HW8

Tianqi Wu

4/2/2020

```
library(tidyverse)
library(edgeR)
library(glmnet)
library(pheatmap)
library(NMF)
library(factoextra)
library(MASS)
library(umap)
library(dbscan)
load("gtex_subset.RData") ## replace path with wherever you stored your file
expr = cpm(gtex_subset, log = TRUE)
expr = t(expr)
```

Problem 1

1(a)

For lasso model, lambda.min has 20 nonzero coeffcients. It means that 20 genes are likely useful for discriminating between tissues.

```
set.seed(123)
# Remove zero variance
removed = c()
for (i in 1:ncol(expr)){
  if (var(expr[, i]) == 0){
    print(i)
    removed = append(removed, colnames(expr)[i])
  }
}
```

```
# Lasso
lasso <- cv.glmnet(x = expr,
                   y = tissue,
                   nfolds = 3,
                   family = "binomial")

print(lasso)
```

```
##
## Call:  cv.glmnet(x = expr, y = tissue, nfolds = 3, family = "binomial")
##
```

```
## Measure: Binomial Deviance
##
##       Lambda  Measure        SE Nonzero
## min 0.004480 0.008922 0.0003782      20
## 1se 0.004693 0.009279 0.0004467      19
```

1(b)

The top 5 genes with the most influence are ENSG00000261012.2, ENSG00000121075.9, ENSG00000255399.3, ENSG00000259203.1 and ENSG00000225383.7

```
lasso.coef = coef(lasso, s = 'lambda.min')
lasso.coef@Dimnames[[1]][2:nrow(lasso.coef)] = gene_names

lasso.o = order(abs(lasso.coef), decreasing = TRUE)
lasso.coef[lasso.o,][1:6]
```

```
##       (Intercept) ENSG00000261012.2 ENSG00000121075.9 ENSG00000255399.3
##        0.64177683       -0.47928043        0.15412239        0.10894668
## ENSG00000259203.1 ENSG00000225383.7
##       -0.08616697        0.04188819
```

1(c)

For elastic net model, lambda.min has 315 nonzero coefficients. It means that 315 genes are likely useful for discriminating between tissues. The top 5 genes with the most influence are ENSG00000132670.20, ENSG00000259203.1, ENSG00000198382.8, ENSG00000267675.1 and ENSG00000255399.3.

```
# Elastic net
elnet <- cv.glmnet(x = expr,
                   y = tissue,
                   n_fold = 3,
                   family = "binomial",
                   alpha=0.5)

print(elnet)
```

```
##
## Call:  cv.glmnet(x = expr, y = tissue, n_fold = 3, family = "binomial",      alpha = 0.5)
##
## Measure: Binomial Deviance
##
##      Lambda  Measure        SE Nonzero
## min 0.00896 0.008897 0.0001978     315
## 1se 0.00896 0.008897 0.0001978     315
```

```
elnet.coef = coef(elnet, s = 'lambda.min')
elnet.coef@Dimnames[[1]][2:nrow(elnet.coef)] = gene_names

elnet.o = order(abs(elnet.coef), decreasing = TRUE)
elnet.coef[elnet.o,][1:6]
```

```
##        (Intercept) ENSG00000132670.20  ENSG00000259203.1  ENSG00000198382.8
##         0.81652318         0.02753132        -0.01548196         0.01286959
##  ENSG00000267675.1  ENSG00000255399.3
```

```
##        -0.01244745         0.01216709
```

Problem 2

The output of summary of p-values are given below. There are 29385 genes differentially expressed at the 0.01 level after Bonferroni correction and 34410 genes differentially expressed at the 0.01 level after FDR correction.

```
## test for differentially expressed genes
ps <- apply(expr[, 1:ncol(expr)], 2, function(y) {
    wilcox.test(y ~ tissue)$p.value
})
```

```
## adjusting p-values
summary(ps)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.0000  0.0000  0.2188  0.4660  1.0000
```

```
sum(p.adjust(ps, method = "bonferroni") <= 0.01) ## bonferroni
```

```
## [1] 29385
```

```
sum(p.adjust(ps, method = "fdr") <= 0.01) ## fdr
```

