Class 16: Analyzing sequencing data in the cloud

AUTHOR
Hyeseung (Frankie) Son PID: A16025601

Downstream analysis

Back on our laptop we can now use R and Bioconductor tools to further explore this large scale dataset.

There's an R function called tximport() in the tximport package, which enables import of Kallisto results

With each sample having its own directory containing the Kallisto output, we can import the transcript count estimates into R using:

```
library(tximport)
library(rhdf5)

# 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
```

Here's the estimated transcript counts for each sample in R. We see how many transcripts in 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, let's filter out 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,]
```

Principal Component Analysis

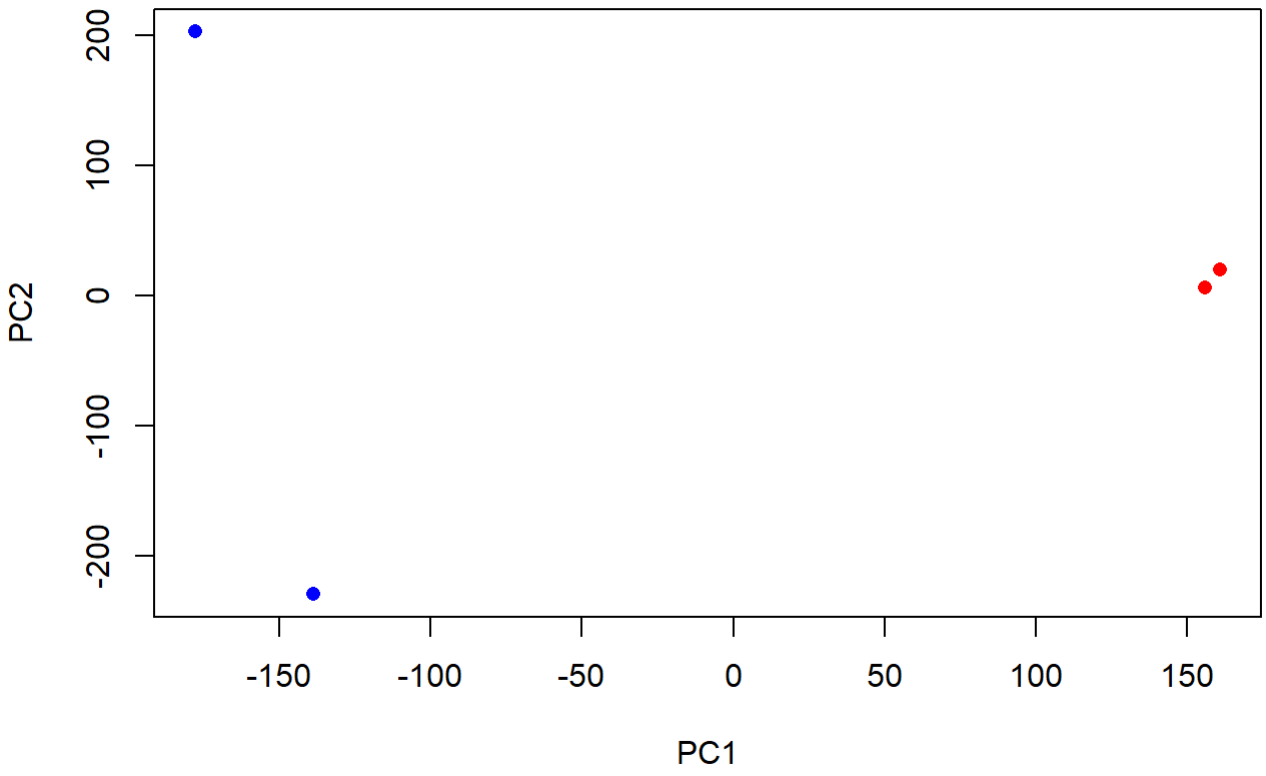
Let's perform a PCA of the transcriptomic profiles of these samples. We'll compute the principal components, centering and scaling each transcript's measured levels so that each feature contributes equally to the PCA:

```
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
```

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 seperate figure PC1 vs PC3 and PC2 vs PC3.

