Final_Project

Tianqi Wu

5/11/2020

```
library(biomaRt)
library(tximport)
library(edgeR)
library(ggplot2)
library(amap)
library(dplyr)
library(gridExtra)
library(org.Mm.eg.db)
library(pheatmap)
library(statmod)
library(umap)
library(NMF)
library(factoextra)
```

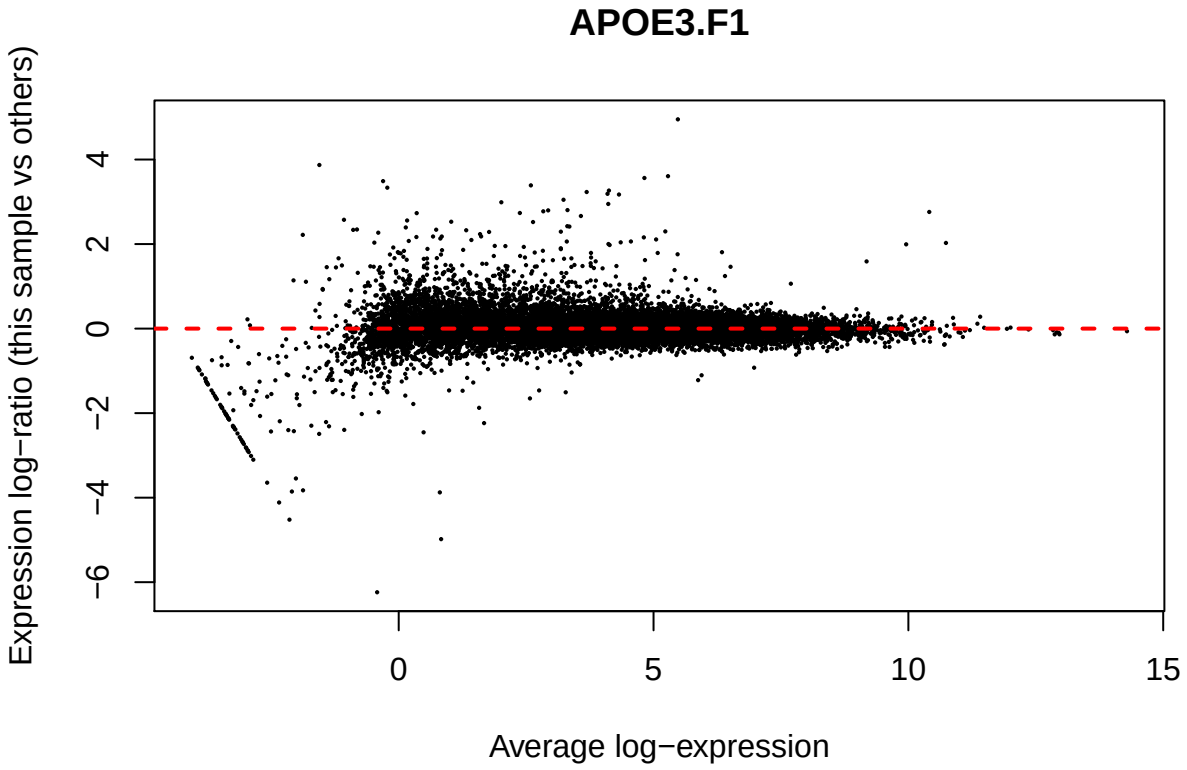
```
# Data Preparation
# ------------------------------------------------
## Import transcript-level abundance.tsv
samples <- read.table(file.path(getwd(), "samples.txt"), header=TRUE)
files <- file.path(getwd(),
                   "kallisto_out",
                   samples$run,
                   "abundance.tsv")

names(files) <- c(paste0("APOE3.F", 1:5), paste0("APOE3.M", 1:5),
                  paste0("APOE4.F", 1:5),paste0("APOE4.M", 1:5))

## Get mapping transcript ID to gene ID
mm = useMart("ensembl",
             dataset = "mmusculus_gene_ensembl")

tx2gene = getBM(attributes = c("ensembl_transcript_id_version",
                               "ensembl_gene_id"),
                mart = mm)

## Get counts
txi <- tximport(files,
                type = "kallisto",
                tx2gene = tx2gene,
                ignoreAfterBar = TRUE,
                countsFromAbundance = "lengthScaledTPM")
cts <- txi$counts
```

```r
## Check missing values
anyNA(cts)

## Save data object
# save(cts, samples, file = 'gene_data.rdata')
```

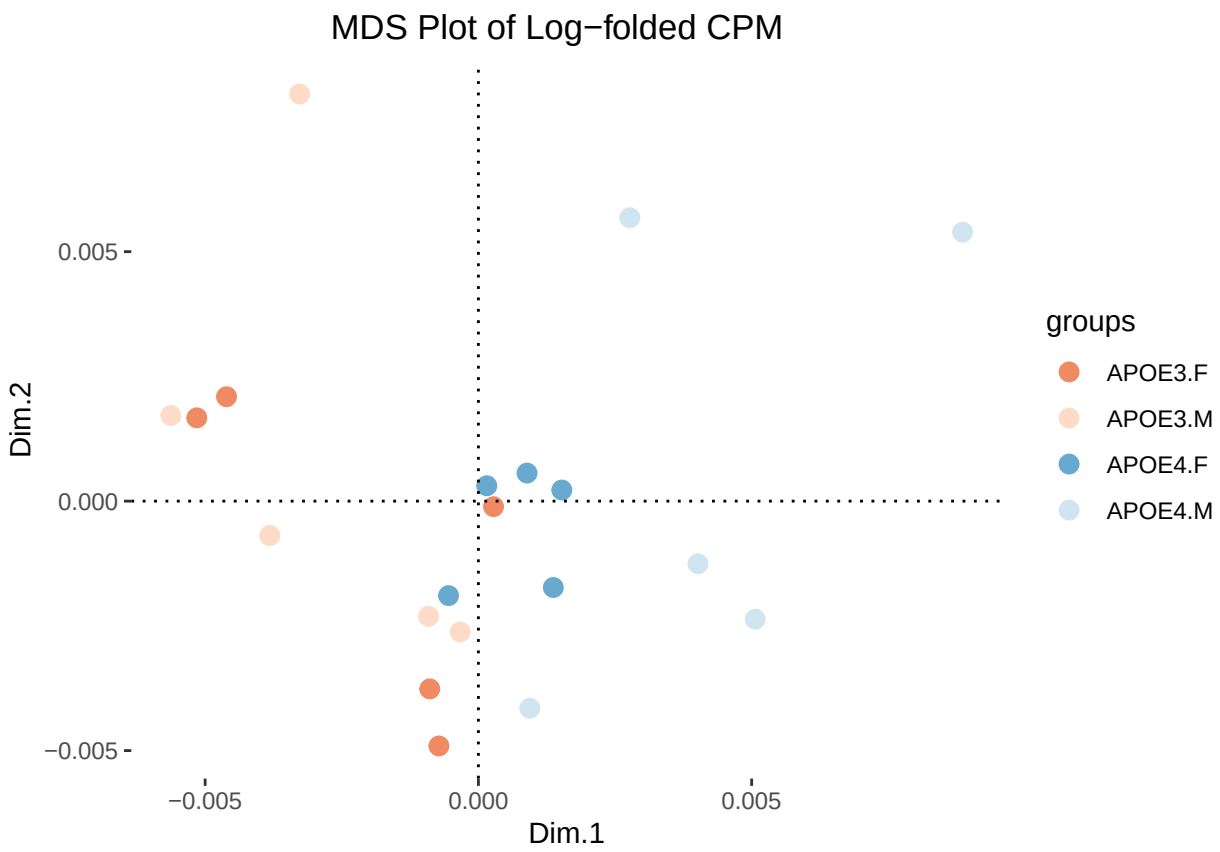
```r
## Load data from gene_data.rdata
load('gene_data.rdata')
genotype = samples[,2]
sex = samples[,3]
groups = samples[,4]
mm = useMart("ensembl",
             dataset = "mmusculus_gene_ensembl")
```

```r
## Create dgList object
## cts = read.table("GSE.txt", sep = ",", header = T, row.names = 1)
expr <- DGEList(counts = cts, group = colnames(cts))

## Keeps genes with minimum cpm of 1 in at least 2 samples in at least one group
countCheck <- (cpm(expr) >= 1)
countCheck = lapply(0:3, function(x)
                           rowSums(countCheck[, x*5 + c(1:5)]) >= 2)
