timecourse_Analysis

Protocol:

1. Use salmon to quantify transcripts

2. Use tximeta to get count matrix

3. use with DEseq for expression

import data table & create table with file paths to sample quant files

```
## importing quantifications
```

```
## reading in files with read_tsv
```

```
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30
## found matching transcriptome:
## [ Ensembl - Mus musculus - release 99 ]
## loading existing EnsDb created: 2022-04-14 17:58:52
## loading existing transcript ranges created: 2022-04-14 17:58:54
## loading existing EnsDb created: 2022-04-14 17:58:52
## obtaining transcript-to-gene mapping from database
## loading existing gene ranges created: 2022-04-14 18:01:41
## summarizing abundance
## summarizing counts
## summarizing length
```

Create DESeq dataset object with gene-level quants, prefilter out rows with zero reads

```r
#differential expression with DESeq2
dds = DESeqDataSet(gse, design = ~ SampleGenotype + SampleTime + SampleGenotype:SampleTime)
```

```
## using counts and average transcript lengths from tximeta
```

```
## Warning in DESeqDataSet(gse, design = ~SampleGenotype + SampleTime +
## SampleGenotype:SampleTime): some variables in design formula are characters,
## converting to factors
```

```r
#use if want to do transcript specific
# dds = DESeqDataSet(se, design = ~ SampleGenotype + SampleTime + SampleGenotype:SampleTime)

#get empty rows
nonzero = rowSums(counts(dds)) > 1
dds = dds[nonzero, ]

#factor so genotypes are grouped together
dds$SampleGenotype = factor(dds$SampleGenotype, levels = c("WT", "KO")) %>%
  relevel(dds$SampleGenotype, ref = "WT")
```

Run DESeq and get log2 fold changes for genotypes

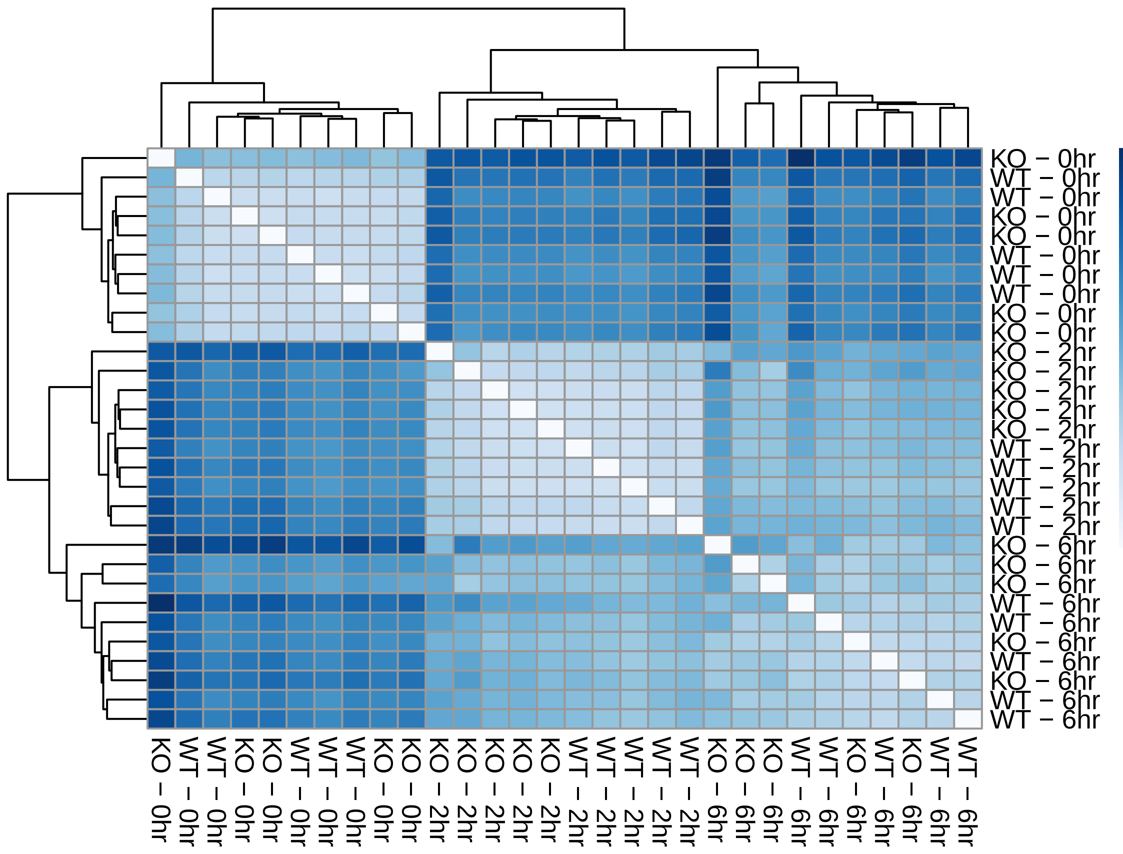
```r
dds = DESeq(dds)
```

```
## 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
#log2 fold changes and pvalues for WT vs KO
dds.res = results(dds, contrast = c("SampleGenotype", "WT", "KO"))


# count.res = dds.res
# count.res$symbol = mcols(dds)$symbol
# stim = plotCounts(dds,
#                   which.min(count.res$padj),
#                   intgroup = c("SampleGenotype", "SampleTime"), returnData = T)
# stim$hour = as.numeric(as.character(stim$SampleTime))
# ggplot(stim,
#        aes(x = SampleTime, y = count, color = SampleGenotype, group = SampleGenotype)) + geom_point()
#   stat_summary(fun = mean, geom = "line") + scale_y_log10()
```

