Exploring Sample Relationships in R

2024-03-19

R Markdown

In this tutorial we are working with mouse mammary tissue at three developmental stages: virgin, pregnant and lactating.

```
library(pheatmap)
library(RColorBrewer)
library(BiocManager)
```

```
## Bioconductor version '3.16' is out-of-date; the current release version '3.18'
##   is available with R version '4.3'; see https://bioconductor.org/install
```

```
BiocManager::install("org.Mm.eg.db")
```

```
## Bioconductor version 3.16 (BiocManager 1.30.22), R 4.2.3 (2023-03-15)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'org.Mm.eg.db'
```

```
## Old packages: 'curl', 'data.table', 'dbplyr', 'deldir', 'digest', 'ggrepel',
##   'glue', 'igraph', 'locfit', 'MASS', 'Matrix', 'pkgbuild', 'processx', 'ps',
##   'ragg', 'Rcpp', 'RcppAnnoy', 'RcppEigen', 'RcppHNSW', 'RCurl', 'readr',
##   'remotes', 'reticulate', 'rlang', 'RSQLite', 'sass', 'Seurat', 'sp',
##   'spatstat.explore', 'spatstat.geom', 'spatstat.random', 'survival',
##   'systemfonts', 'tidyr', 'timechange', 'tinytex', 'uuid', 'xfun', 'XML'
```

```
BiocManager::install("DESeq2")
```

```
## Bioconductor version 3.16 (BiocManager 1.30.22), R 4.2.3 (2023-03-15)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'DESeq2'
```

```
## Old packages: 'curl', 'data.table', 'dbplyr', 'deldir', 'digest', 'ggrepel',
##   'glue', 'igraph', 'locfit', 'MASS', 'Matrix', 'pkgbuild', 'processx', 'ps',
##   'ragg', 'Rcpp', 'RcppAnnoy', 'RcppEigen', 'RcppHNSW', 'RCurl', 'readr',
##   'remotes', 'reticulate', 'rlang', 'RSQLite', 'sass', 'Seurat', 'sp',
##   'spatstat.explore', 'spatstat.geom', 'spatstat.random', 'survival',
##   'systemfonts', 'tidyr', 'timechange', 'tinytex', 'uuid', 'xfun', 'XML'
```

```
BiocManager::install("genefilter")
```

```
## Bioconductor version 3.16 (BiocManager 1.30.22), R 4.2.3 (2023-03-15)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   'force = TRUE' to re-install: 'genefilter'

## Old packages: 'curl', 'data.table', 'dbplyr', 'deldir', 'digest', 'ggrepel',
##   'glue', 'igraph', 'locfit', 'MASS', 'Matrix', 'pkgbuild', 'processx', 'ps',
##   'ragg', 'Rcpp', 'RcppAnnoy', 'RcppEigen', 'RcppHNSW', 'RCurl', 'readr',
##   'remotes', 'reticulate', 'rlang', 'RSQLite', 'sass', 'Seurat', 'sp',
##   'spatstat.explore', 'spatstat.geom', 'spatstat.random', 'survival',
##   'systemfonts', 'tidyr', 'timechange', 'tinytex', 'uuid', 'xfun', 'XML'
```

```
library(org.Mm.eg.db)
```

```
## Loading required package: AnnotationDbi

## 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, setdiff, sort,
##     table, tapply, union, unique, unsplit, which.max, which.min

## 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")'.

## Loading required package: IRanges

## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

##
```

```
library(DESeq2)
```

```
## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: MatrixGenerics

## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians

##
## 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

## The following object is masked from 'package:Biobase':
##
##     rowMedians
```

```
library(genefilter)
```

```
##
## Attaching package: 'genefilter'

## The following objects are masked from 'package:MatrixGenerics':
##
##     rowSds, rowVars

## The following objects are masked from 'package:matrixStats':
##
##     rowSds, rowVars
```

