Class17

Jordan Prych (A17080226)

Table of contents

Code to log into cloud and copy files back into personal comuputer

Instance ID

i-0f6c8de4006ff1785 (BIMM143_jprych) Open an SSH client.

Locate your private key file. The key used to launch this instance is bimm143_jprych.pem

Run this command, if necessary, to ensure your key is not publicly viewable. chmod 400 "bimm143_jprych.pem"

Connect to your instance using its Public DNS: ec2-54-149-116-189.us-west-2.compute.amazonaws.com

Example:

ssh -i "bimm143_jprych.pem" ubuntu@ec2-54-149-116-189.us-west-2.compute.amazonaws.com

scp -r -i "bimm143_jprych.pem" ubuntu@ec2-54-149-116-189.us-west-2.compute.amazonaws.com:~/*_quant
.

Downstream Analysis

We can now use R and Bioconductor tools to explore the dataset on our iwn computers.

Use `tximport()` to enable straightforward import of Kallisto results.

1

```
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          0
ENST00000576455          0          0          0          0
ENST00000510508          0          0          0          0
ENST00000474471          0          1          0          0
ENST00000381700          0          0          0          0
ENST00000445946          0          0          0          0
```

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

```
SRR2156848 SRR2156849 SRR2156850 SRR2156851
   2563611     586450     492477     435677
```

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

[1] 87913

Filter out abandoned transcripts with no reads:

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

Filter out those with no change over the samples:

2

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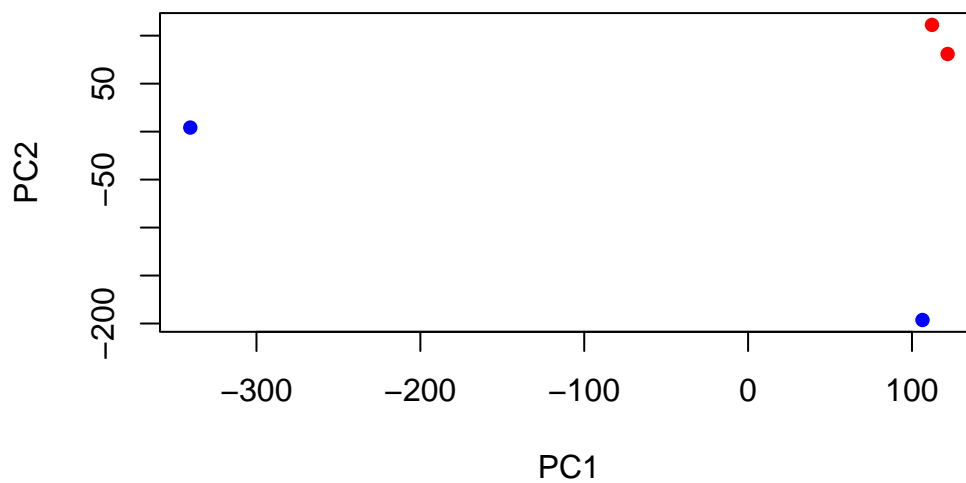
PCA

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

```
summary(pca)
```

```
Importance of components:
                          PC1      PC2      PC3       PC4
Standard deviation    227.0180 138.4195 131.1978 3.885e-11
Proportion of Variance  0.5863   0.2180   0.1958 0.000e+00
Cumulative Proportion   0.5863   0.8042   1.0000 1.000e+00
```