Annotating dataframe with gene symbols

QC: Sample Distances

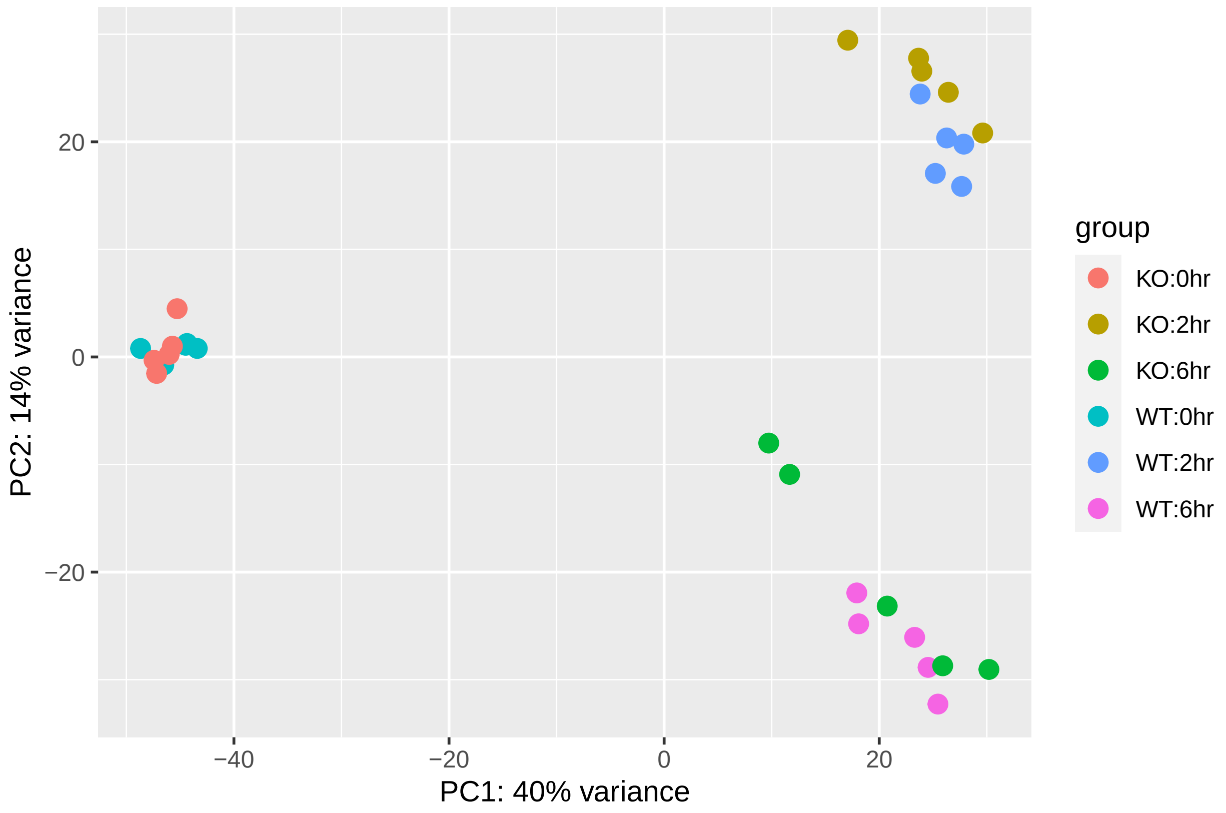


Visualizing sample distances

QC: PCA plots

PCA plot (all time points)

```
# library(vsn)

#variance stabilizing transformation on non-normalized counts
vsd = vst(dds, blind = T)

#use ntop to get all rows of matrix (default is 500)
plotPCA(vsd, intgroup = c("SampleGenotype", "SampleTime"), ntop = nrow(vsd))
```