Show counts, exp_counts and metadata object

```
##
## 1 new("standardGeneric", .Data = function (object, ...)
## 2 standardGeneric("counts"), generic = structure("counts", package = "BiocGenerics"),
## 3     package = "BiocGenerics", group = list(), valueClass = character(0),
## 4     signature = "object", default = NULL, skeleton = (function (object,
## 5         ...)
## 6     stop(gettextf("invalid call in method dispatch to '%s' (no default method)",

## [1] 22943

##                       cell_type dev_stage replicate
## GSM1480291_LVirgin          lum    virgin         A
## GSM1480292_LVirgin          lum    virgin         B
## GSM1480293_LPregnant        lum      preg         A
## GSM1480294_LPregnant        lum      preg         B
## GSM1480295_LLactate         lum      lact         A
## GSM1480296_LLactate         lum      lact         B
## GSM1480297_BVirgin        basal    virgin         A
## GSM1480298_BVirgin        basal    virgin         B
## GSM1480299_BPregnant      basal      preg         A
## GSM1480300_BPregnant      basal      preg         B
## GSM1480301_BLactate       basal      lact         A
## GSM1480302_BLactate       basal      lact         B
```

```
#use org.Mm.eg.db to create a two-column table called "mapping" that takes the Ensembl identifiers in es
mapping <-select(org.Mm.eg.db, as.character(exp_counts$ENSEMBL), keytype = "ENSEMBL", column="SYMBOL")
```

