Class 16 - Analyzing sequencing data in the cloud

Hannah Kim

6/15/23

Downstream Analysis

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

we can import the transcript count estimates into R using:

```
# Install BiocManager if not already installed
# BiocManager::install("tximport")
#BiocManager::install("rhdf5")
```

```
#load tximport library
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
```

```
# setup the folder and filenames to read
folders <- dir(pattern="SRR21568*")
samples <- sub("_quant", "", folders)
```

```r
files <- file.path( folders, "abundance.h5" )
names(files) <- samples
```

```r
# Display the head of the counts matrix
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:

```r
# Compute column sums of the counts matrix
colSums(txi.kallisto$counts)
```

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

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

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

```
[1] 94561
```

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

```r
# Identify rows with non-zero counts
to.keep <- rowSums(txi.kallisto$counts) > 0

# Subset the counts matrix to include rows with non-zero counts
kset.nonzero <- txi.kallisto$counts[to.keep,]
```

And those with no change over the samples:

```
# Calculate row-wise standard deviation and identify rows with non-zero standard deviation
keep2 <- apply(kset.nonzero,1,sd)>0

x <- kset.nonzero[keep2,]
```

Principal Component Analysis

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

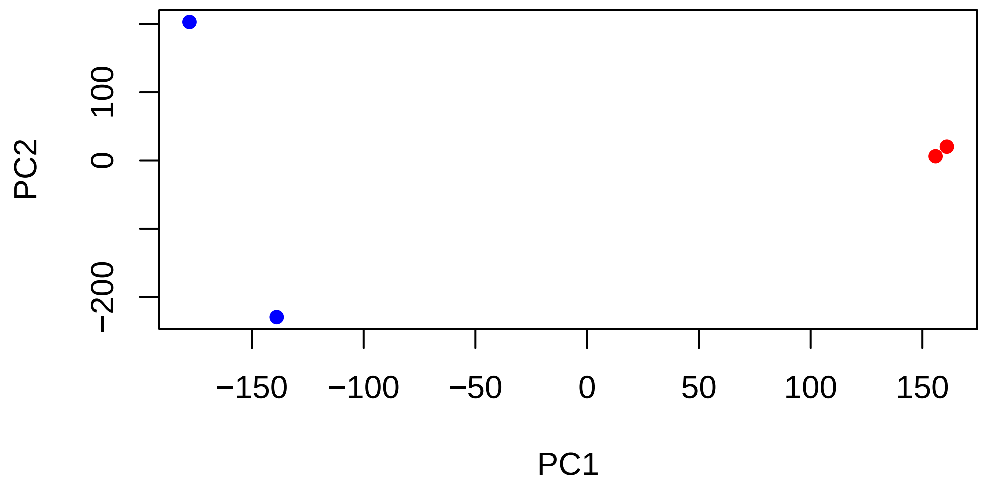
```
# Perform Principal Component Analysis (PCA)
pca <- prcomp(t(x), scale=TRUE)

# Obtain summary of PCA results
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:

```
  # Create scatter plot of the first two principal components
  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

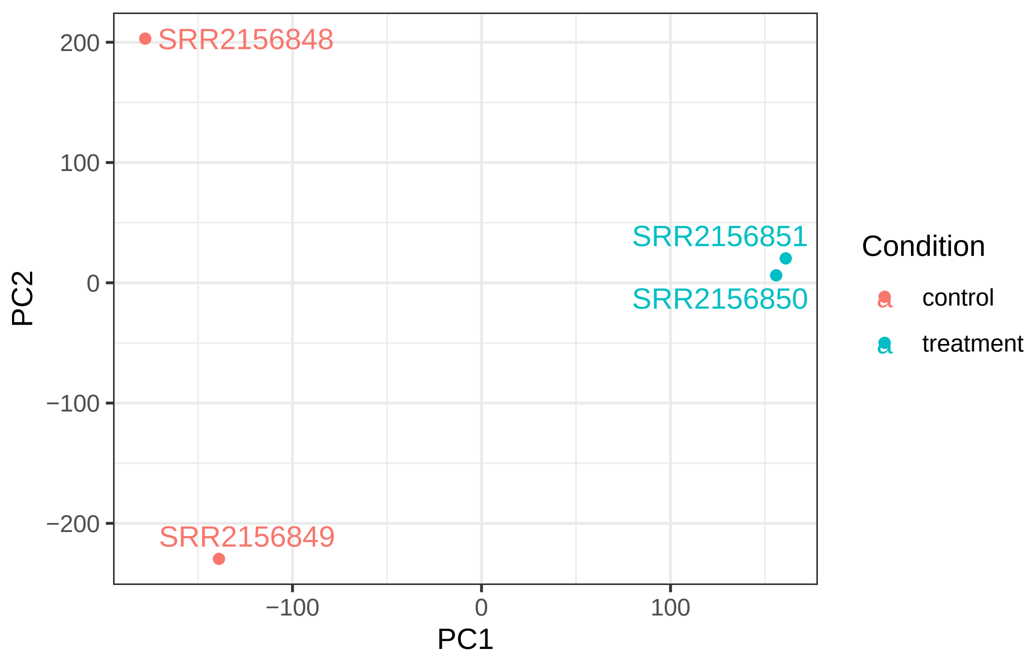
vs PC3 and PC2 vs PC3.

```
  # Load ggplot2 and ggrepel libraries
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)

# Create scatter plot with labeled data points using ggplot2
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).