countCheck = Reduce('+', countCheck)

expr <- expr[which(countCheck > 0), ] # 15983 genes left

## TMM normalization
expr <- calcNormFactors(expr, method="TMM")
```

```r
# Data Visualization after filtering
plotMD(cpm(expr, log=TRUE), column=1)
abline(h=0, col="red", lty=2, lwd=2)
```

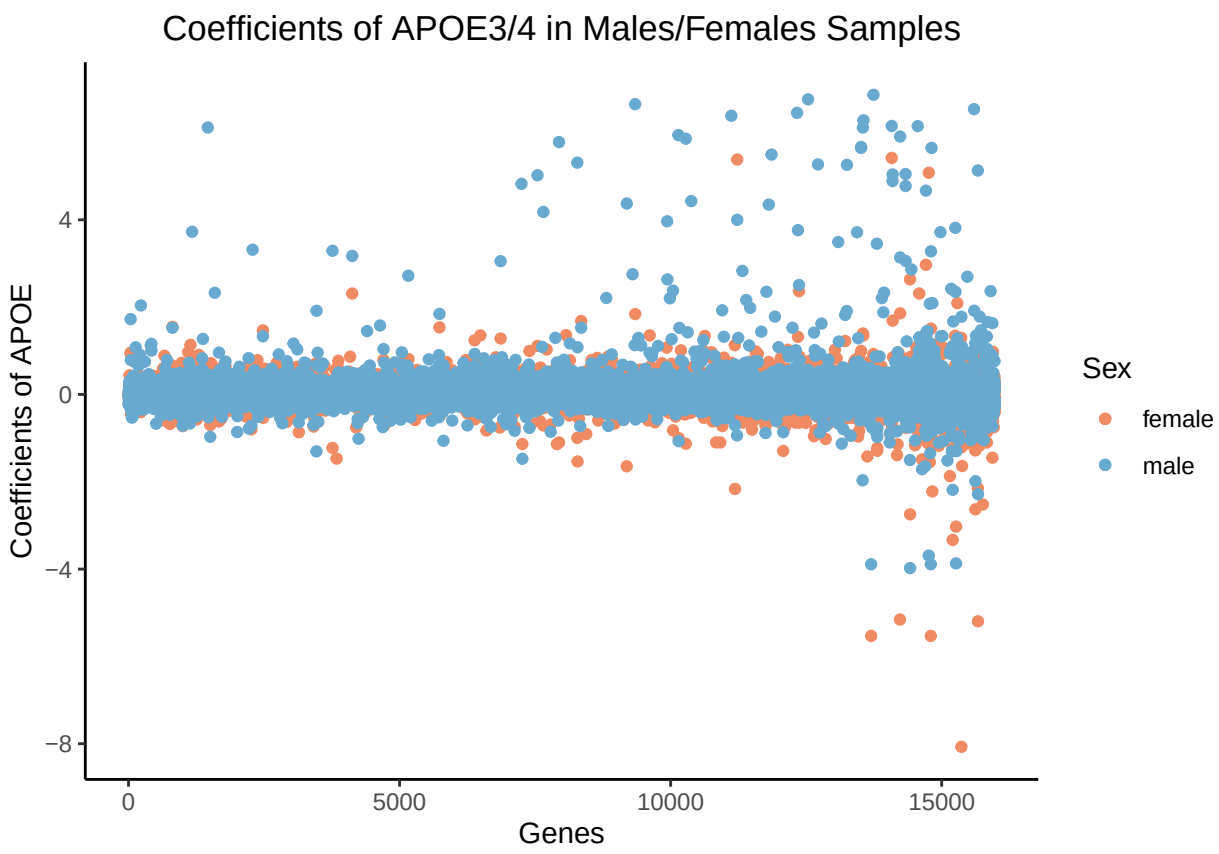



```
mycolor = c("#EF8A62", "#FDDBC7", "#67A9CF", "#D1E5F0")

# Data Exploration
# ---------------------------------------------
myMDS = function(expr, method, title=" ") {
  logexpr = cpm(expr, log=T)
  D = Dist(t(logexpr), method = method)

  # Convert distance to mds coordinates
  coords = data.frame(cmdscale(D), groups)
  colnames(coords) = c("Dim.1", "Dim.2", "groups")

  # Plot samples on principle coordinates (MDS)
  ggplot(coords) +
    geom_point(aes(Dim.1, Dim.2, group = groups, color = groups), size=3) +
    theme_classic() +
    geom_hline(yintercept = 0, linetype = 3) +
    geom_vline(xintercept = 0, linetype = 3) +
    labs(title = paste0('MDS Plot of Log-folded CPM ', title)) +
    theme(axis.line.y = element_blank(),
          axis.line.x = element_blank(),
          plot.title = element_text(hjust = 0.5),
          legend.position = "right") +
    scale_color_manual(values = mycolor)
}

myMDS(expr, "pearson")
```




```
# Detect interaction effect
# ------------------------------------------------
# Construct subgroups for sex = male and sex = female & model matrix
females = expr[, c(1:5, 11:15)]
males = expr[, c(6:10, 16:20)]
design = c(rep("APOE3", 5), rep("APOE4", 5))
design = model.matrix(~design)

# Estimate dispersion (within-group variation) seperately
females <- estimateDisp(females, design, robust=T)
males <- estimateDisp(males, design, robust=T)

# Run differential expression analysis respectively
femalesCoef <- glmQLFit(females, design, females$tagwise.dispersion)$coefficients
malesCoef <- glmQLFit(males, design, males$tagwise.dispersion)$coefficients

# Plot regression coefficients for males/females
ggplot() +