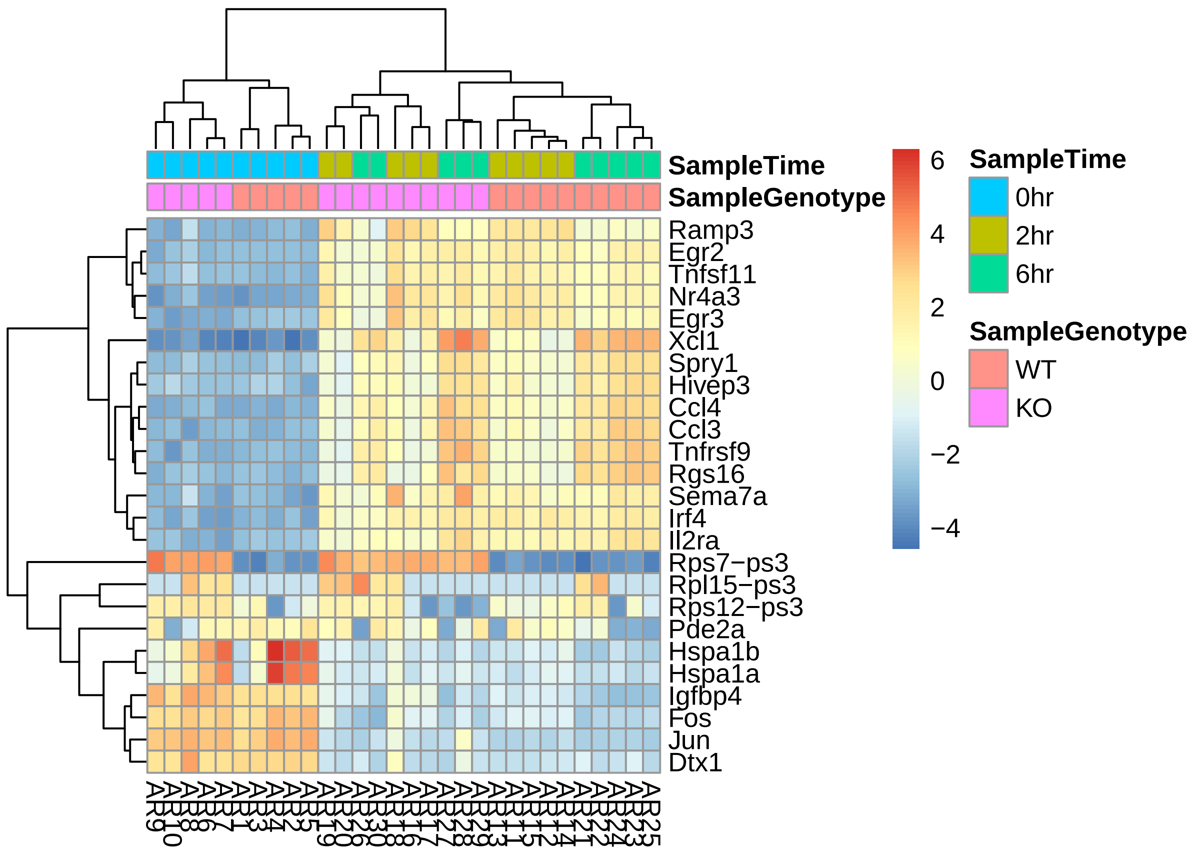


Gene cluster heatmap

```
# library(genefilter)


topVarGenes = head(order(rowVars(assay(vsd)), decreasing = T), 25)
mat = assay(vsd)[ topVarGenes, ]



rownames(mat) = mapIds(org.Mm.eg.db,
                    keys = rownames(mat),
                    column = "SYMBOL",
                    keytype = "ENSEMBL",
                    multiVals = "first")
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
# # which genes are NA?
# unmatched_ENSEMBL = rownames(mat)[which(is.na(rownames(mat)), arr.ind = F)]
#
# #look like pseudogenes; remove
# mat = mat[-which(is.na(rownames(mat))),]

mat = mat - rowMeans(mat)
anno = as.data.frame(colData(vsd)[, c("SampleGenotype", "SampleTime")])

pheatmap::pheatmap(mat, annotation_col = anno)
```



```r
# png(file = "/active/allenspach_e/AllenspachRNASeqData/2022_SH2B3_TcellSeq/src/RNAseq/h1eatmap.png", w
# png('heatmap_plot.png', type = 'cairo')
# p
# print(p)
# while (!is.null(dev.list()))  dev.off()
```