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

```
head(mapping)
```

```
##              ENSEMBL  SYMBOL
## 1 ENSMUSG00000061937 Csn1s2a
## 2 ENSMUSG00000063157    Csn2
## 3 ENSMUSG00000070702  Csn1s1
```

```
## 4 ENSMUSG00000022491 Glycam1
## 5 ENSMUSG00000064351    COX1
## 6 ENSMUSG00000032554     Trf
```

```
nrow(mapping)
```

```
## [1] 23191
```

```
#Print the number of duplicate rows
d <- duplicated(mapping$ENSEMBL)
sum(d)
```

```
## [1] 248
```

```
mapping <- mapping[!d,]
nrow(mapping)
```

```
## [1] 22943
```

```
#merge expression counts and mapping
exp_counts <- merge(exp_counts, mapping, by = "ENSEMBL")
head(exp_counts)
```

```
##              ENSEMBL GSM1480291_LVirgin GSM1480292_LVirgin GSM1480293_LPregnant
## 1 ENSMUSG00000000001               5450               6087                5772
## 2 ENSMUSG00000000003                  0                  0                   0
## 3 ENSMUSG00000000028                260                330                 267
## 4 ENSMUSG00000000031                135                215                 172
## 5 ENSMUSG00000000037                 66                 97                  65
## 6 ENSMUSG00000000049                  0                  3                   0
##   GSM1480294_LPregnant GSM1480295_LLactate GSM1480296_LLactate
## 1                 5168                2192                2192
## 2                    0                   0                   0
## 3                  102                 232                 207
## 4                  259                   3                   0
## 5                   40                  24                  22
## 6                    0                   0                   0
##   GSM1480297_BVirgin GSM1480298_BVirgin GSM1480299_BPregnant
## 1               4354               4380                4554
## 2                  0                  0                   0
## 3                306                261                 199
## 4                 25                 24                  33
## 5                180                161                 230
## 6                  3                  6                   1
##   GSM1480300_BPregnant GSM1480301_BLactate GSM1480302_BLactate SYMBOL
## 1                 4080                3553                2799  Gnai3
## 2                    0                   0                   0   Pbsn
## 3                  152                  44                  64  Cdc45
## 4                   20                 262                 128    H19
## 5                  197                 238                 188  Scml2
## 6                    1                   2                   2   Apoh
```

```
#Remove rows where ENSEMBL did not map to a gene symbol
missing <- is.na(exp_counts$SYMBOL)
exp_counts <- exp_counts[!missing,]
nrow(exp_counts)
```

```
## [1] 22621
```

```
#Remove duplicate gene symbols
o <- order(rowSums(exp_counts[,c(2:13)]), decreasing=TRUE)
exp_counts <- exp_counts[o,]
d2 <- duplicated(exp_counts$SYMBOL)
exp_counts <- exp_counts[!d2,]
```

```
sum(d2)
```

```
## [1] 8
```

```
nrow(exp_counts)
```

```
## [1] 22613
```

```
row.names(exp_counts) <- exp_counts$SYMBOL
exp_counts$SYMBOL <- NULL
exp_counts$ENSEMBL <- NULL
head(exp_counts)
```

```
##         GSM1480291_LVirgin GSM1480292_LVirgin GSM1480293_LPregnant
## Csn1s2a              12323              15333              1187699
## Csn2                 15121              25594              1000240
## Csn1s1               39203              54096               801556
## Glycam1               1467               1630               307878
## COX1                290125             299519               408228
## Trf                 133375             130927               533306
##         GSM1480294_LPregnant GSM1480295_LLactate GSM1480296_LLactate
## Csn1s2a              1326863             2887969             2816350
## Csn2                 1088278             2575890             2501120
## Csn1s1                871795             2261418             2254038
## Glycam1               428510             1622370             1492270
## COX1                  411393              287889              307488
## Trf                   524878             1010943              978723
##         GSM1480297_BVirgin GSM1480298_BVirgin GSM1480299_BPregnant
## Csn1s2a                169                246                48802
## Csn2                   228                445                40087
## Csn1s1                 597                945                32489
## Glycam1                 89                 82                13144
## COX1                273892             264027               268185
## Trf                   6382               8747                28811
##         GSM1480300_BPregnant GSM1480301_BLactate GSM1480302_BLactate
## Csn1s2a                29363              28626              24203
## Csn2                   23462              22364              21742
```

```
## Csn1s1                 18427              21147              19270
## Glycam1                 9795              16017              14527
## COX1                  271967             241355             226391
## Trf                    20364              11039              10181
```

```
#Create deseq dataset
data_deseq <- DESeqDataSetFromMatrix(countData = exp_counts, colData = sample_data, design = ~ 1)
head(counts(data_deseq))
```

```
##         GSM1480291_LVirgin GSM1480292_LVirgin GSM1480293_LPregnant
## Csn1s2a              12323              15333              1187699
## Csn2                 15121              25594              1000240
## Csn1s1               39203              54096               801556
## Glycam1               1467               1630               307878
## COX1                290125             299519               408228
## Trf                 133375             130927               533306
##         GSM1480294_LPregnant GSM1480295_LLactate GSM1480296_LLactate
## Csn1s2a              1326863             2887969             2816350
## Csn2                 1088278             2575890             2501120
## Csn1s1                871795             2261418             2254038
## Glycam1               428510             1622370             1492270
## COX1                  411393              287889              307488
## Trf                   524878             1010943              978723
##         GSM1480297_BVirgin GSM1480298_BVirgin GSM1480299_BPregnant
## Csn1s2a                169                246                48802
## Csn2                   228                445                40087
## Csn1s1                 597                945                32489
## Glycam1                 89                 82                13144
## COX1                273892             264027               268185
## Trf                   6382               8747                28811
##         GSM1480300_BPregnant GSM1480301_BLactate GSM1480302_BLactate
## Csn1s2a                29363              28626              24203
## Csn2                   23462              22364              21742
## Csn1s1                 18427              21147              19270
## Glycam1                 9795              16017              14527
## COX1                  271967             241355             226391
## Trf                    20364              11039              10181
```

```
nrow(data_deseq)
```

```
## [1] 22613
```

```
data_deseq <- data_deseq[ rowSums(counts(data_deseq)) > 1, ]
rld <- rlog(data_deseq, blind=FALSE)
```