  geom_point(aes(1:nrow(femalesCoef),
                 femalesCoef[,2],
                 group = 'female',
                 color = "female")) +
  geom_point(aes(1:nrow(malesCoef),
                 malesCoef[,2],
                 group = 'male',
                 color = "male")) +
  theme_classic() +
```

```
  labs(x = "Genes",
       y = "Coefficients of APOE",
       title = "Coefficients of APOE3/4 in Males/Females Samples") +
  scale_color_manual(values = c('female' = mycolor[1],
                                'male' = mycolor[3]),
                     name = "Sex") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
# Differential Expression Analysis: Preparation
# ------------------------------------------------

# Function for regression and test
myTest = function(fit, contrast, p.value=0.05, n=2000) {

  # Run quasi-likelihood test
  lrt <- glmQLFTest(fit, contrast = contrast)
  #print(summary(decideTests(lrt, p.value=p.value)))

  # Pick genes with fdr p-value greatee than our thresold
  tops = topTags(lrt, n=n, p.value=p.value)$table

  # Find names of top_genes
  top_gene_name = getBM(filters= "ensembl_gene_id",
                        attributes = c('ensembl_gene_id', 'external_gene_name'),
                        values = rownames(tops),
                        mart = mm)
  colnames(top_gene_name) = c("gene_id", "gene_names")
```

```
  # Return names of differentially expressed genes and there fdr adjusted p-values
  result <- data.frame(gene = as.character(rownames(tops)),
                       p.value = tops$PValue,
                       logFC = tops$logFC)

  result = left_join(result, top_gene_name, by = c('gene' = 'gene_id'))
  return (result)
}

myBarplot = function(df, sex, gene) {

  p = ggplot(df) +
      geom_bar(aes_string(1:10, gene, group = 'group', fill = 'group'),
               stat = 'identity') +
      theme_classic() +
      theme(axis.line.y = element_blank(),
            axis.ticks.y = element_blank(),