Processing data as time course: Start with likelihood ratio test to remove genotype specific differences - the remaining genes with small p values showed genotype specific effects after time 0hr.

```r
# library(tidyverse)
ddsTC = DESeqDataSet(gse, design = ~ SampleGenotype + SampleTime + SampleGenotype:SampleTime)
```

```
## using counts and average transcript lengths from tximeta

## Warning in DESeqDataSet(gse, design = ~SampleGenotype + SampleTime +
## SampleGenotype:SampleTime): some variables in design formula are characters,
## converting to factors
```

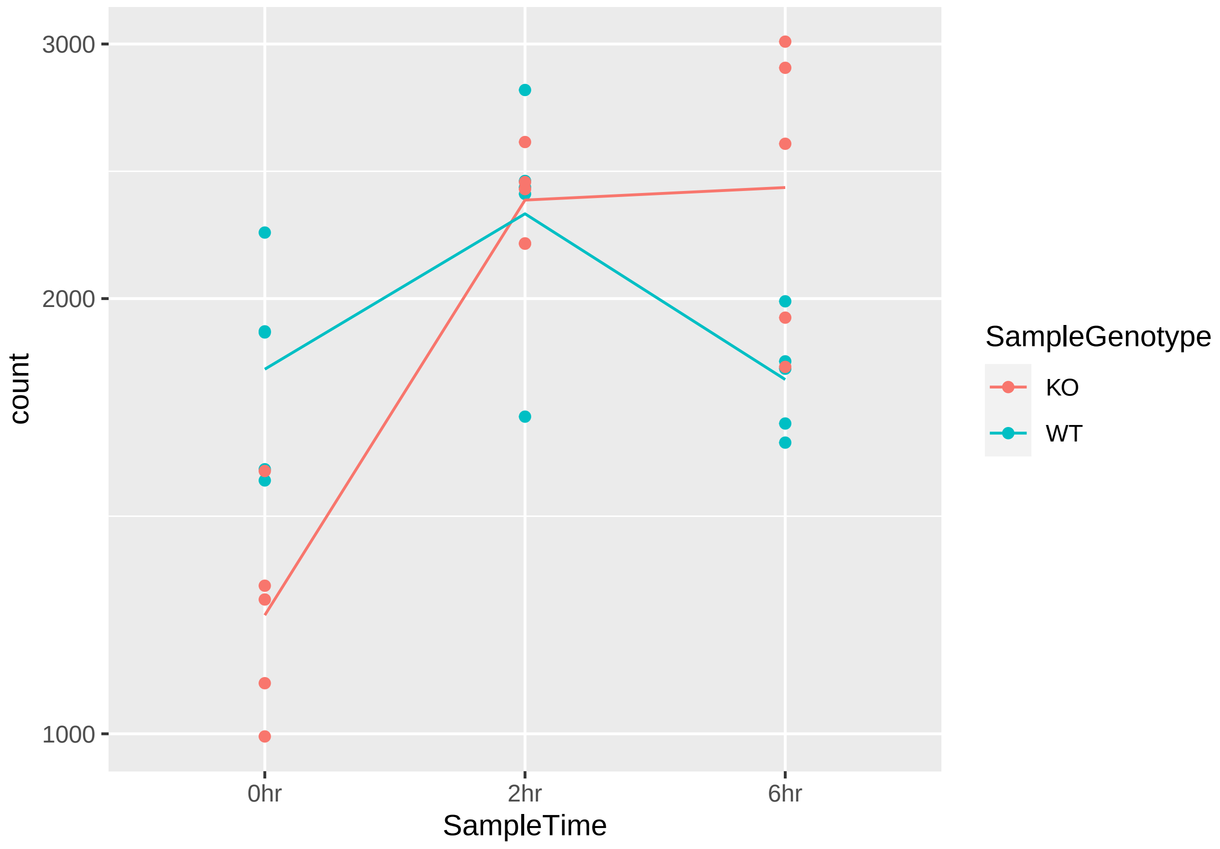
```r
#likelyhood ratio test
ddsTC = DESeq(ddsTC, test="LRT", reduced = ~ SampleGenotype + SampleTime)
```

```
## 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
```

```
resTC = results(ddsTC)
resTC$symbol = mcols(ddsTC)$symbol

stim = plotCounts(ddsTC, which.min(resTC$padj),
                  intgroup = c("SampleGenotype", "SampleTime"), returnData = T)
stim$hour = as.numeric(as.character(stim$SampleTime))
```