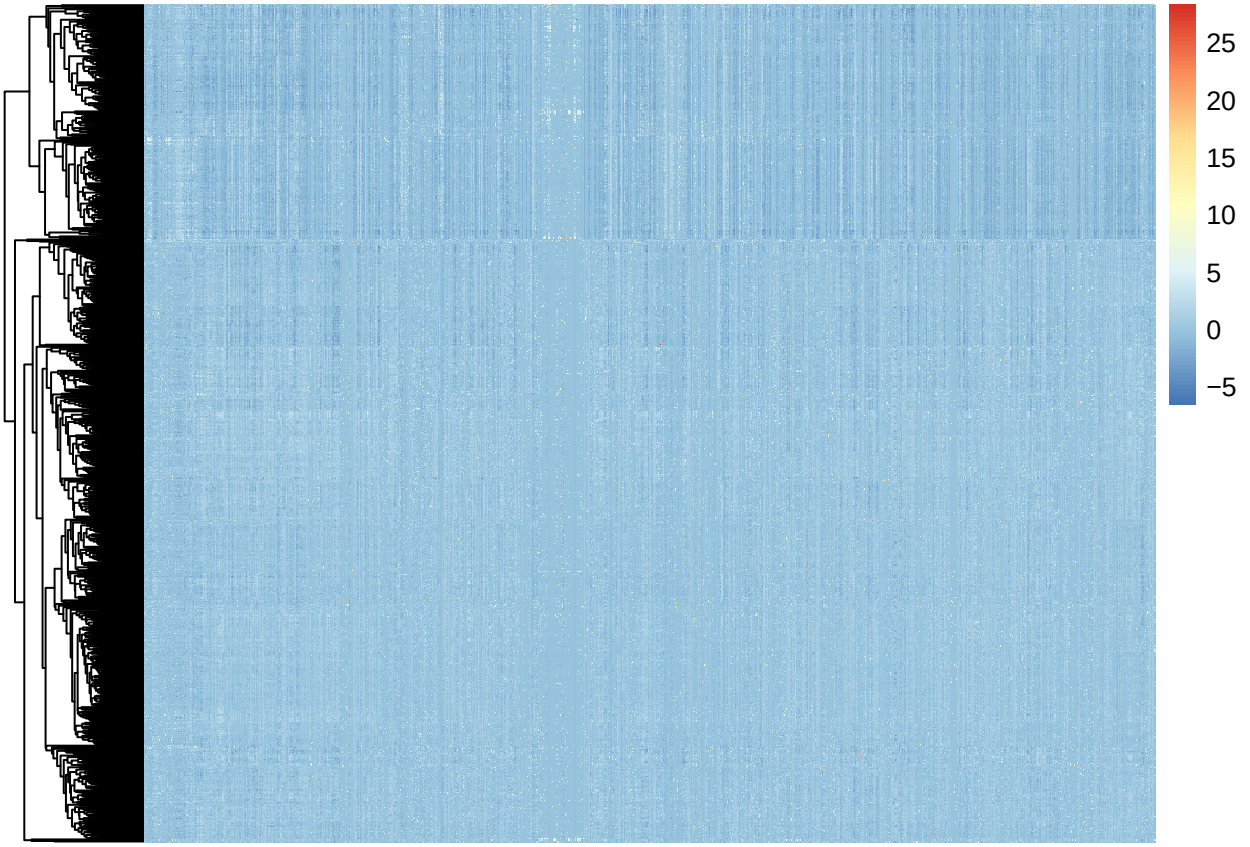
```
## [1] 34410
```

Problem 3

3(a)

From the hierarchical clustering, we can see that most entries have similar values but there are some entries with extreme values. Although it is not very clear, we can see that hierarchical clustering separates the samples into two clusters.

```
## hierarchical clustering
expr.scale.1000 = scale(expr[,1:1000])
res = pheatmap(expr.scale.1000,
               cluster_cols = FALSE,
               cluster_distance_rows = "correlation",
               show_rownames = FALSE,
               show_colnames = FALSE)
```