```
library(ggplot2)
```

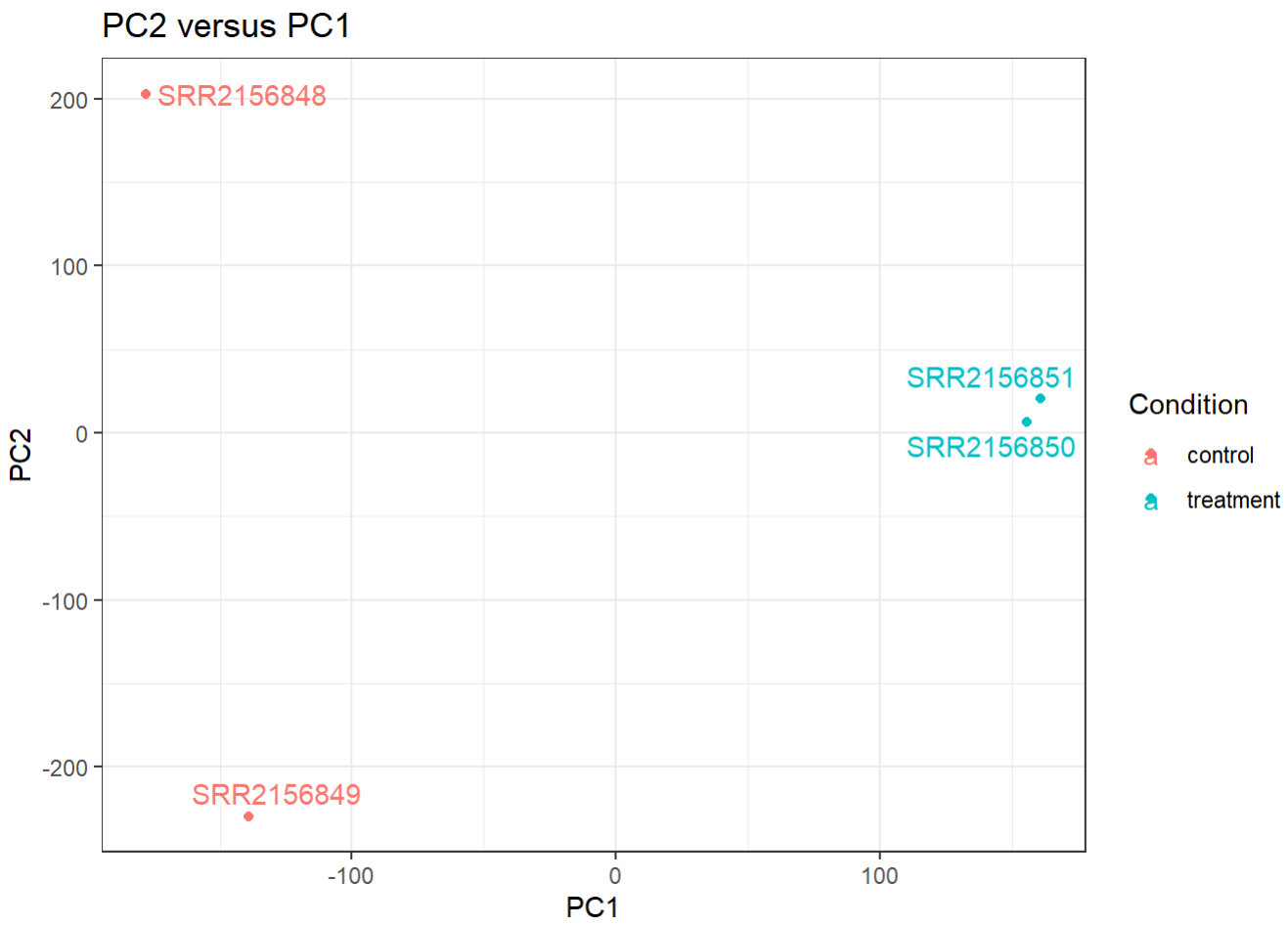
```
Warning: package 'ggplot2' was built under R version 4.2.3
```