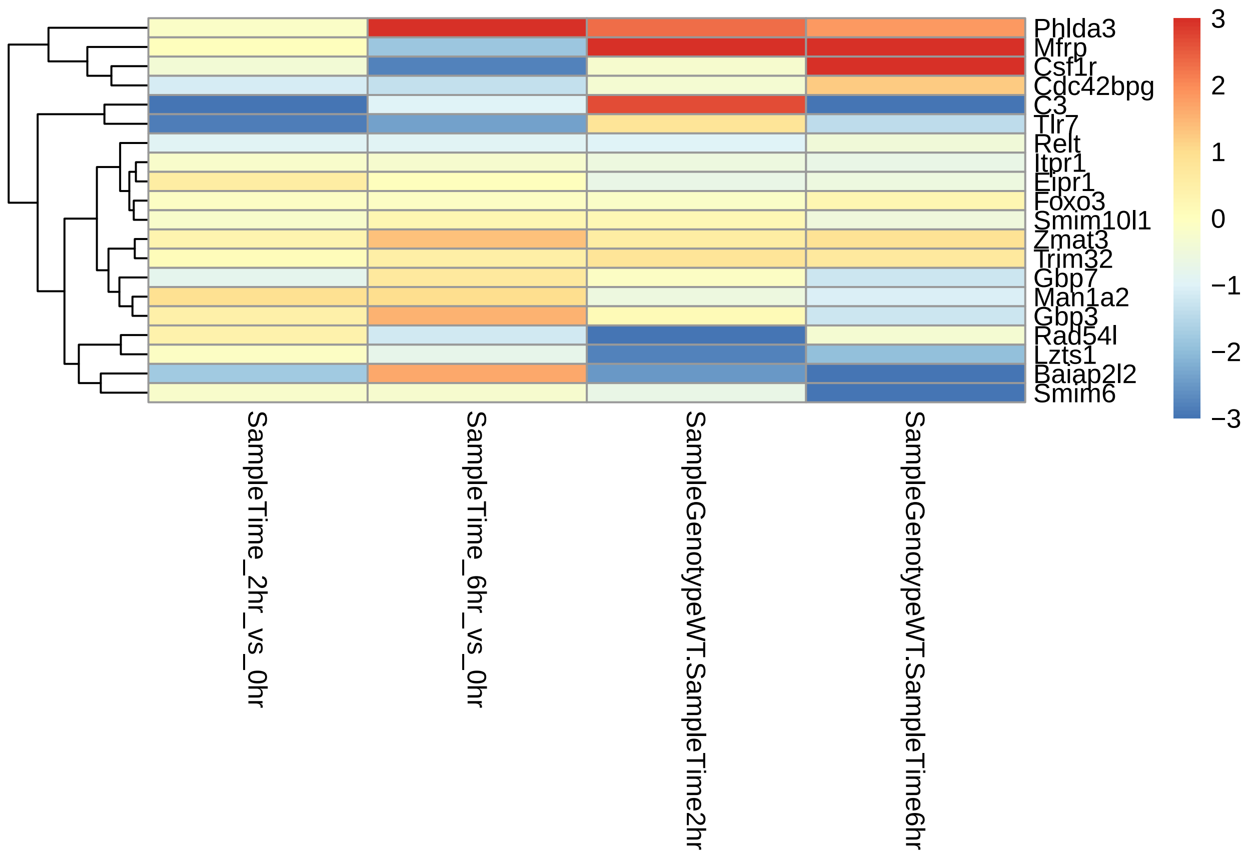
```
## Warning: NAs introduced by coercion
```

```
ggplot(stim,
       aes(x = SampleTime, y = count, color = SampleGenotype, group = SampleGenotype)) + geom_point() +
  stat_summary(fun = mean, geom = "line") + scale_y_log10()
```



Cluster significant genes by profile

```
## 'select()' returned 1:1 mapping between keys and columns
```



Phlda3
Mfrp
Csf1r
Cdc42bpg
C3
Tlr7
Relt
Itpr1
Eipr1
Foxo3
Smim10l1
Zmat3
Trim32
Gbp7
Man1a2
Gbp3
Rad54l
Lzts1
Baiap2l2
Smim6
3
2
1
0
−1
−2
−3
SampleTime_2hr_vs_0hr
SampleTime_6hr_vs_0hr
SampleGenotypeWT.SampleTime2hr
SampleGenotypeWT.SampleTime6hr