initialAnalysis

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

```
data_table = read.delim("data_table")

txdata = data_table %>% dplyr::select(c(SampleName, SampleGenotype, SampleTime, QuantFile)) %>%
  dplyr::rename(names = SampleName, files = QuantFile)

dirPath = "/active/allenspach_e/AllenspachRNASeqData/2022_SH2B3_TcellSeq/data/"

#rename files column to include full path

txdata = txdata %>% mutate(files = paste(dirPath, files, sep = ""))

#import quantifications
se = tximeta(data.frame(as_tibble(txdata)))
```

```
## 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
```

```
#get quants at gene level
gse = summarizeToGene(se)
```

```
## 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
```

```
# analyzedPath = "/active/allenspach_e/AllenspachRNASeqData/2022_SH2B3_TcellSeq/analyzed_data/"
#
# #save datasets as RDS
# saveRDS(gse, paste(analyzedPath,"gse.rds", sep = ""))
# saveRDS(se, paste(analyzedPath,"se.rds", sep = ""))
```

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

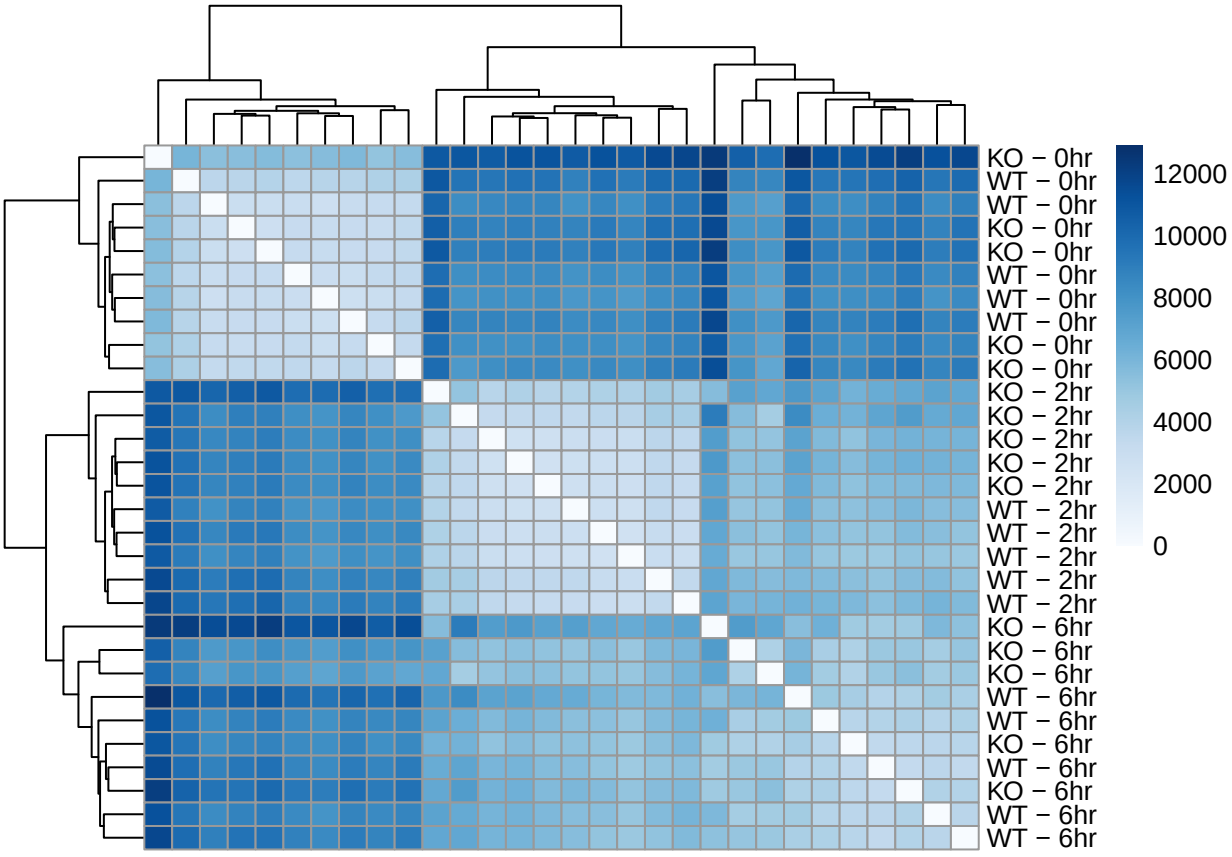
```
## using counts and average transcript lengths from tximeta
```

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

```
#filter out empty rows

#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")
```




```
library(vsn)

#variance stabilizing transformation
vsd = vst(dds)
```

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

```
plotPCA(vsd, intgroup = c("SampleGenotype", "SampleTime"))
```