3(b)

From the contingency table, only one liver sample is misclassified as lung if we cut the tree at two clusters. Hence, the clustering result is nearly perfect.

```
## cut tree
clust <- cutree(res$tree_row, k = 2)

table(clust, tissue)
```

```
##      tissue
## clust Liver Lung
##     1     1  578
##     2   225    0
```

3(c)

If we choose k=128, there are three clusters and two of those should belong to the same cluster for actual data. The clustering result is worse than hierarchical clustering.

```
## snn clustering
expr.scale = scale(expr)
res <- sNNclust(expr.scale,
                k = 128,
                eps = 5,
                minPts = 5)
```

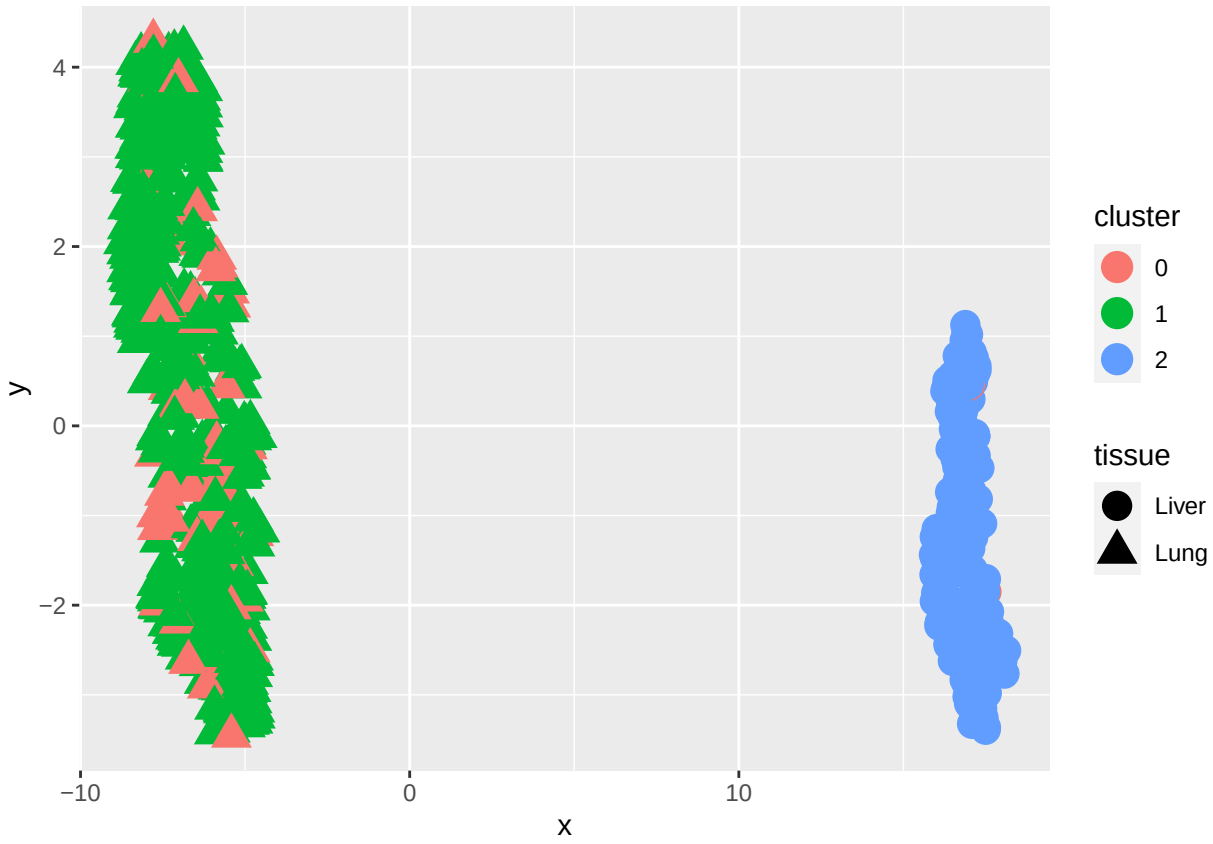
```r
clust <- res$cluster

table(clust, tissue)
```

```
##      tissue
## clust Liver Lung
##     0     5   71
##     1     0  507
##     2   221    0
```

3(d)

The umap result is shown below and we can see that two of the clusters should actually be merged into one.