            axis.text.y = element_blank(),
            legend.position = "right") +
      labs(x = paste0(sex, " Samples"),
           y = paste0("Expression level of ", gene)) +
      scale_fill_manual(values = c('APOE3' = mycolor[2],
                                   'APOE4' = mycolor[4]),
                    name = "") +
      coord_flip()

  return(p)
}

myLineplot = function(df, gene) {

p = ggplot(mapping = aes_string('APOE', gene, group = 'sex', color = 'sex'),
           mean_inter_expr) +
    geom_point() +
    geom_line(aes(linetype = sex), size = 1) +
    theme_classic() +
    theme(axis.line.x = element_blank(),
          axis.ticks.x = element_blank()) +
    scale_x_discrete(labels=c("APOE3","APOE4"),
                     expand=c(0.1, 0.1)) +
    labs(x = "", y = "Expression", title = gene)

  return(p)
}

# Model matrix
design = model.matrix(~0+groups)
colnames(design) = levels(groups)

# Estimate dispersion
expr <- estimateDisp(expr, design, robust=T)

# Contrast for tests
```

```
my.contrast = makeContrasts(APOE3vs4.F = APOE4.F-APOE3.F,
                            APOE3vs4.M = APOE4.M-APOE3.M,
                            APOE3vs4.FvsM = (APOE4.F-APOE3.F) - (APOE4.M-APOE3.M),
                            levels = design)
```

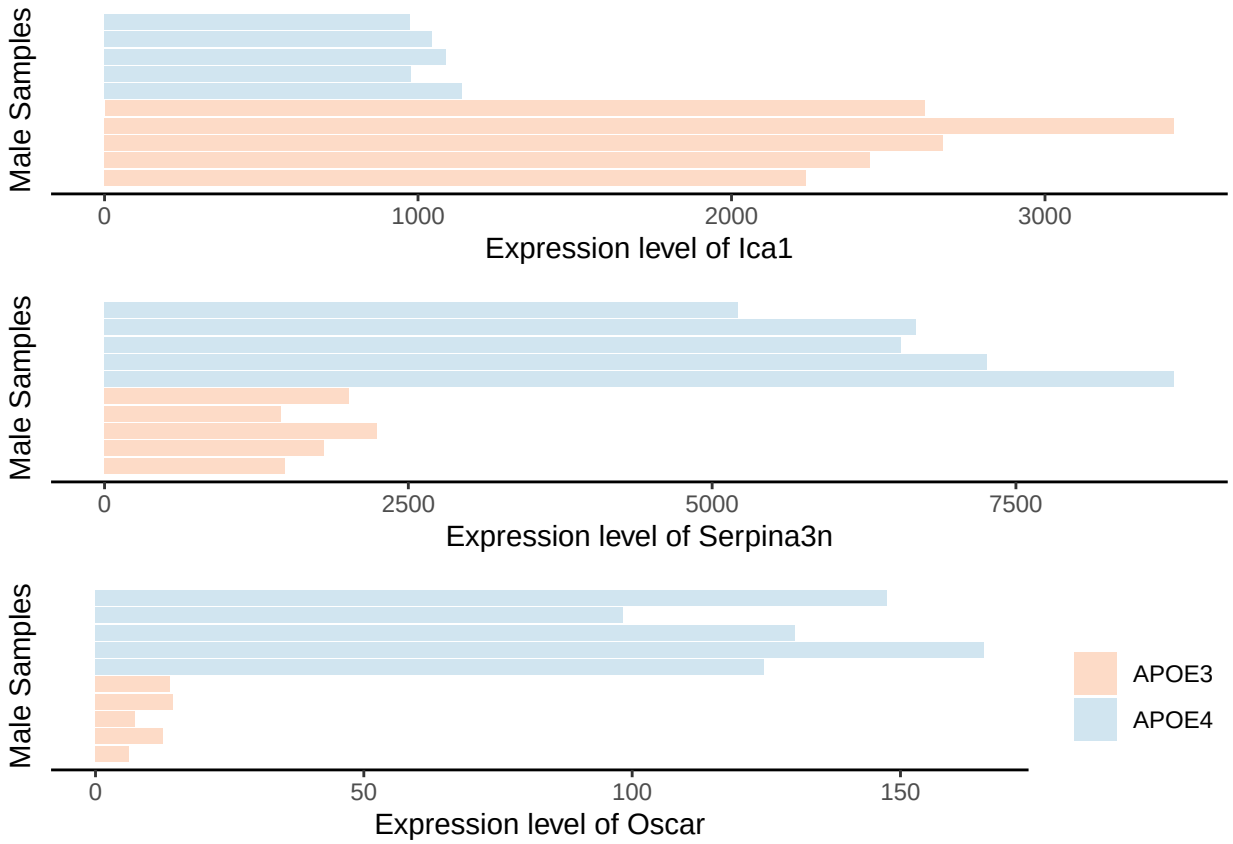
hAPOE4 causes differential expression of Ica1, Serpina3n and Oscar in both males and females

After determining the needs of interaction term, we start to build the design matrix. The GLM in the following analysis is ~0+groups, where groups are specified above (APOE3.F, APOE3.M, APOE4.F, APOE4.M) and 0 is included to drop the intercept column. We further define contrasts to specify the tests interested. There are three tests in total: F.APOE3vs4 (The effect of APOE for female group), M.APOE3vs4 (The effect the APOE for male group), FvsM.APOE3vs4 (The difference of effect of APOE for female vs. male groups). Note that the last term can be seen as interaction term.