```
library(ggrepel)
```

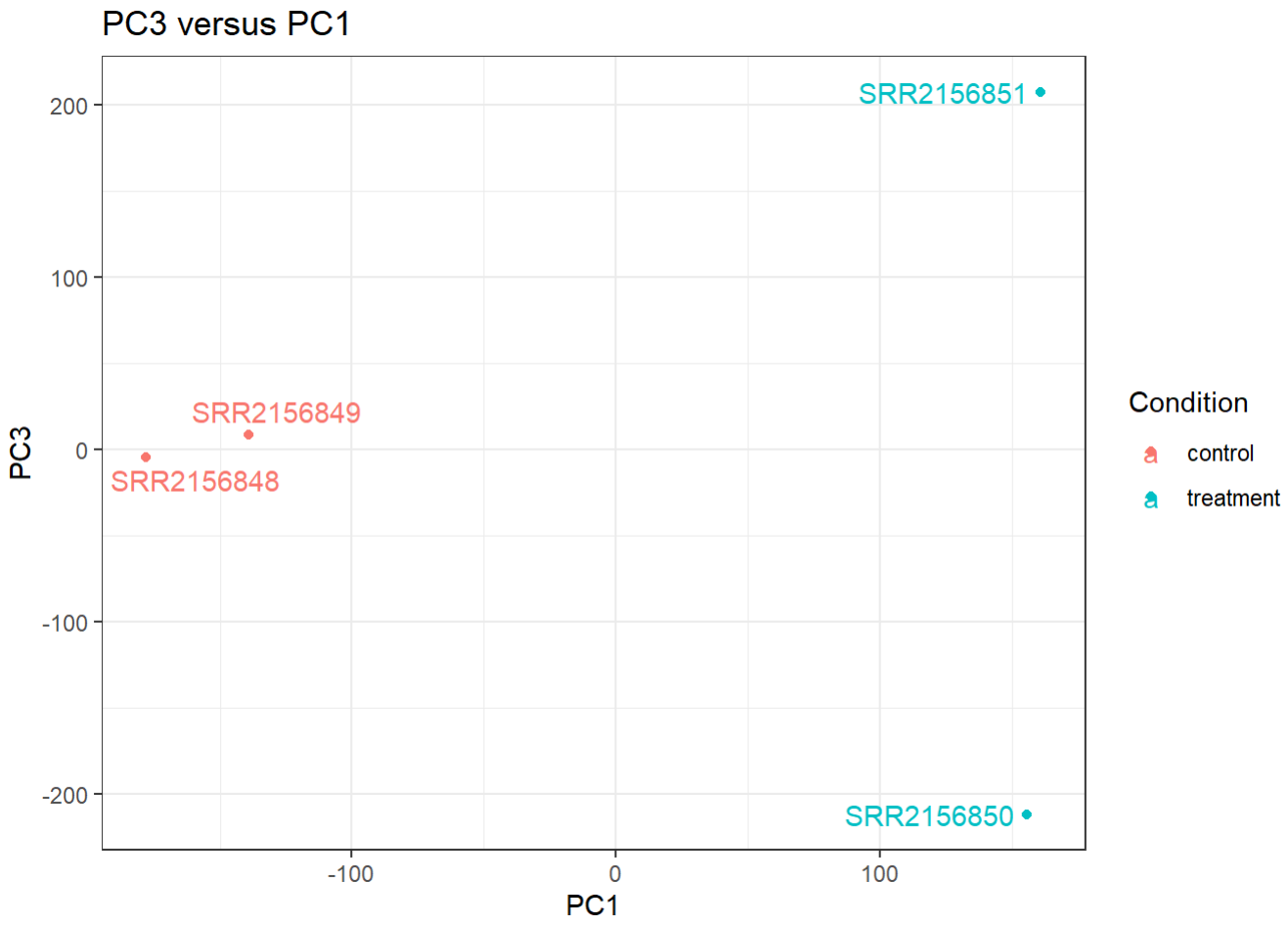
```
Warning: package 'ggrepel' was built under R version 4.2.3
```