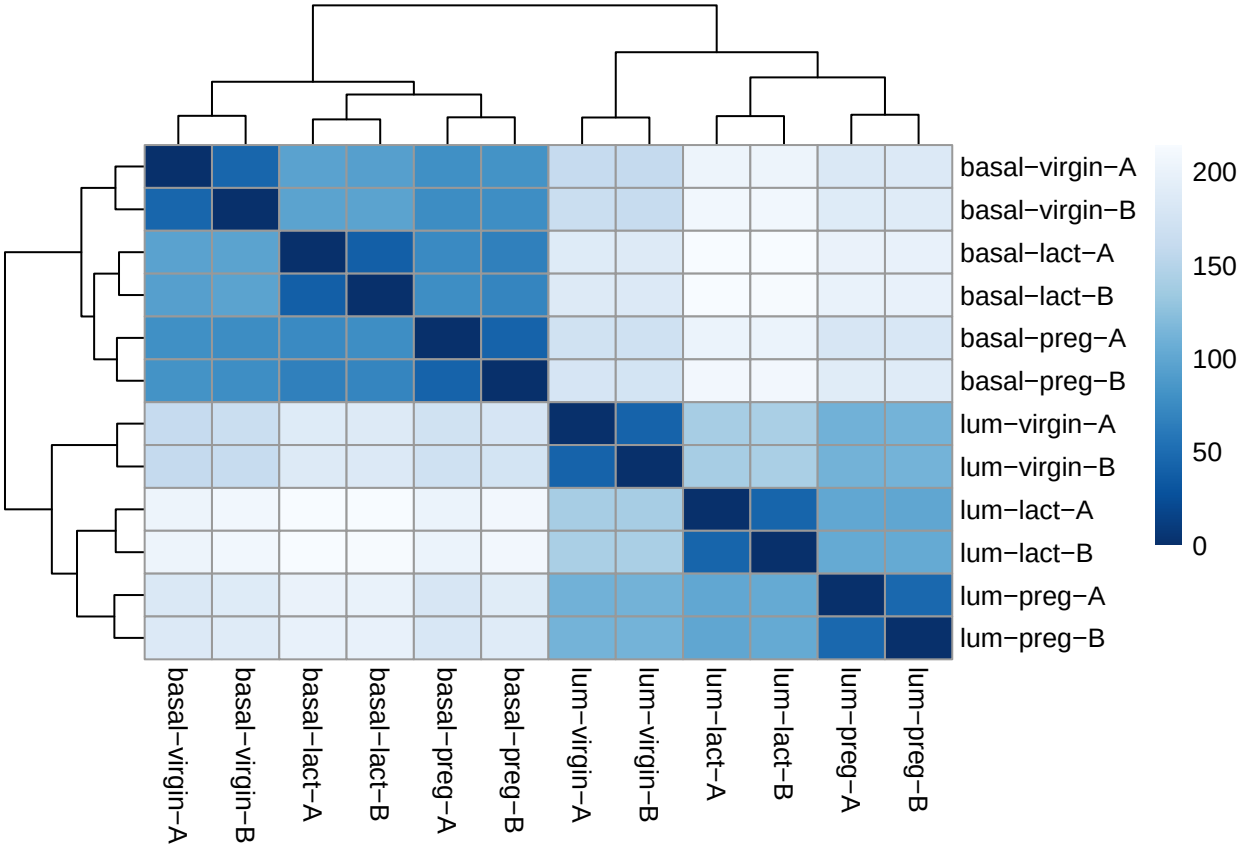
```
sampleDists <- dist(t(assay(rld)))
sampleDists
```

```
##                       GSM1480291_LVirgin GSM1480292_LVirgin GSM1480293_LPregnant
## GSM1480292_LVirgin             42.24451
## GSM1480293_LPregnant          110.69524          111.26442
```