```
# Differential Expression Analysis: APOE in males
# -----------------------------------------------
# Fit glm for quasi-likelihood
fit <- glmQLFit(expr, design, expr$tagwise.dispersion)

# Run tests for hAPOE4 in males
DEGinMales = myTest(fit, my.contrast[,'APOE3vs4.M'])

# Show results
head(DEGinMales)
```

```
##                  gene      p.value     logFC gene_names
## 1 ENSMUSG00000046952 1.137843e-39  5.761883     Gm5815
## 2 ENSMUSG00000062995 1.609619e-38 -1.367038       Ica1
## 3 ENSMUSG00000021091 1.140679e-37  1.921515  Serpina3n
## 4 ENSMUSG00000054594 2.045639e-37  3.614955      Oscar
## 5 ENSMUSG00000096768 1.777273e-36 -5.582101    Gm47283
## 6 ENSMUSG00000038608 1.091809e-18  3.971723     Dock10
```

```
# cat(paste(DEGinMales$gene, collapse = '\n'))

# Find their corresponding expression levels of Ica1, Serpina3n and Oscar
idx_males = sapply(DEGinMales$gene[c(2:4, 6)], function(x) which(rownames(males$counts) == x))
top_male_expr = as.data.frame(t(males$counts[idx_males, ]))
colnames(top_male_expr) = DEGinMales$gene_names[c(2:4, 6)]
top_male_expr$group = c(rep('APOE3', 5), rep('APOE4', 5))
# top_male_expr

# Plot the expression of these three genes
ica1_male = myBarplot(top_male_expr, 'Male', 'Ica1')
serpina3n_male = myBarplot(top_male_expr, 'Male', 'Serpina3n')
oscar_male = myBarplot(top_male_expr, 'Male', 'Oscar')
dock10_male = myBarplot(top_male_expr, 'Male', 'Dock10')

grid.arrange(ica1_male + theme(legend.position="none"),
             serpina3n_male + theme(legend.position="none"),
             oscar_male,
             nrow = 3)
```




```
# Differential Expression Analysis: hAPOE in females
# ------------------------------------------------
DEGinFemales = myTest(fit, my.contrast[,'APOE3vs4.F'])
```

```
## Cache found
```

```
## Warning: Column `gene`/`gene_id` joining factor and character vector, coercing
## into character vector
```

```
# Show results
head(DEGinFemales, 10)
```

```
##                  gene      p.value      logFC gene_names
## 1  ENSMUSG00000046952 2.591195e-47  7.7680787     Gm5815
## 2  ENSMUSG00000021091 8.500178e-45  2.1170827  Serpina3n
## 3  ENSMUSG00000062995 6.834049e-38 -1.3544659       Ica1
## 4  ENSMUSG00000054594 1.356927e-34  3.4115980      Oscar
## 5  ENSMUSG00000096768 1.303109e-25 -4.3660795    Gm47283
## 6  ENSMUSG00000110275 1.962790e-20 -3.7975726     Gm5905
## 7  ENSMUSG00000111619 1.464131e-16 -1.6919008    Gm48348
## 8  ENSMUSG00000073643 9.482035e-16 -0.6685258      Wdfy1
## 9  ENSMUSG00000006154 1.477418e-15  2.2331768     Eps8l1
## 10 ENSMUSG00000113337 1.587518e-13 -3.6378287    Gm19220
```

```
head(DEGinMales, 10)
```

```
##                  gene      p.value      logFC gene_names
## 1  ENSMUSG00000046952 1.137843e-39  5.7618828     Gm5815
## 2  ENSMUSG00000062995 1.609619e-38 -1.3670385       Ica1
```

```