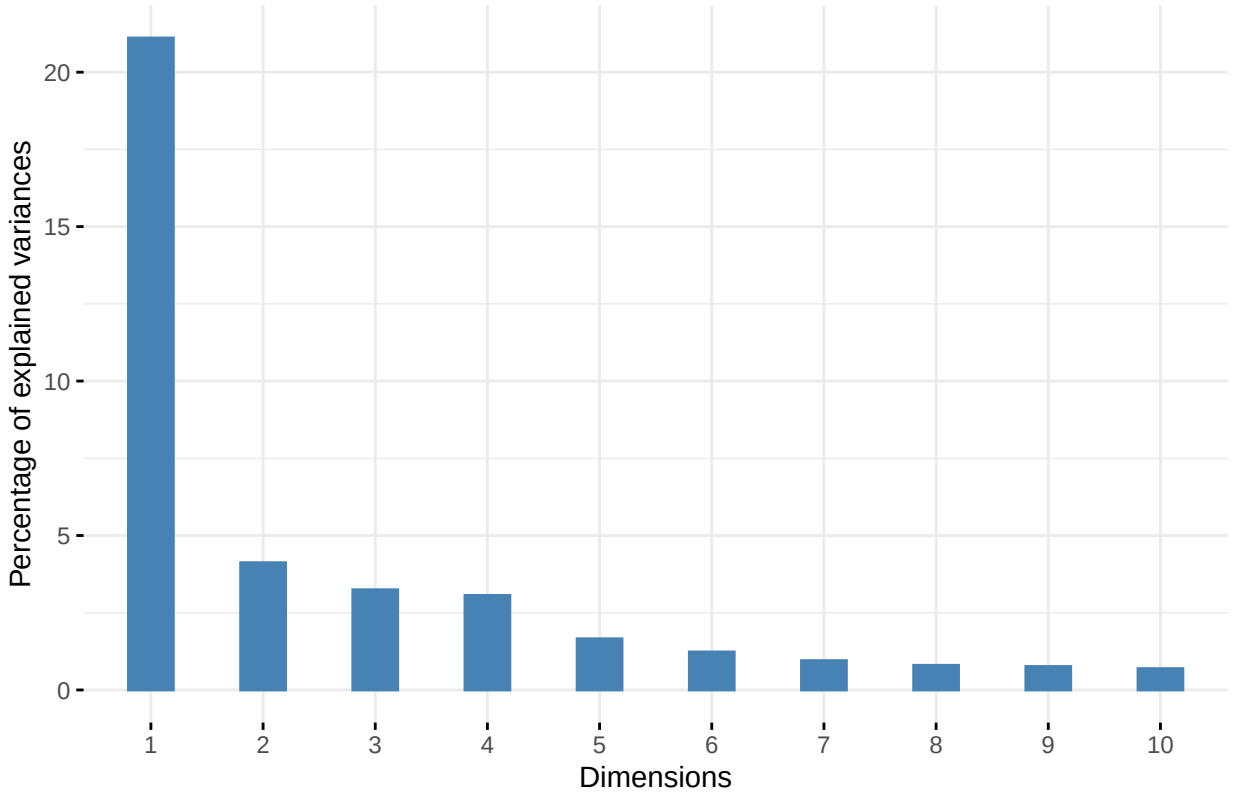
```r
## umap
um = umap(expr)
df = data.frame(x = um$layout[,1],
                y = um$layout[,2],
                tissue_type = tissue)

df$cluster = as.factor(clust)
ggplot(data = df,
       mapping = aes(x = x,
                     y = y,
                     shape = tissue,
                     color = cluster)) +
    geom_point(size = 5)
```



3(e)

From the scree plot, the first four principle components explain ~30% of the variation in data. Since all other principle components explain less than 2% of the variation. Four principle components might be enough to summarize the data.

