# Class 16 EC

Angela Liu

## Downstream analysis

We will download the R function, `tximport()` so that we can import the Kallisto results.

```
# BiocManager::install("tximport")
library(tximport)

#BiocManager::install("rhdf5")
library(rhdf5)
```

Each sample has its own directory with the Kallisto output. We can import the transcript count estimates into R.

```
# setup the folder and filenames to read
folders <- dir(pattern="SRR21568*")
samples <- sub("_quant", "", folders)
files <- file.path( folders, "abundance.h5" )
names(files) <- samples

txi.kallisto <- tximport(files, type = "kallisto", txOut = TRUE)
```

```
1 2 3 4
```

```
head(txi.kallisto$counts)
```

```
                SRR2156848 SRR2156849 SRR2156850 SRR2156851
ENST00000539570          0          0    0.00000          0
ENST00000576455          0          0    2.62037          0
ENST00000510508          0          0    0.00000          0
```

```
ENST00000474471          0          1    1.00000          0
ENST00000381700          0          0    0.00000          0
ENST00000445946          0          0    0.00000          0
```

Now, we have estimated the transcript counts for each sample.

To see how many transcripts per sample:

```
colSums(txi.kallisto$counts)
```

```
SRR2156848 SRR2156849 SRR2156850 SRR2156851
   2563611    2600800    2372309    2111474
```

Let's see how many transcripts are detected in at least one sample:

```
# look at only the rows with more than zero transcripts
sum(rowSums(txi.kallisto$counts)>0)
```

```
[1] 94561
```

Next, let's filter out any of the annotated transcripts with no reads.

```
# keep only the transcripts with reads
to.keep <- rowSums(txi.kallisto$counts) > 0
kset.nonzero <- txi.kallisto$counts[to.keep,]
```

We will also filter the annotated transcripts with no change over the samples.

```
keep2 <- apply(kset.nonzero,1,sd)>0
x <- kset.nonzero[keep2,]
```

## PCA

We'll now apply PCA to the transcriptomic profiles of these samples.

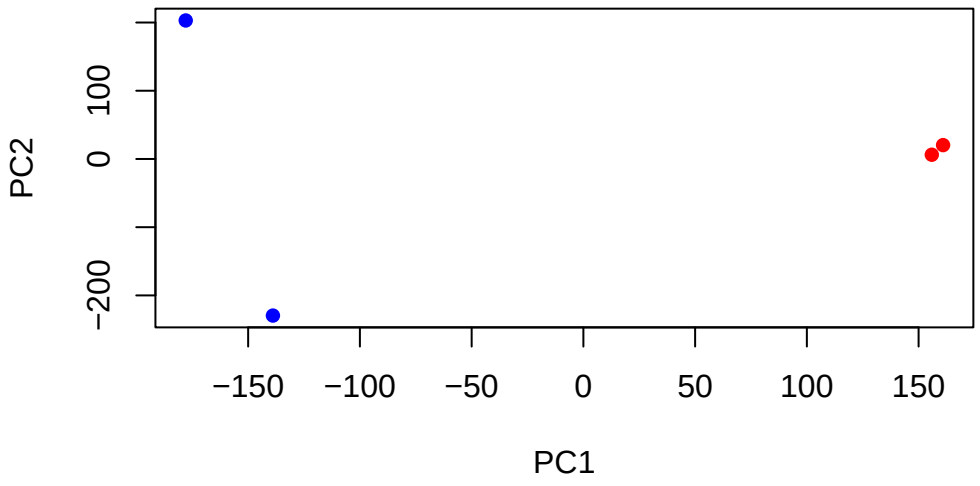
```
pca <- prcomp(t(x), scale=TRUE)
```

```
summary(pca)
```

```
Importance of components:
                            PC1      PC2      PC3   PC4
Standard deviation     183.6379 177.3605 171.3020 1e+00
Proportion of Variance   0.3568   0.3328   0.3104 1e-05
Cumulative Proportion    0.3568   0.6895   1.0000 1e+00
```

Let's use the first two PC's as coordinate system for visualizing the summarized transcriptomic profiles of each sample.

```
plot(pca$x[,1], pca$x[,2],
     col=c("blue","blue","red","red"),
     xlab="PC1", ylab="PC2", pch=16)
```



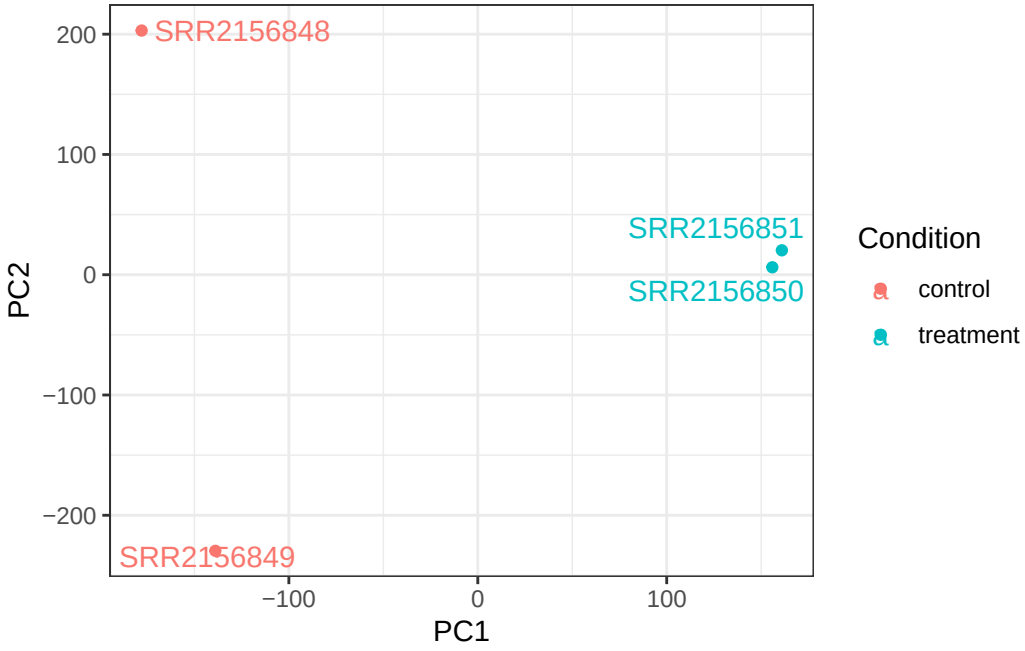

Q. Use ggplot to make a similar figure of PC1 vs PC2 and a separate figure PC1 vs PC3 and PC2 vs PC3.

```
library(ggplot2)
library(ggrepel)

# Make metadata object for the samples
colData <- data.frame(condition = factor(rep(c("control", "treatment"), each = 2)))
rownames(colData) <- colnames(txi.kallisto$counts)
```

```
# Make the data.frame for ggplot
y <- as.data.frame(pca$x)
y$Condition <- as.factor(colData$condition)

ggplot(y) +
  aes(PC1, PC2, col=Condition) +
  geom_point() +
  geom_text_repel(label=rownames(y)) +
  theme_bw()
```




This plot shows that PC1 sepearates the controls of SRR2156848 and SRR2156849 (pink data points) from the enhancer-targeting CRISPR samples (teal data points). PC2 separates the control samples from each other and PC3 separates the CRISPR samples. This graph shows that there is a difference between the treated and control samples.