## 3  ENSMUSG00000021091 1.140679e-37  1.9215153  Serpina3n
## 4  ENSMUSG00000054594 2.045639e-37  3.6149549      Oscar
## 5  ENSMUSG00000096768 1.777273e-36 -5.5821013    Gm47283
## 6  ENSMUSG00000038608 1.091809e-18  3.9717229     Dock10
## 7  ENSMUSG00000073643 1.208775e-16 -0.6908893      Wdfy1
## 8  ENSMUSG00000075014 3.314765e-16  7.2797392    Gm10800
## 9  ENSMUSG00000073294 3.875043e-16  7.0520898   AU022751
## 10 ENSMUSG00000006154 3.371668e-15  2.2038524     Eps8l1
```

```
# Find their corresponding expression levels of Ica1, Serpina3n and Oscar
idx_females = sapply(DEGinFemales$gene[c(2:4)],
                     function(x) which(rownames(females$counts) == x))
top_female_expr = as.data.frame(t(females$counts[idx_females, ]))
colnames(top_female_expr) = DEGinFemales$gene_names[c(2:4)]
top_female_expr$group = c(rep('APOE3', 5), rep('APOE4', 5))
top_female_expr
```

```
##          Serpina3n      Ica1      Oscar group
## APOE3.F1  1532.731 2254.2686   6.405579 APOE3
## APOE3.F2  1725.996 2609.8303  13.847520 APOE3
## APOE3.F3  1302.973 2002.1684  13.673584 APOE3
## APOE3.F4  1780.632 2736.9906  14.217104 APOE3
## APOE3.F5  2416.142 3049.9454  12.471361 APOE3
## APOE4.F1  6894.533 1000.6258 130.522712 APOE4
## APOE4.F2  9607.032 1168.2745 145.124247 APOE4
## APOE4.F3  9619.218 1256.2147 148.220892 APOE4
## APOE4.F4  7396.161  942.8958 107.181529 APOE4
## APOE4.F5  7988.916 1074.4484 188.545475 APOE4
```

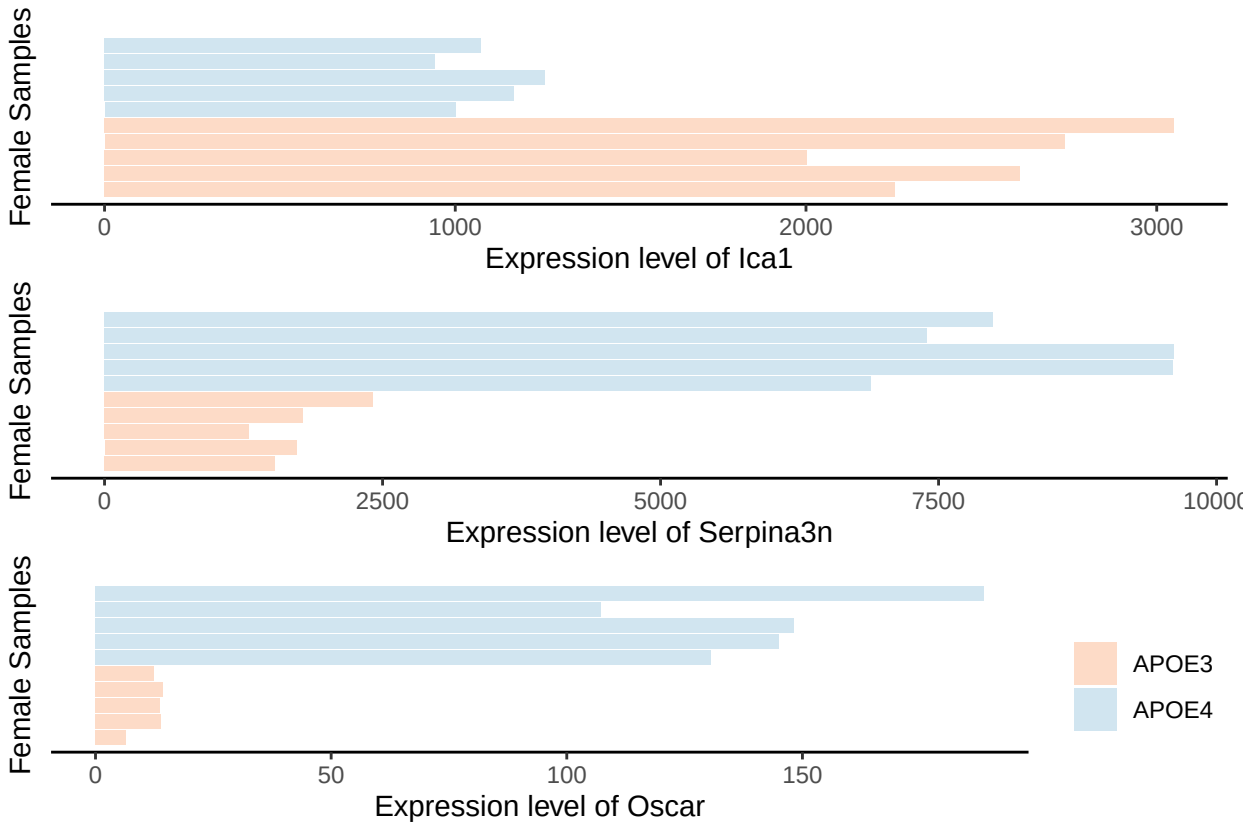
```
top_female_expr
```

```
##          Serpina3n      Ica1      Oscar group
## APOE3.F1  1532.731 2254.2686   6.405579 APOE3
## APOE3.F2  1725.996 2609.8303  13.847520 APOE3
## APOE3.F3  1302.973 2002.1684  13.673584 APOE3
## APOE3.F4  1780.632 2736.9906  14.217104 APOE3
## APOE3.F5  2416.142 3049.9454  12.471361 APOE3
## APOE4.F1  6894.533 1000.6258 130.522712 APOE4
## APOE4.F2  9607.032 1168.2745 145.124247 APOE4
## APOE4.F3  9619.218 1256.2147 148.220892 APOE4
## APOE4.F4  7396.161  942.8958 107.181529 APOE4
## APOE4.F5  7988.916 1074.4484 188.545475 APOE4
```

```
top_male_expr
```

```
##               Ica1 Serpina3n      Oscar    Dock10 group
## APOE3.M1 2238.5481  1482.481   6.316991  4229.932 APOE3
## APOE3.M2 2442.4542  1806.062  12.526648  3365.626 APOE3
## APOE3.M3 2675.5902  2240.194   7.371719  3354.618 APOE3
## APOE3.M4 3412.4018  1454.861  14.438374  3297.259 APOE3
## APOE3.M5 2615.8026  2015.034  13.815773  4602.736 APOE3
## APOE4.M1 1139.2324  8802.915 124.655683  5153.461 APOE4
## APOE4.M2  977.4824  7260.984 165.588774 99688.816 APOE4
## APOE4.M3 1089.8211  6551.872 130.260579 67409.642 APOE4
## APOE4.M4 1045.8830  6675.338  98.234979 73740.093 APOE4
## APOE4.M5  973.3483  5214.073 147.454795 45505.697 APOE4
```

```
# Plot the expression of these three genes
ica1_female = myBarplot(top_female_expr, 'Female', 'Ica1')
serpina3n_female = myBarplot(top_female_expr, 'Female', 'Serpina3n')
oscar_female = myBarplot(top_female_expr, 'Female', 'Oscar')

grid.arrange(ica1_female + theme(legend.position="none"),
             serpina3n_female + theme(legend.position="none"),
             oscar_female,
             nrow = 3,
             top = "")
```




