# Class 17: Extra Credit - Analyzing sequencing data in the cloud

Pamelina Lo (AID: 16735368)

## Downstream analysis

For this section of the lab, we can now use R and Bioconductor tools to further explore this large scale dataset.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install(c("rhdf5", "tximport"))
```

```
Bioconductor version 3.19 (BiocManager 1.30.25), R 4.4.1 (2024-06-14)
```

```
Warning: package(s) not installed when version(s) same as or greater than current; use
  `force = TRUE` to re-install: 'rhdf5' 'tximport'
```

```
Old packages: 'boot', 'curl', 'dendextend', 'evaluate', 'fontawesome',
  'foreign', 'fs', 'glue', 'gtable', 'httr2', 'knitr', 'later', 'MASS',
  'Matrix', 'mvtnorm', 'nlme', 'promises', 'quantreg', 'Rcpp', 'RcppArmadillo',
  'rmarkdown', 'RSQLite', 'survival', 'tinytex', 'usethis', 'waldo', 'withr',
  'xfun'
```

```
library(rhdf5)
library(tximport)

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
```

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

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

How many transcripts are detected in at least one sample?

```
sum(rowSums(txi.kallisto$counts)>0)
```

```
[1] 94561
```

```
to.keep <- rowSums(txi.kallisto$counts) > 0
kset.nonzero <- txi.kallisto$counts[to.keep,]
```

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

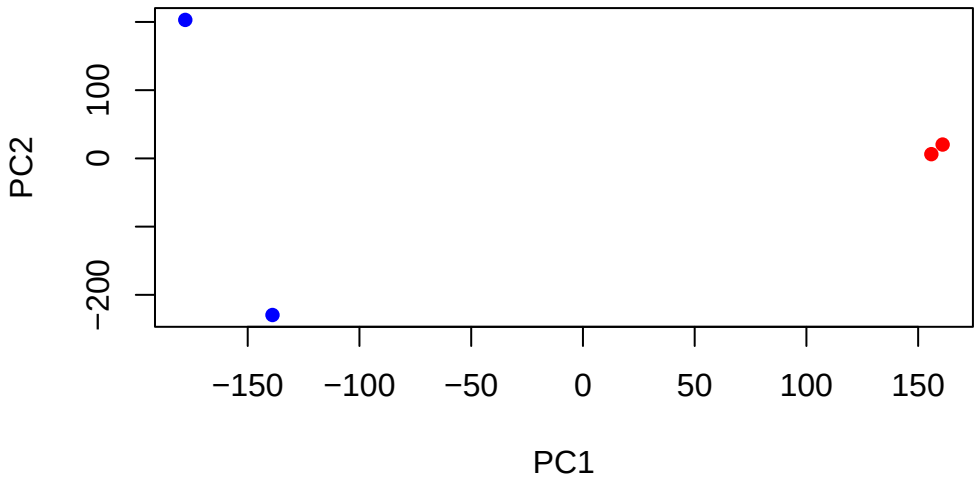
## Principal Component Analysis

We can now apply any exploratory analysis technique to this counts matrix. As an example, we will perform a PCA of 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
```

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

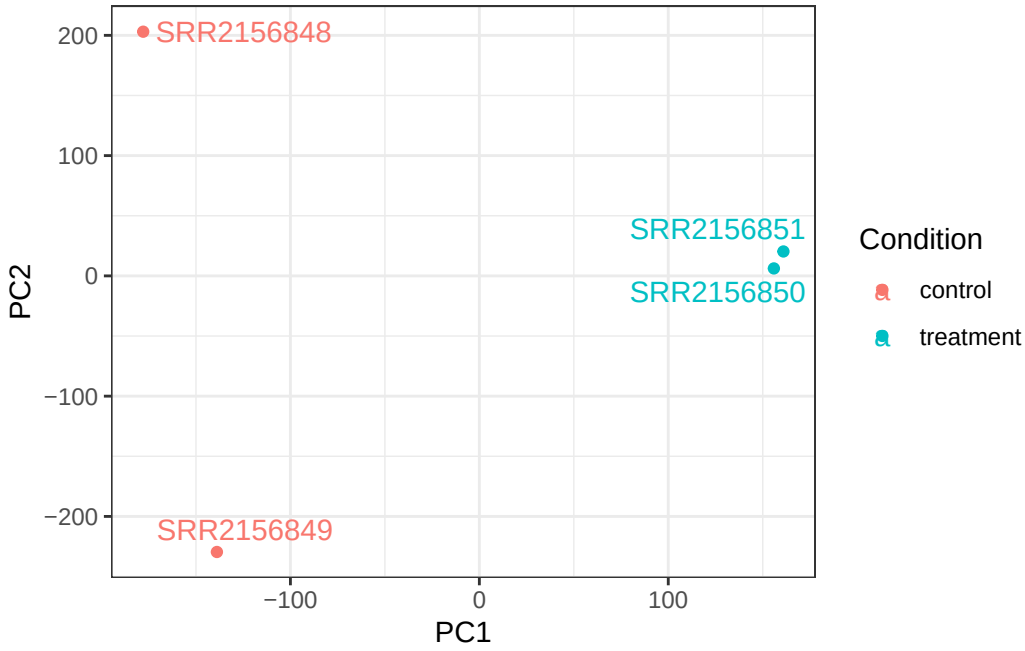



Use ggplot to make similar PC1 vs PC2 and a seperate figure PC1 vs PC3 and PC2 vs PC3.