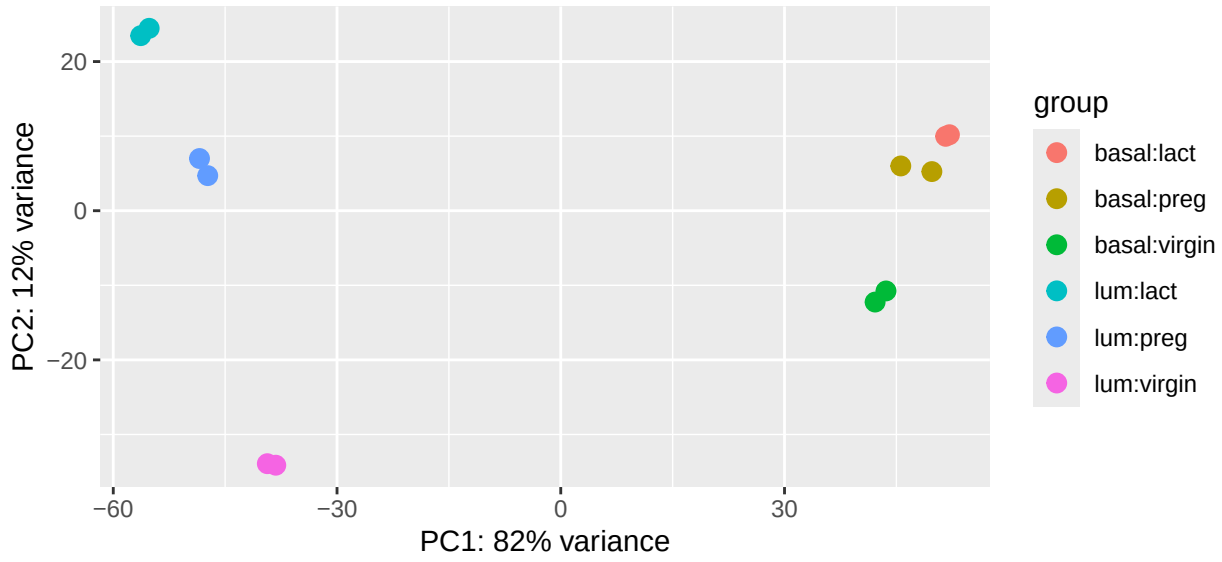
```
## GSM1480294_LPregnant          112.08970          112.08671           46.47887
## GSM1480295_LLactate           139.98438          140.19656          101.10293
## GSM1480296_LLactate           142.18877          142.34460          103.75387
## GSM1480297_BVirgin            161.14177          159.95337          183.85925
## GSM1480298_BVirgin            165.53151          162.87384          187.80455
## GSM1480299_BPregnant          172.24409          170.61001          180.25705
## GSM1480300_BPregnant          178.11699          176.19416          189.34646
## GSM1480301_BLactate           187.63664          186.40141          200.14792
## GSM1480302_BLactate           186.52452          185.32965          199.91993
##                       GSM1480294_LPregnant GSM1480295_LLactate
## GSM1480292_LVirgin
## GSM1480293_LPregnant
## GSM1480294_LPregnant
## GSM1480295_LLactate             100.69408
## GSM1480296_LLactate             103.55123           44.38443
## GSM1480297_BVirgin              184.84766          203.92468
## GSM1480298_BVirgin              188.22946          207.53578
## GSM1480299_BPregnant            181.41861          201.60565
## GSM1480300_BPregnant            188.65420          208.48975
## GSM1480301_BLactate             198.54883          214.22082
## GSM1480302_BLactate             198.54890          213.60791
##                       GSM1480296_LLactate GSM1480297_BVirgin GSM1480298_BVirgin
## GSM1480292_LVirgin
## GSM1480293_LPregnant
## GSM1480294_LPregnant
## GSM1480295_LLactate
## GSM1480296_LLactate
## GSM1480297_BVirgin              204.09069
## GSM1480298_BVirgin              207.99783           44.82520
## GSM1480299_BPregnant            202.01916           79.67973           76.77562
## GSM1480300_BPregnant            208.83402           81.80897           77.49947
## GSM1480301_BLactate             214.01731           96.14490           97.06044
## GSM1480302_BLactate             213.05181           94.08350           96.83213
##                       GSM1480299_BPregnant GSM1480300_BPregnant
## GSM1480292_LVirgin
## GSM1480293_LPregnant
## GSM1480294_LPregnant
## GSM1480295_LLactate
## GSM1480296_LLactate
## GSM1480297_BVirgin
## GSM1480298_BVirgin
## GSM1480299_BPregnant
## GSM1480300_BPregnant              42.54994
## GSM1480301_BLactate               73.96654             65.53985
## GSM1480302_BLactate               77.48158             69.83525
##                       GSM1480301_BLactate
## GSM1480292_LVirgin
## GSM1480293_LPregnant
## GSM1480294_LPregnant
## GSM1480295_LLactate
## GSM1480296_LLactate
## GSM1480297_BVirgin
## GSM1480298_BVirgin
## GSM1480299_BPregnant
```

```
## GSM1480300_BPregnant
## GSM1480301_BLactate
## GSM1480302_BLactate             39.34284
```

