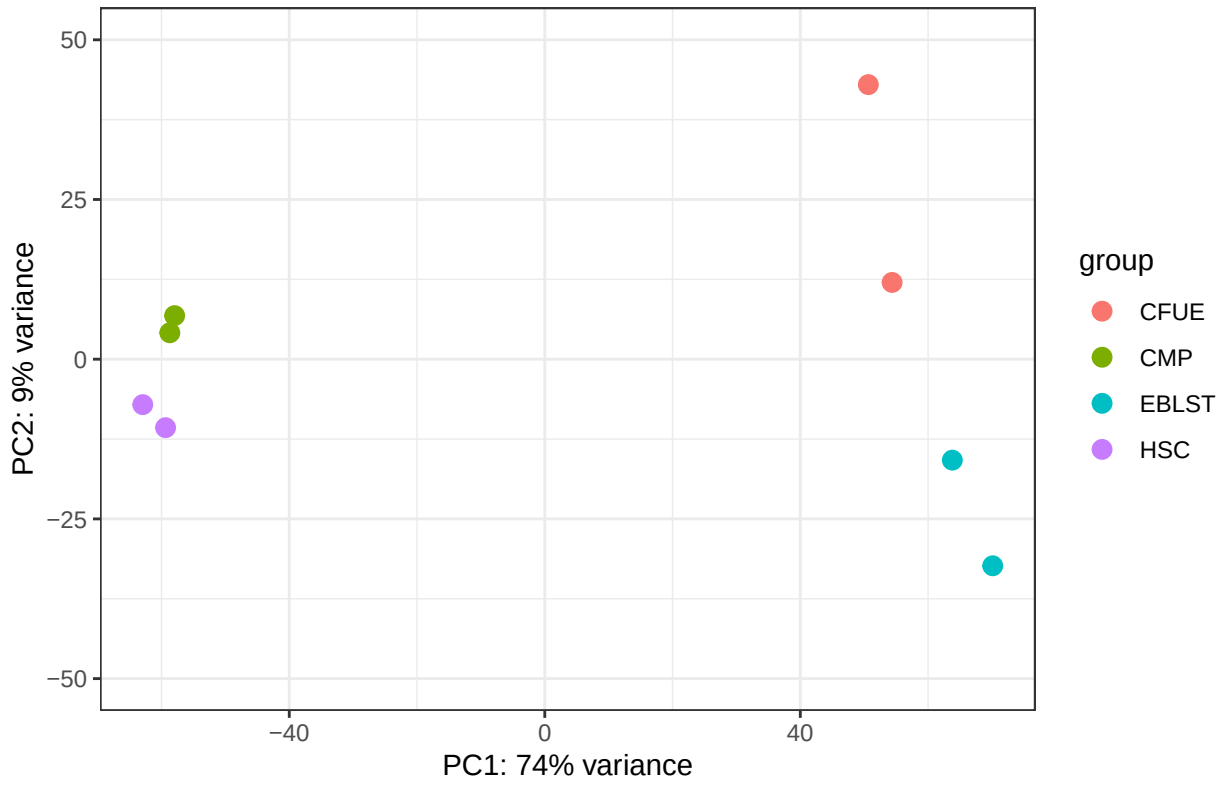
```
library(stats)

row_variances <- apply(tpm, 1, var)
tpm <- tpm[order(row_variances, decreasing = TRUE)[1:5000], ]
tpm <- log2(tpm + 1)

d=dist(t(tpm))
hc=hclust(d,method="complete")
plot(hc)
```