```
# How APOE4 differentially influences males and females?
# ------------------------------------------------
# Run myTest
DEGinteraction = myTest(fit, my.contrast[,'APOE3vs4.FvsM'])

# Show results
DEGinteraction[1:10,]
```

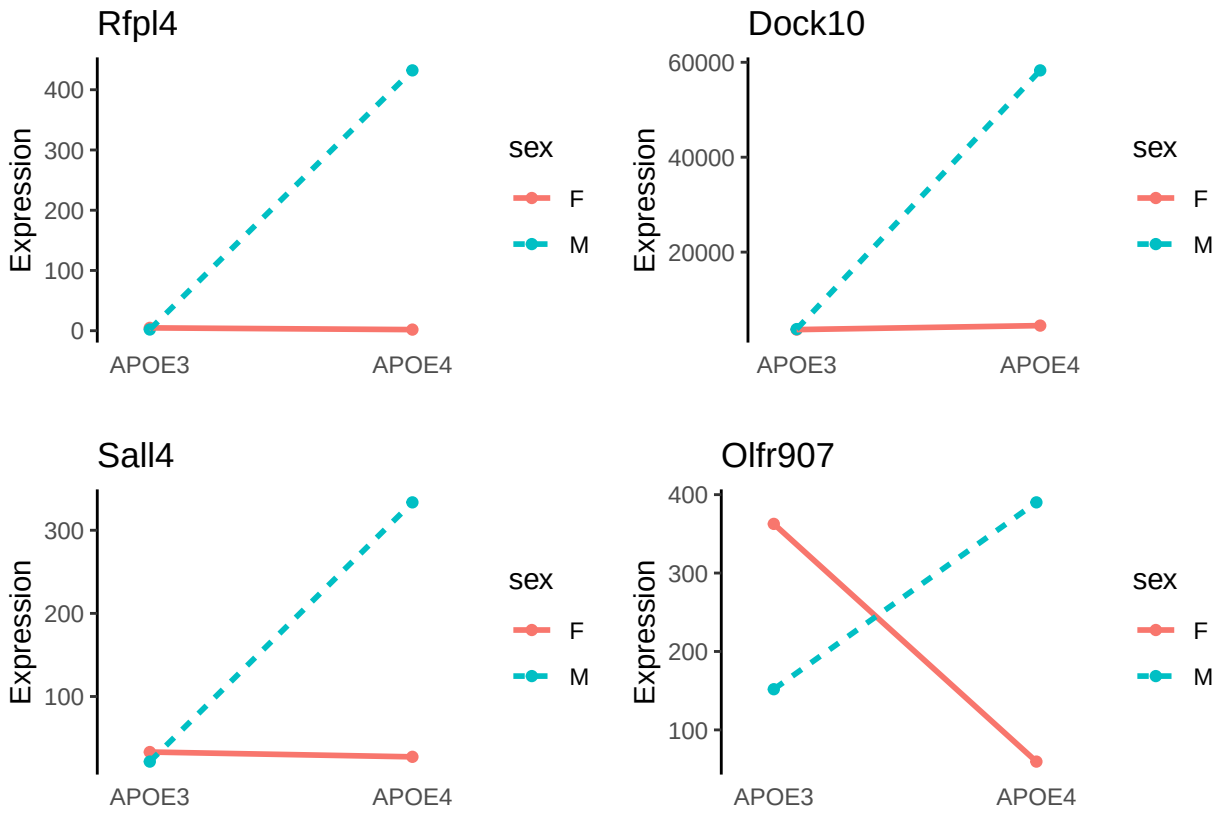
```
##                  gene      p.value     logFC gene_names
## 1  ENSMUSG00000083933 3.066949e-14 -5.227185    Gm15515
## 2  ENSMUSG00000085998 3.179439e-14 -6.223944   AW822252
## 3  ENSMUSG00000082368 9.645073e-13 -3.163522    Gm11225
## 4  ENSMUSG00000075014 1.536303e-12 -7.796276    Gm10800
## 5  ENSMUSG00000073294 3.527270e-12 -7.696191   AU022751
## 6  ENSMUSG00000035191 3.991041e-11 -9.092027      Rfpl4
## 7  ENSMUSG00000038608 6.319109e-11 -3.816766     Dock10
## 8  ENSMUSG00000027547 2.709274e-10 -4.333558      Sall4
```

```
## 9  ENSMUSG00000094810 8.027287e-10 -4.042918    Olfr907
## 10 ENSMUSG00000062248 2.522253e-09 -3.541679       Cks2
```

```
# Find their corresponding expression levels of Ica1, Serpina3n and Oscar
idx_inter = sapply(DEGinteraction$gene[c(6:10, 12)],
                   function(x) which(rownames(expr$counts) == x))
top_inter_expr = as.data.frame(t(expr$counts[idx_inter, ]))

colnames(top_inter_expr) = DEGinteraction$gene_names[c(6:10, 12)]
top_inter_expr$group = sapply(rownames(top_inter_expr),
                              function(x) substr(x, 1, 7))

top_inter_expr[1:10,]
```

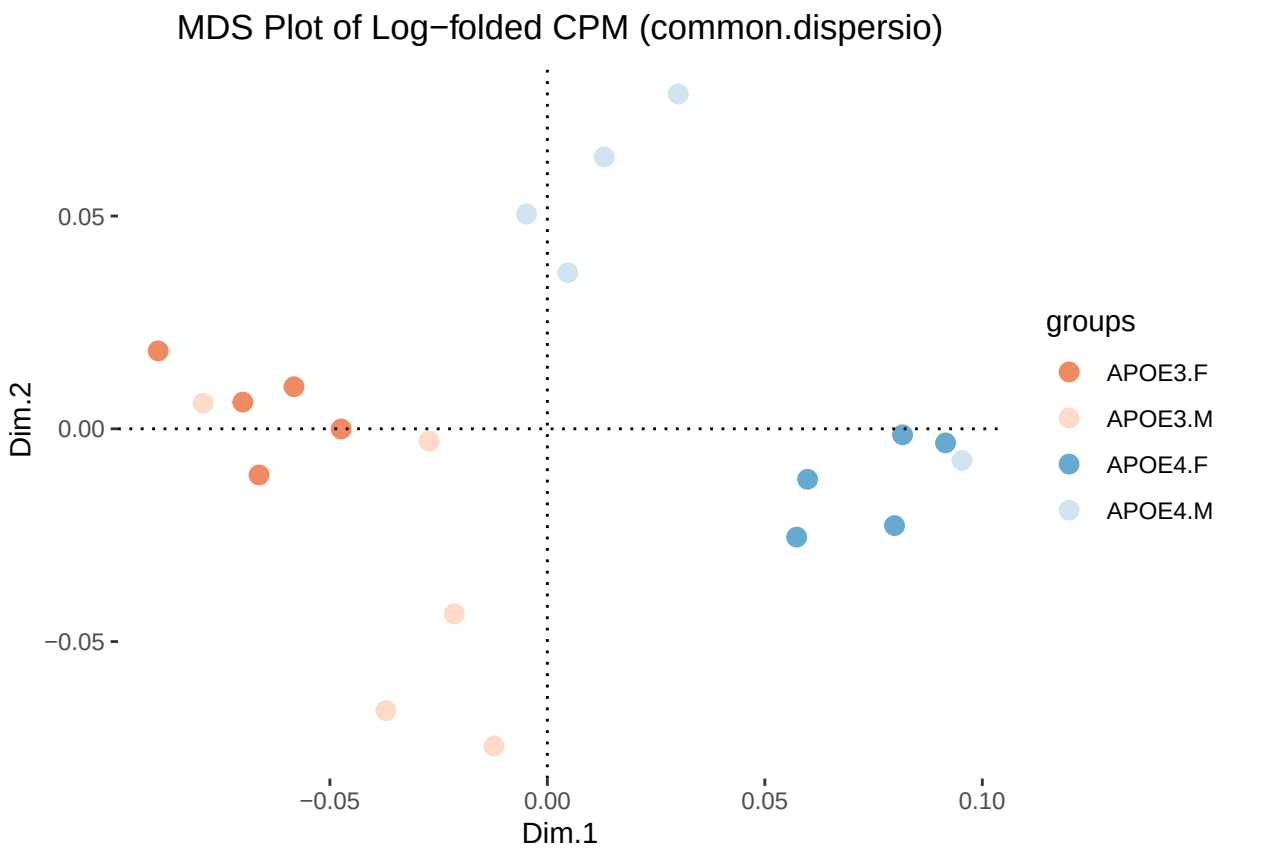
```