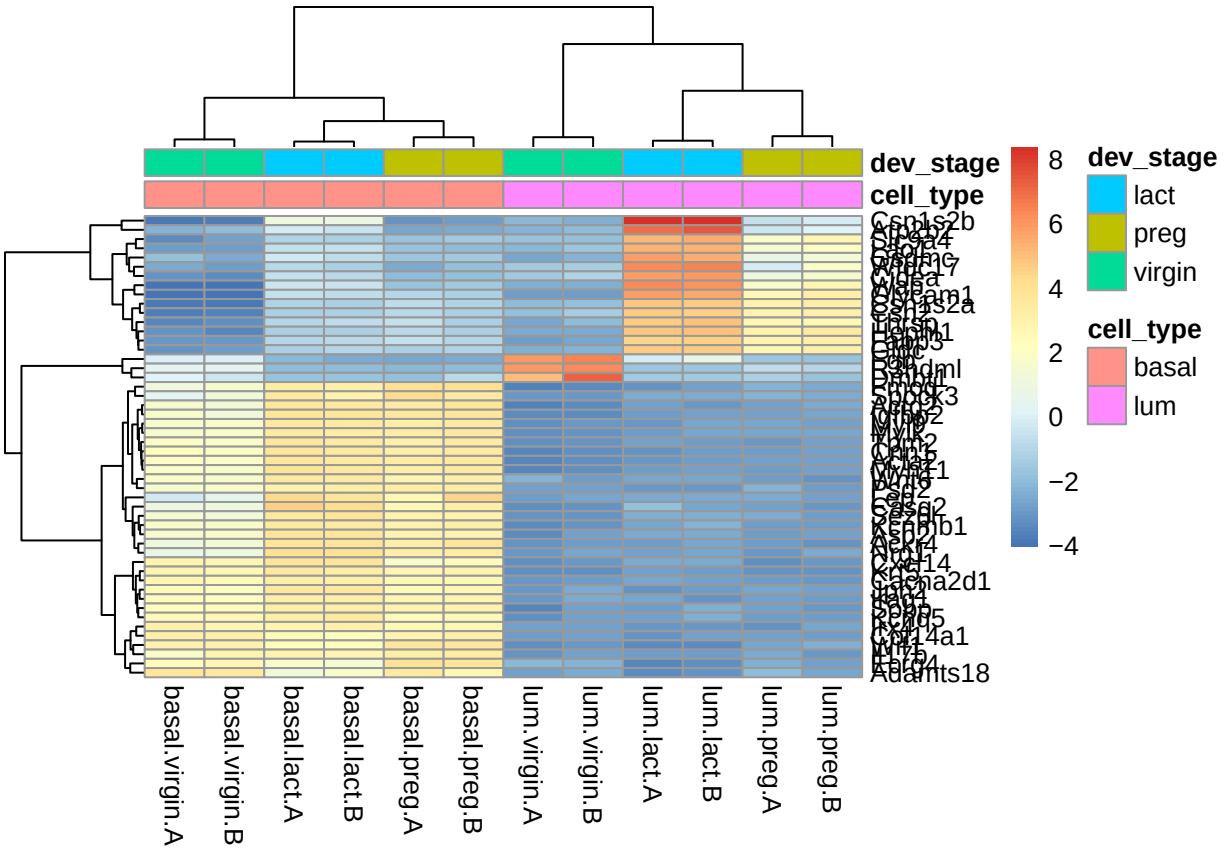
```
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste( rld$cell_type, rld$dev_stage, rld$replicate, sep="-" )
DistMatrix <- as.matrix(sampleDists)
colnames(sampleDistMatrix) <- paste( rld$cell_type, rld$dev_stage, rld$replicate, sep="-" )
colors <- colorRampPalette(rev(brewer.pal(9, "Blues")))(255)
pheatmap(sampleDistMatrix, clustering_distance_rows=sampleDists, clustering_distance_cols=sampleDists,
```




```
plotPCA(rld, intgroup = c("cell_type", "dev_stage"))
```




```
geneVars <- rowVars(assay(rld))
geneVarsOrdered <- order(geneVars, decreasing = TRUE)
topVarGenes <- head(geneVarsOrdered, 50)
mat <- assay(rld)[ topVarGenes, ]
mat <- mat - rowMeans(mat)
df <- as.data.frame(colData(rld)[,c("cell_type","dev_stage")])
clear_col_names <- paste( rld$cell_type, rld$dev_stage, rld$replicate, sep=".")
topGenesHeatmap <- pheatmap(mat, annotation_col=df, labels_col = clear_col_names)
```



1. ID Mapping

a. What was the overall goal of the ID mapping section? In this step, we are cleaning up the data. By converting each gene into from one type to another, we maintain readability and consistency of ID.

b. Describe at least two steps of the ID mapping where information was lost. Two steps that removed information after mapping: 1) removing duplicates and removing duplicates after the first duplicated item 2) ENSEMBL failed to map and we removed rows with missing values.