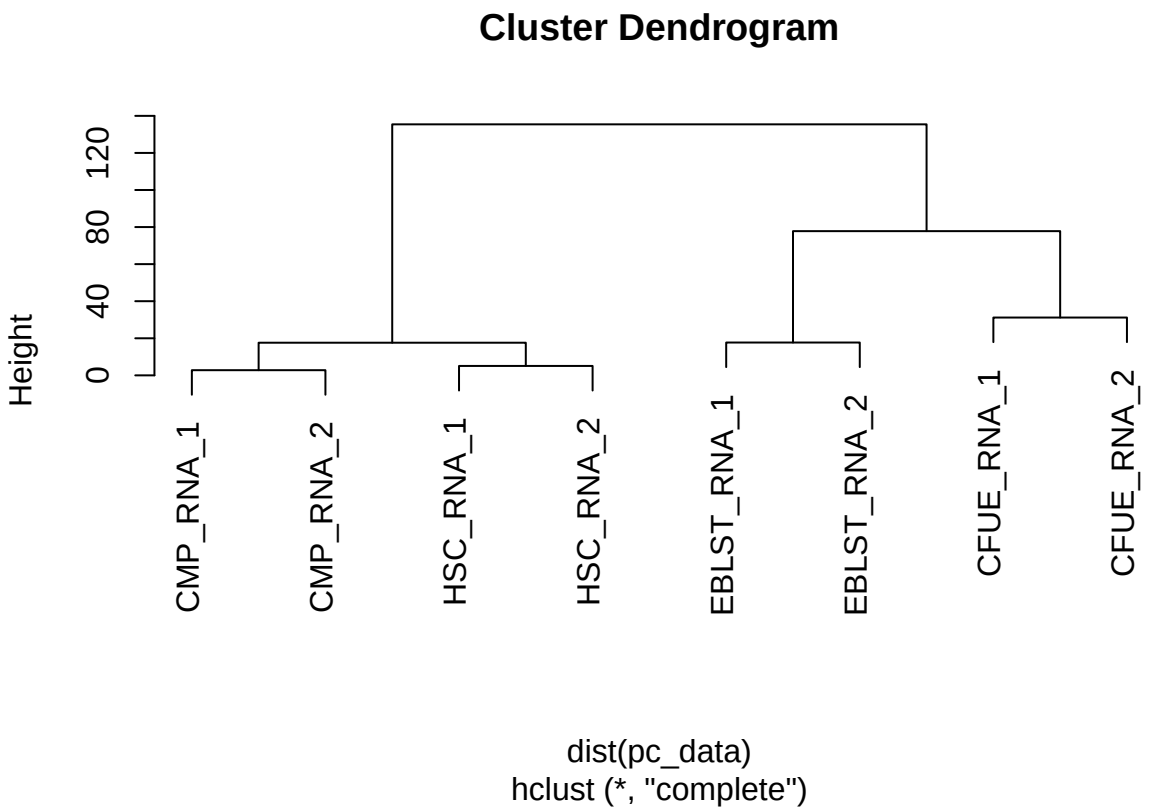
Advanced clustering and PCA

```
library(DESeq2)
library(scales)
library(ggplot2)

selected_samples = c()
selected_lines = c()

selected_lines = c('CMP', 'CFUE', 'EBLST', 'HSC')

selected_samples <- colData %>%
  filter(group %in% selected_lines) %>%
  mutate(group = factor(group, levels = selected_lines)) %>%
  arrange(group) %>%
  pull(source_name)

designFormula <- "~ group"

#create a DESeq dataset object from the count matrix and the colData
dds <- DESeqDataSetFromMatrix(
  countData = countData[,selected_samples],
  colData = colData[match(selected_samples, colData$source_name),],
  design = as.formula(designFormula)
```

```
)

#For each gene, we count the total number of reads for that gene in all samples
#and remove those that don't have at least 1 read.
dds <- dds[rowSums(DESeq2::counts(dds)) > 1, ]

dds <- DESeq(dds)

# Normalized cluster plot
rld <- rlog(dds)
de_pc <- DESeq2::plotPCA(rld, ntop = 500, intgroup = 'group') + ylim(-50, 50) + theme_bw()
plot(de_pc)
```