**PC1 vs PC2**

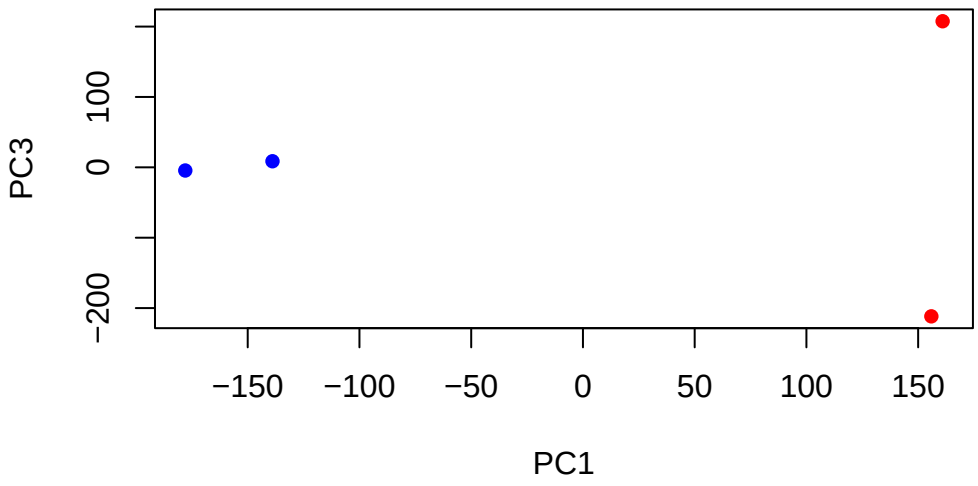
```
library(ggplot2)
library(ggrepel)

colData <- data.frame(condition = factor(rep(c("control", "treatment"), each = 2)))
```

```
rownames(colData) <- colnames(txi.kallisto$counts)

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()
```

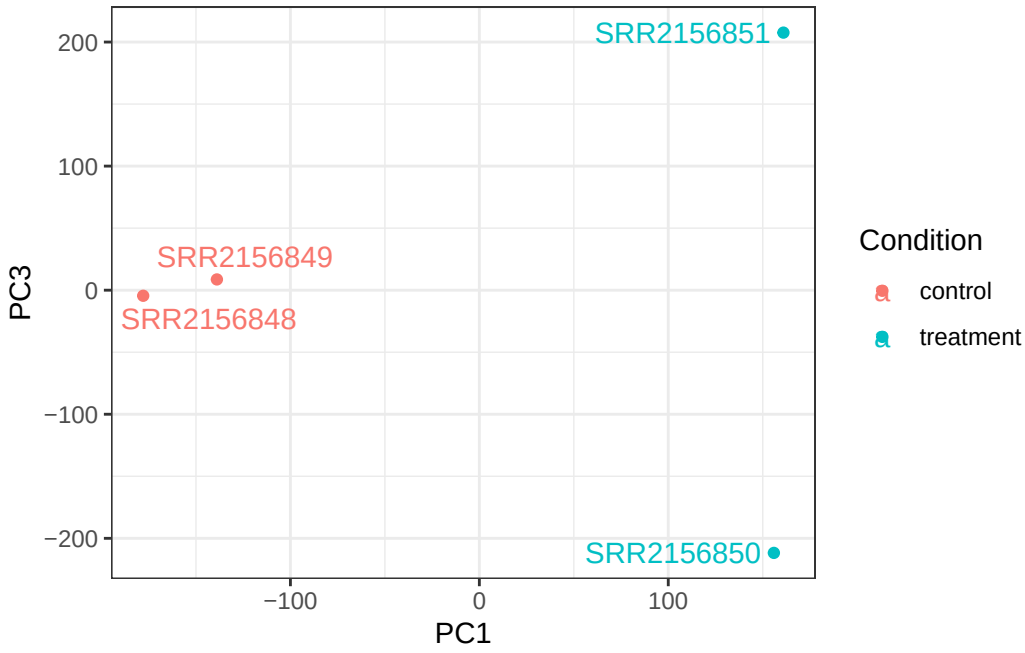



### PC1 vs PC3

```
plot(pca$x[,1], pca$x[,3],
     col=c("blue","blue","red","red"),
     xlab="PC1", ylab="PC3", pch=16)
```