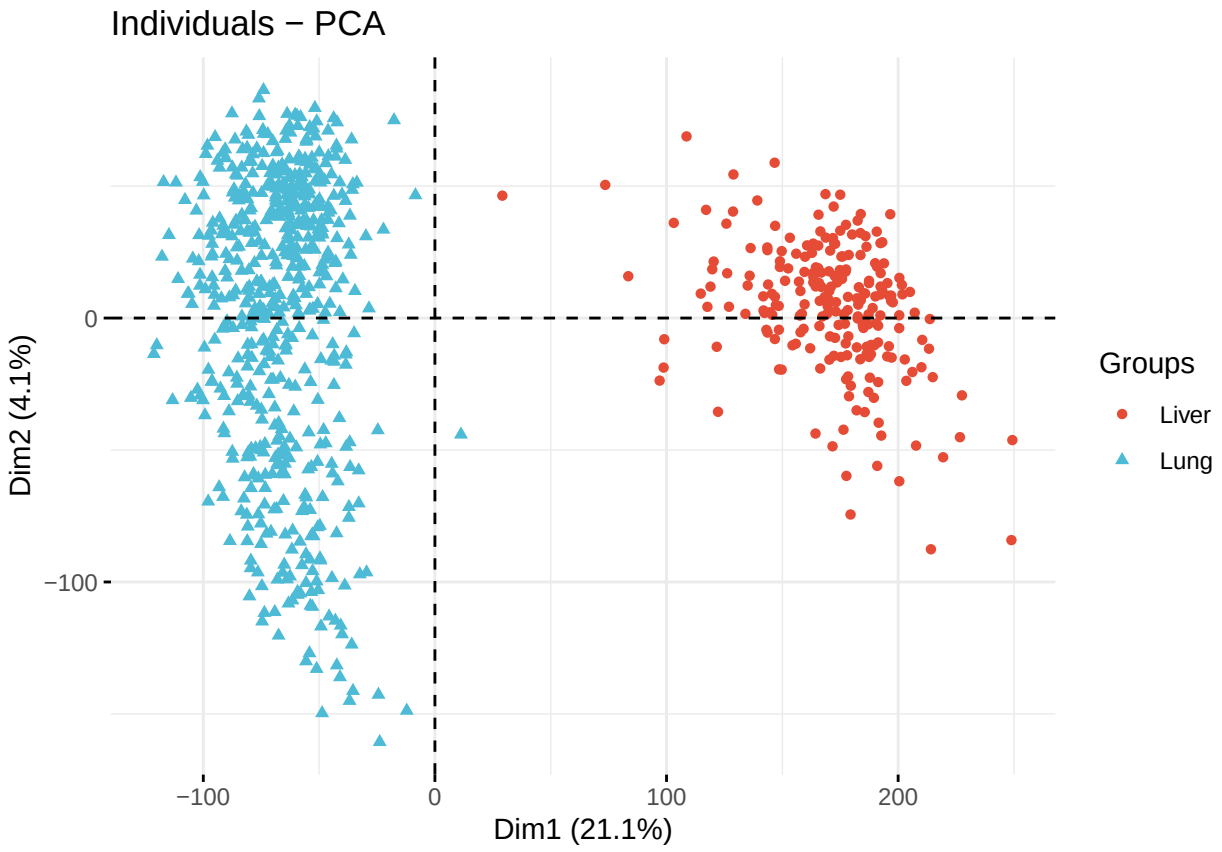
```
## pca
pca_out <- prcomp(expr, center = TRUE, scale = TRUE)
fviz_eig(pca_out, geom = "bar", bar_width = 0.4) + ggtitle("")
```



3(f)

From the plot, two clusters are separated nearly perfect.

```
## plot pca
fviz_pca_ind(pca_out,
             geom = "point",
             habillage = tissue,
             palette = "npg",
             mean.point = FALSE
)
```


Individuals − PCA
Dim2 (4.1%)
0
−100
−100
0
100
200
Dim1 (21.1%)
Groups
Liver
Lung