

class 17

Mason Lew (PID: A17533139)

2025-03-12

Table of contents

After doing a bunch of stuff in the terminal, we can import our data back to R-studio

```
library(tximport)

# 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

We now have our estimated transcript counts for each sample in R. We can see how many transcripts we have for each sample:

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

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

And how many transcripts are detected in at least one sample:

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

```
[1] 94561
```

Before subsequent analysis, we might want to filter out those annotated transcripts with no reads:

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

And those with no change over the samples:

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

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.

Now we compute the principal components, centering and scaling each transcript's measured levels so that each feature contributes equally to the PCA:

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

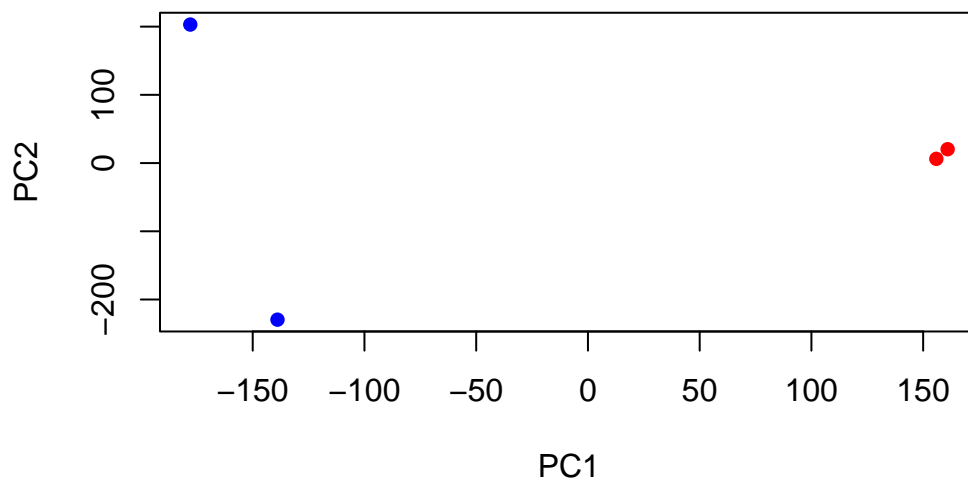
```
summary(pca)
```

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

Now we can use the first two principal components as a co-ordinate 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)
```



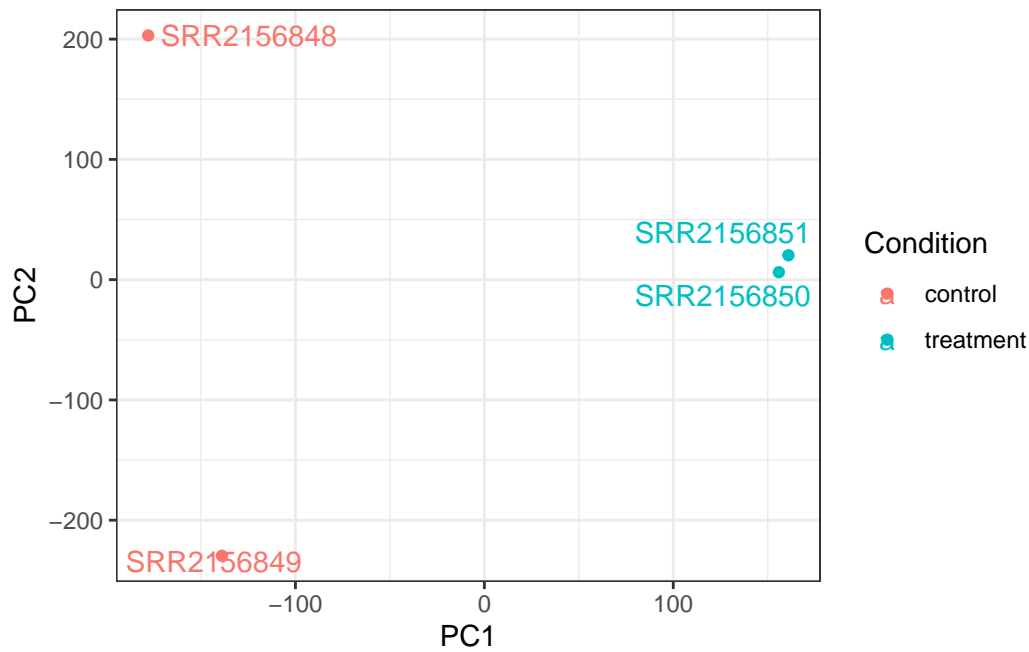
Q. Use ggplot to make a similar figure of PC1 vs PC2 and a separate figure PC1 vs PC3 and PC2 vs PC3.

```
library(ggplot2)
library(ggrepel)

# Make metadata object for the samples
colData <- data.frame(condition = factor(rep(c("control", "treatment"), each = 2)))
rownames(colData) <- colnames(tx1.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()
```



The plot makes it clear that PC1 separates the two control samples (SRR2156848 and SRR2156849) from the two enhancer-targeting CRISPR-Cas9 samples (SRR2156850 and SRR2156851). PC2 separates the two control samples from each other, and PC3 separates the two enhancer-targeting CRISPR samples from each other. This could be investigated further to see which genes result in these separation patterns. It is also at least slightly reassuring, implying that there are considerable differences between the treated and control samples.