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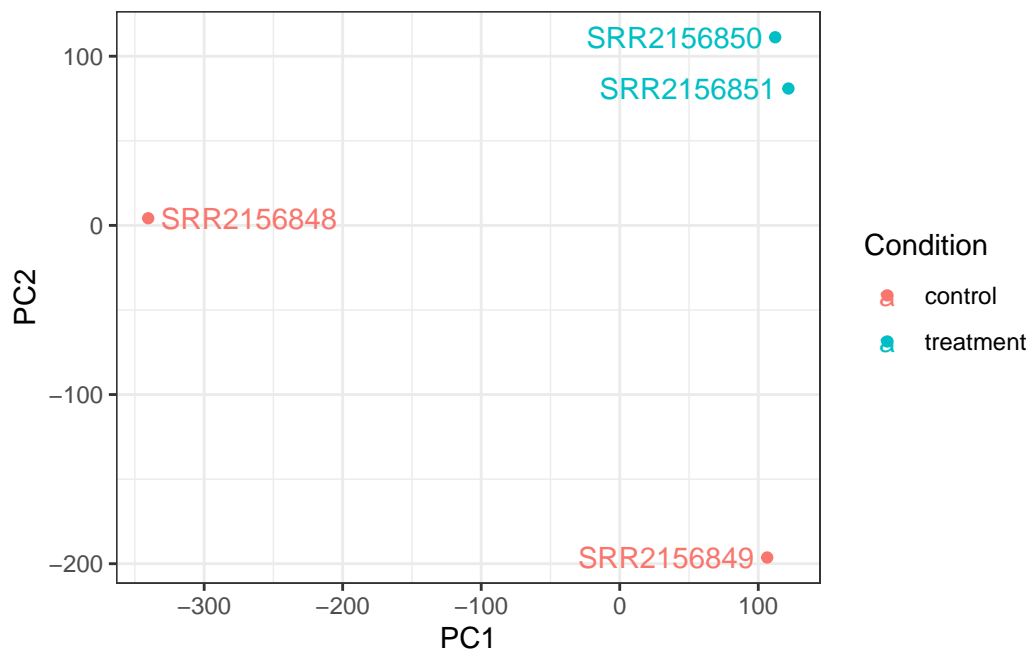
ggplot

```
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()
```
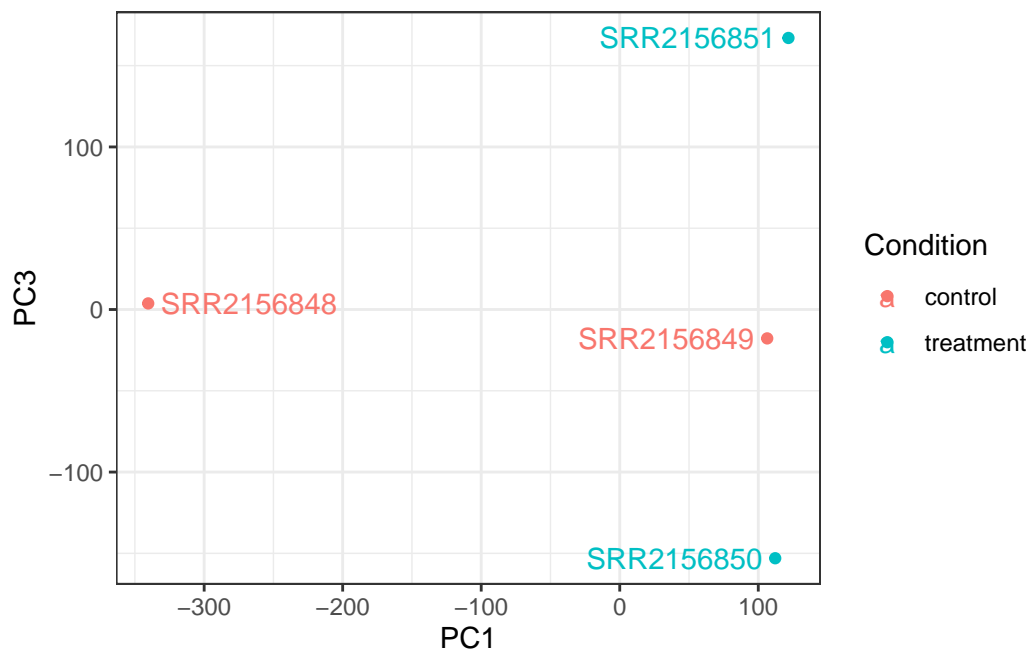


```
library(ggplot2)
library(ggrepel)

# 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, PC3, col=Condition) +
  geom_point() +
  geom_text_repel(label=rownames(y)) +
  theme_bw()
```