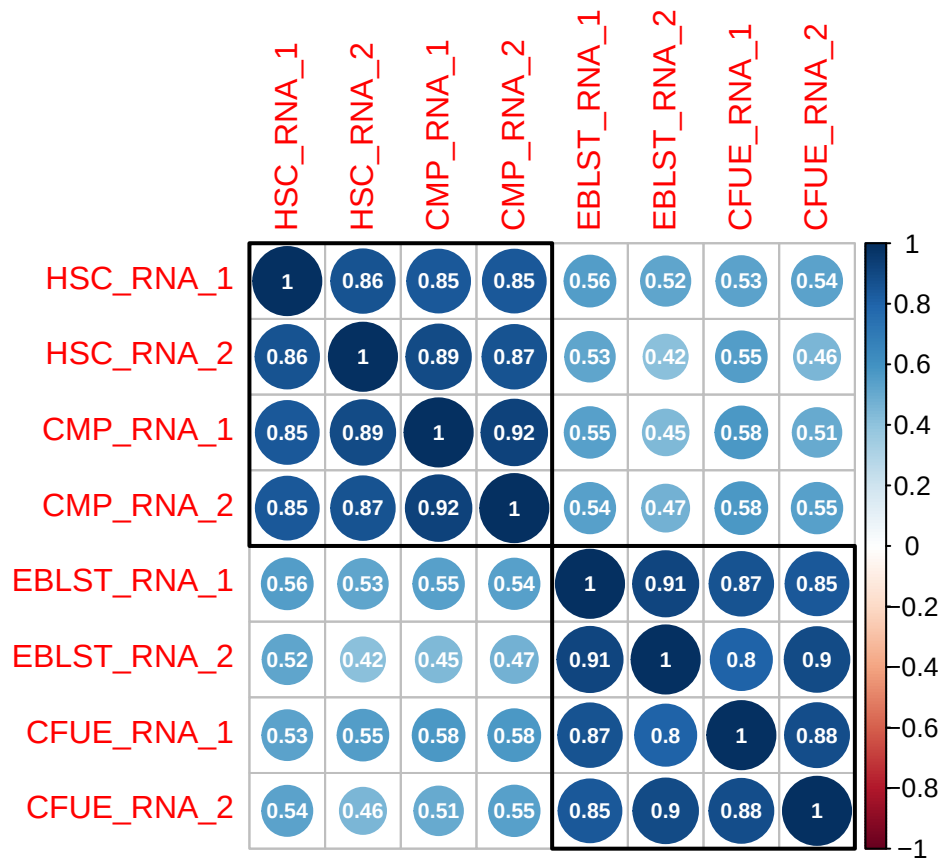



```
pc_coords <- ggplot_build(de_pc)$data[[1]]
pc_data <- pc_coords[, c("x", "y")]
rownames(pc_data) = selected_samples

hc <- hclust(dist(pc_data))
plot(hc)
```



Correlation plots

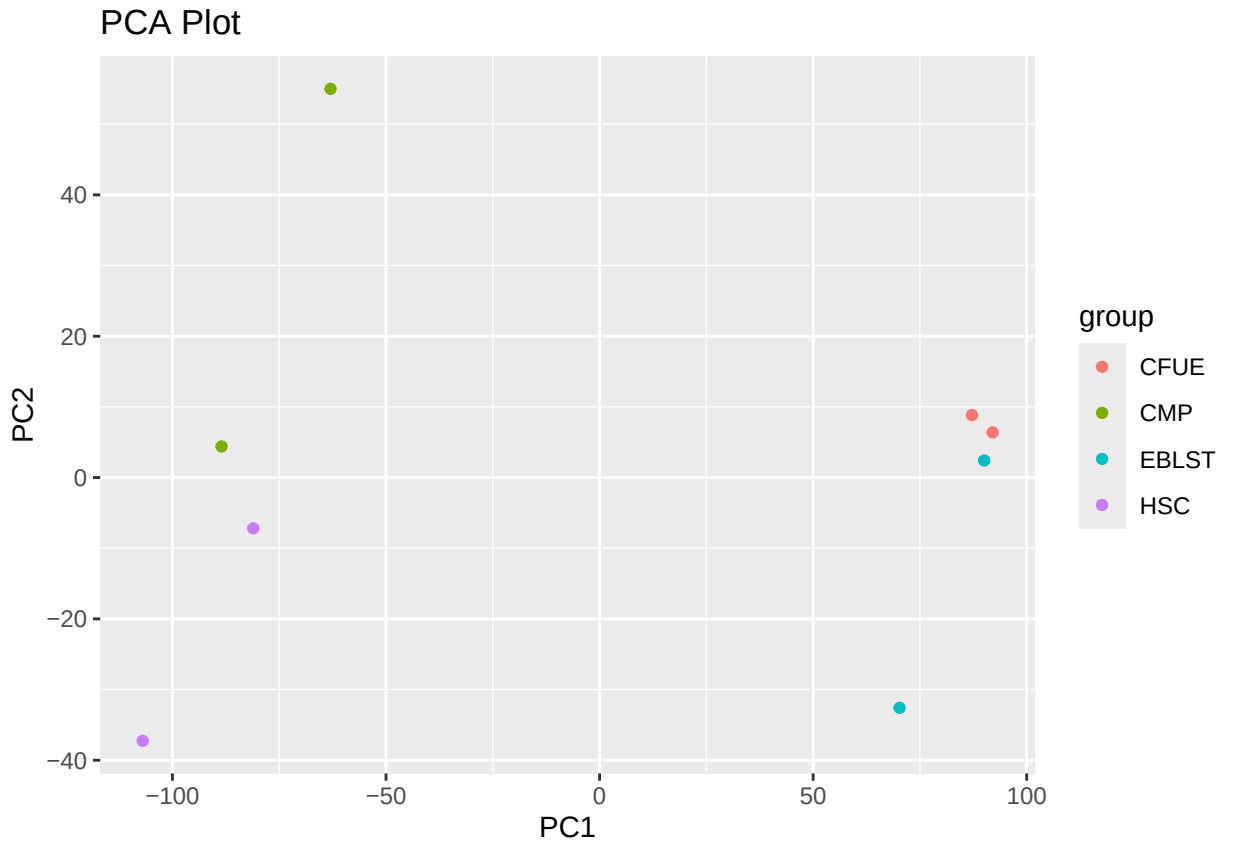
```
library(corrplot)
correlationMatrix <- cor(tpm)
corrplot(correlationMatrix, order = 'hclust',
         addrect = 2, addCoef.col = 'white',
         number.cex = 0.7)
```