```
# 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() +
```

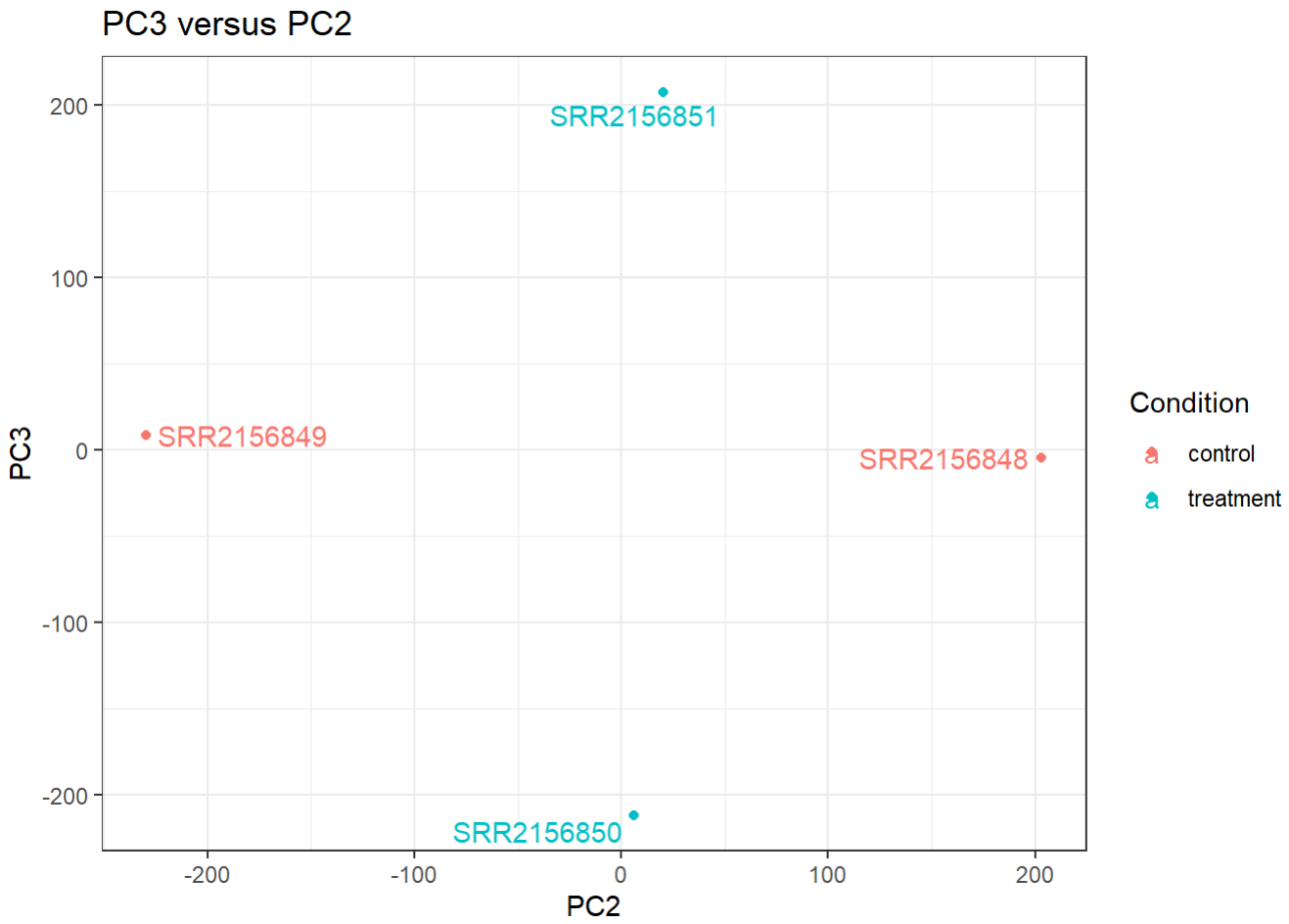
```
  geom_text_repel(label=rownames(y)) +
  theme_bw() +
  labs(title = "PC2 versus PC1")
```




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




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



The plot makes clear:

- PC1 separates 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.
- PC3 separates the two enhancer-targeting CRISPR samples from each other.