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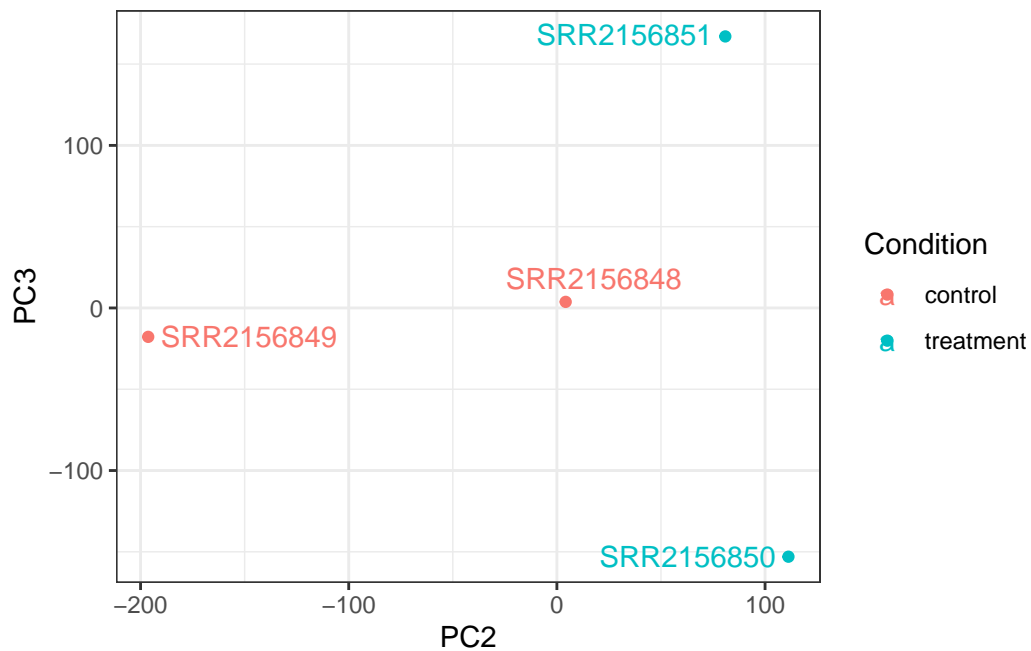

OPTIONAL: Differential-Expression Analysis

```
library(DESeq2)
```

```
Loading required package: S4Vectors

Loading required package: stats4

Loading required package: BiocGenerics


Attaching package: 'BiocGenerics'
```

The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

    anyDuplicated, aperm, append, as.data.frame, basename, cbind,
    colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
    get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
    match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
    Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
    table, tapply, union, unique, unsplit, which.max, which.min

Attaching package: 'S4Vectors'

The following object is masked from 'package:utils':

    findMatches

The following objects are masked from 'package:base':

    expand.grid, I, unname

Loading required package: IRanges

Attaching package: 'IRanges'

The following object is masked from 'package:grDevices':

    windows

Loading required package: GenomicRanges

Loading required package: GenomeInfoDb

Loading required package: SummarizedExperiment

```
Loading required package: MatrixGenerics


Loading required package: matrixStats



Attaching package: 'MatrixGenerics'


The following objects are masked from 'package:matrixStats':

    colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
    colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
    colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
    colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
    colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
    colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
    colWeightedMeans, colWeightedMedians, colWeightedSds,
    colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
    rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
    rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
    rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
    rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
    rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
    rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
    rowWeightedSds, rowWeightedVars


Loading required package: Biobase


Welcome to Bioconductor

    Vignettes contain introductory material; view with
    'browseVignettes()'. To cite Bioconductor, see
    'citation("Biobase")', and for packages 'citation("pkgname")'.



Attaching package: 'Biobase'


The following object is masked from 'package:MatrixGenerics':

    rowMedians
```

```
The following objects are masked from 'package:matrixStats':

    anyMissing, rowMedians
```

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

```r
dds <- DESeqDataSetFromTximport(txi.kallisto,
                                sampleTable,
                                ~condition)
```

```
using counts and average transcript lengths from tximport
```

```r
dds <- DESeq(dds)
```

```
estimating size factors
```

```
using 'avgTxLength' from assays(dds), correcting for library size
```

```
estimating dispersions
```

```
gene-wise dispersion estimates
```

```
mean-dispersion relationship
```

```
final dispersion estimates
```

```
fitting model and testing
```

```r
res <- results(dds)
head(res)
```

```
log2 fold change (MLE): condition treatment vs control
Wald test p-value: condition treatment vs control
DataFrame with 6 rows and 6 columns
                 baseMean log2FoldChange     lfcSE      stat    pvalue
               <numeric>      <numeric> <numeric> <numeric> <numeric>
ENST00000539570  0.000000             NA        NA        NA        NA
```

```
ENST00000576455  0.000000              NA       NA       NA       NA
ENST00000510508  0.000000              NA       NA       NA       NA
ENST00000474471  0.330126       -0.953866  4.91415 -0.194106  0.846093
ENST00000381700  0.000000              NA       NA       NA       NA
ENST00000445946  0.000000              NA       NA       NA       NA
                      padj
                 <numeric>
ENST00000539570        NA
ENST00000576455        NA
ENST00000510508        NA
ENST00000474471  0.999934
ENST00000381700        NA
ENST00000445946        NA
```