##PCA – TPM

```
library(ggplot2)

pcaResults <- prcomp(t(tpm))
pc <- data.frame(PC1 = pcaResults$x[,1],
                 PC2 = pcaResults$x[,2],
                 PC3 = pcaResults$x[,3],
                 PC4 = pcaResults$x[,4],
                 PC5 = pcaResults$x[,5]
                 )

#Create a colData matrix
colData = data.frame(
  source_name = c('CMP_RNA_1','CMP_RNA_2',
                  'CFUE_RNA_1','CFUE_RNA_2',
                  'HSC_RNA_1','HSC_RNA_2',
                  'EBLST_RNA_1','EBLST_RNA_2'
                 ),
  group = c('CMP', 'CMP', 'CFUE', 'CFUE', 'HSC', 'HSC', 'EBLST', 'EBLST')
  )

# Add sample metadata from colData
pc <- cbind(pc, colData)

# Plot PCA results using ggplot2
```

```
ggplot(pc, aes(x = PC1, y = PC2, color = group)) +
  geom_point() +
  labs(title = "PCA Plot", x = "PC1", y = "PC2")
```




```
summary(pcaResults)
```

```
## Importance of components:
##                             PC1      PC2      PC3      PC4     PC5      PC6
## Standard deviation     91.7525 28.46336 23.46976 19.62037 15.1122 13.73010
## Proportion of Variance  0.7853  0.07557  0.05138  0.03591  0.0213  0.01759
## Cumulative Proportion   0.7853  0.86089  0.91227  0.94818  0.9695  0.98707
##                             PC7       PC8
## Standard deviation     11.77356 8.776e-14
## Proportion of Variance  0.01293 0.000e+00
## Cumulative Proportion   1.00000 1.000e+00
```

EDA - Heat map

Here, we look at the TPM (trascripts per million) for our two cell lines (2 replicates each), and we pull out the top 100 genes with the highest variance of expression across samples. Then we plot their normalized expression levels across the samples using a heat map. We should see that the replicates within each cell line should have a more similar expression pattern than across cell lines.

```r
library(pheatmap)

#compute the variance of each gene across samples
variances <- apply(tpm, 1, var)
colData$group = as.character(colData$group)
selectedGenes <- order(variances, decreasing = TRUE)[1:100]
pheatmap(tpm[selectedGenes,], scale = 'row', show_rownames = FALSE, cluster_cols = FALSE)
```