2. In this tutorial, we filtered the counts table to remove rows where the total counts were less than 1. However, there are many ways to filter the counts to retain "more interesting" genes. Name a different filtering criterion, explain why you chose it, and show the R command that you would apply to data_deseq to filter according to this criterion. data_deseq_1 <- data_deseq[rowSums(counts(data_deseq)) < 1,] data_deseq <- data_deseq_1[!data_deseq_1,]

3. Sample distance heatmap

a. Show an image of your sample distance heatmap. (look above)

b. Discuss the relationships among the samples as seen in the heatmap. Which samples are more closely related? Which are less similar? Does this correspond to what you would expect based on the biology of the experiment? Explain. Luminal cell tissue is more closely related to luminal cell tissue than basal cell tissue samples. Basal cell tissue is more closely related to basal cell tissue than luminal cell tissue. Each tissue is closely related by developmental stage. You can identify these relationships through chared cohorts in the dendrogram.

4. PCA plot

a. Show an image of your PCA plot.

b. What percent of the variance is captured by: PC1 = 81% PC2 = 12%

c. PC1 separates the samples into two major groups. What sample characteristic distinguishes these groups? It looks as though PC1 separates the samples into basal and luminal groups.

d. What sample characteristic is mainly distinguished by PC2? PC2 distinguishes developmental stages.

e. Do the replicates agree well with each other? I believe that the samples agree well with each other. The pregnant basal cells seems to have a bit more variance between the replicates compared to other sample replicates.

5. Gene heatmap

a. Provide an image of your heatmap.

b. Look at the clustering of the samples (columns of the heatmap). The tree diagram has four levels of branching. Which samples are separated from each other at the first branching level of the tree? The second branching level? The third branching level? The fourth level? The fourth level separates into different cell types: basal and luminal. The third level separates into virgin as one group and pregnant and lactating as another group. The second level separates lactating ang pregnant cells. Developmental stages are separated at the first level of the tree.

c. Look at the clustering of the genes (rows of the heatmap). The first branch of the tree diagram separates the genes into two clusters–a larger cluster of ~35 genes and a smaller cluster of ~15 genes. For the genes in the larger cluster, what samples show high expression of these genes? Low expression? For the genes in the larger cluster, we mainly see luminal cells showing low expression (with the exception of virgin luminal cells with high expression) and basal cells showing higher expression.

For the genes in the smaller cluster, what samples show the highest expression? The lowest expression? Intermediate levels of expression? For the genes in the smaller cluster, we mainly see virgin luminal cells and all basal cells showing low expression. Lactating luminal cells and show high expression and pregnant luminal cells show moderate expression.

d. Pick one gene from each of the two clusters in the heatmap and look up their function in bioinformatics resources such as UniProtKB or GeneCards. Based on the functional information you find, is the expression pattern of these genes in the different cell types and developmental stages what you expect? Explain. I chose Dmbt1 (Deleted In Malignant Brain Tumors 1) because it is expressed more in virgin luminal cells and downregulated in every other cell sample. According to UniProtKB, this gene is involved in providing mucosal and cellular immunity and is expressed highly from 18.5 dpc (days post coitum) to birth and gradually decreases as the mouse enters into adulthood. Therefore, it makes sense that this gene is upregulated in virgin mice. Yet, we do not see the same expression pattern in virgin basal cells. I expect this is due to difference in the cellular functions. The next gene I chose was ACTG2 (Actin, gamma-enteric smooth muscle) which is mainly expressed in the basal cells. According to UniProt, this gene is expressed in smooth muscle.

Note that mammary basal cells are also called myoepithelial cells. They resemble smooth muscle cells and secrete proteins that make up the basement membrane, a type of extracellular matrix. The luminal cells are responsible for making and secreting milk during lactation. A good paper discussing the role of mammary cell types can be found here: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3193434/.