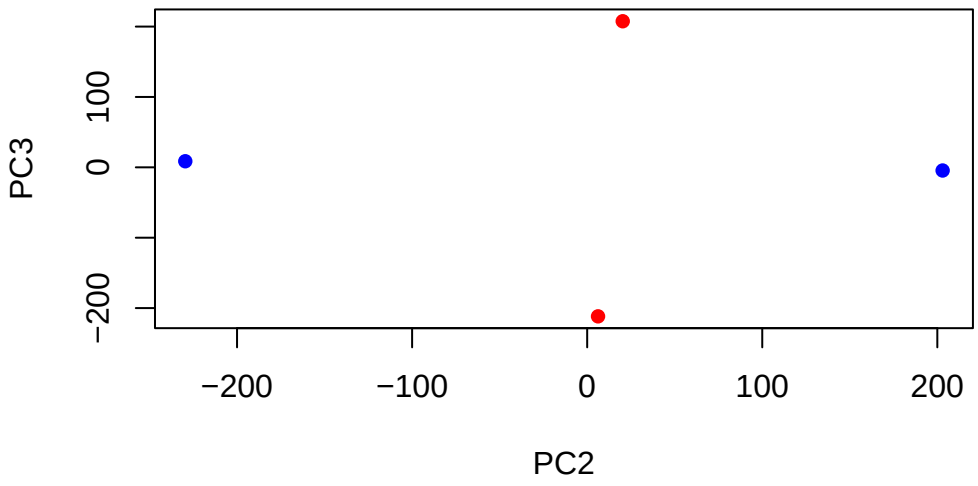



```
colData <- data.frame(condition = factor(rep(c("control", "treatment"), each = 2)))
rownames(colData) <- colnames(txi.kallisto$counts)

y <- as.data.frame(pca$x)
y$Condition <- as.factor(colData$condition)

ggplot(y) +
  aes(PC1, PC3, col=Condition) +
  geom_point() +
  geom_text_repel(label=rownames(y)) +
  theme_bw()
```

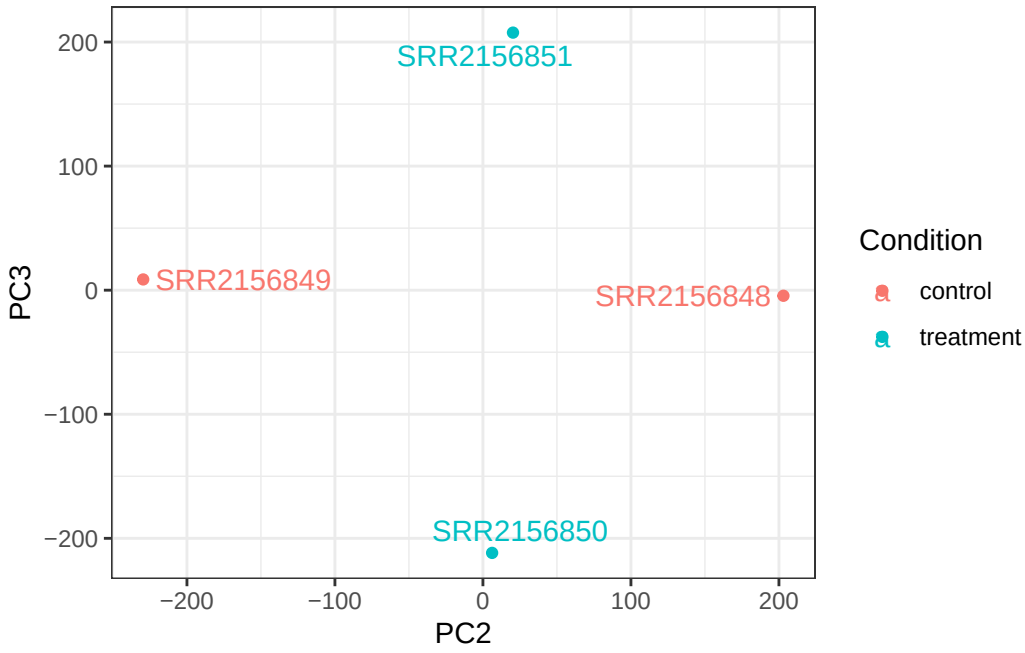



### PC2 vs PC3

```
plot(pca$x[,2], pca$x[,3],
     col=c("blue","blue","red","red"),
     xlab="PC2", ylab="PC3", pch=16)
```




```
colData <- data.frame(condition = factor(rep(c("control", "treatment"), each = 2)))
rownames(colData) <- colnames(txi.kallisto$counts)

y <- as.data.frame(pca$x)
y$Condition <- as.factor(colData$condition)

ggplot(y) +
  aes(PC2, PC3, col=Condition) +
  geom_point() +
  geom_text_repel(label=rownames(y)) +
  theme_bw()
```


SRR2156851
SRR2156849
SRR2156848
SRR2156850
PC3
PC2
200
100
0
−100
−200
−200
−100
0
100
200
Condition
control
treatment