##              Rfpl4   Dock10    Sall4   Olfr907     Cks2     Nlrp4f   group
## APOE3.F1 4.9754564 3638.431 28.82336 274.26097 28.77256  8.1599343 APOE3.F
## APOE3.F2 3.9675169 4479.458 58.44345 297.65865 23.66138  3.0844879 APOE3.F
## APOE3.F3 4.7096818 3150.746 13.38178 273.62399 34.25266  7.8740172 APOE3.F
## APOE3.F4 2.9513545 3075.582 38.77057 235.87728 46.50327  9.3845805 APOE3.F
## APOE3.F5 6.9029691 3994.370 26.48854 731.98870 60.46108 10.4088158 APOE3.F
## APOE3.M1 6.0055046 4229.932 20.05023  81.51938 34.62351  3.9822788 APOE3.M
## APOE3.M2 0.9948535 3365.626 32.30169 140.77849 29.12178  0.9139597 APOE3.M
## APOE3.M3 0.0000000 3354.618 19.56502 329.32021 42.29364 10.1267694 APOE3.M
## APOE3.M4 0.9829071 3297.259 20.09666 124.10474 27.74768  3.8626017 APOE3.M
## APOE3.M5 1.9631902 4602.736 16.33530  83.58288 90.51879  3.0797942 APOE3.M
```

```
# Mean expression grouped by group
mean_inter_expr = top_inter_expr %>%
                  group_by(group) %>%
                  summarise(Rfpl4 = mean(Rfpl4),
                            Dock10 = mean(Dock10),
                            Sall4 = mean(Sall4),
                            Olfr907 = mean(Olfr907),
                            Cks2 = mean(Cks2),
                            Nlrp4f = mean(Nlrp4f)) %>%
                  mutate(sex = c('F', 'M', 'F', 'M'),
                         APOE = c('APOE3', 'APOE3', 'APOE4', 'APOE4'))



# Visualization of Rfpl4, Dock10, Sall4
rfpl4 = myLineplot(mean_inter_expr, 'Rfpl4')
dock10 = myLineplot(mean_inter_expr, 'Dock10')
sall4 = myLineplot(mean_inter_expr, 'Sall4')
olfr907 = myLineplot(mean_inter_expr, 'Olfr907')
cks2 = myLineplot(mean_inter_expr, 'Cks2')
nlrp4f = myLineplot(mean_inter_expr, 'Nlrp4f')

grid.arrange(rfpl4, dock10, sall4, olfr907, nrow = 2)
```

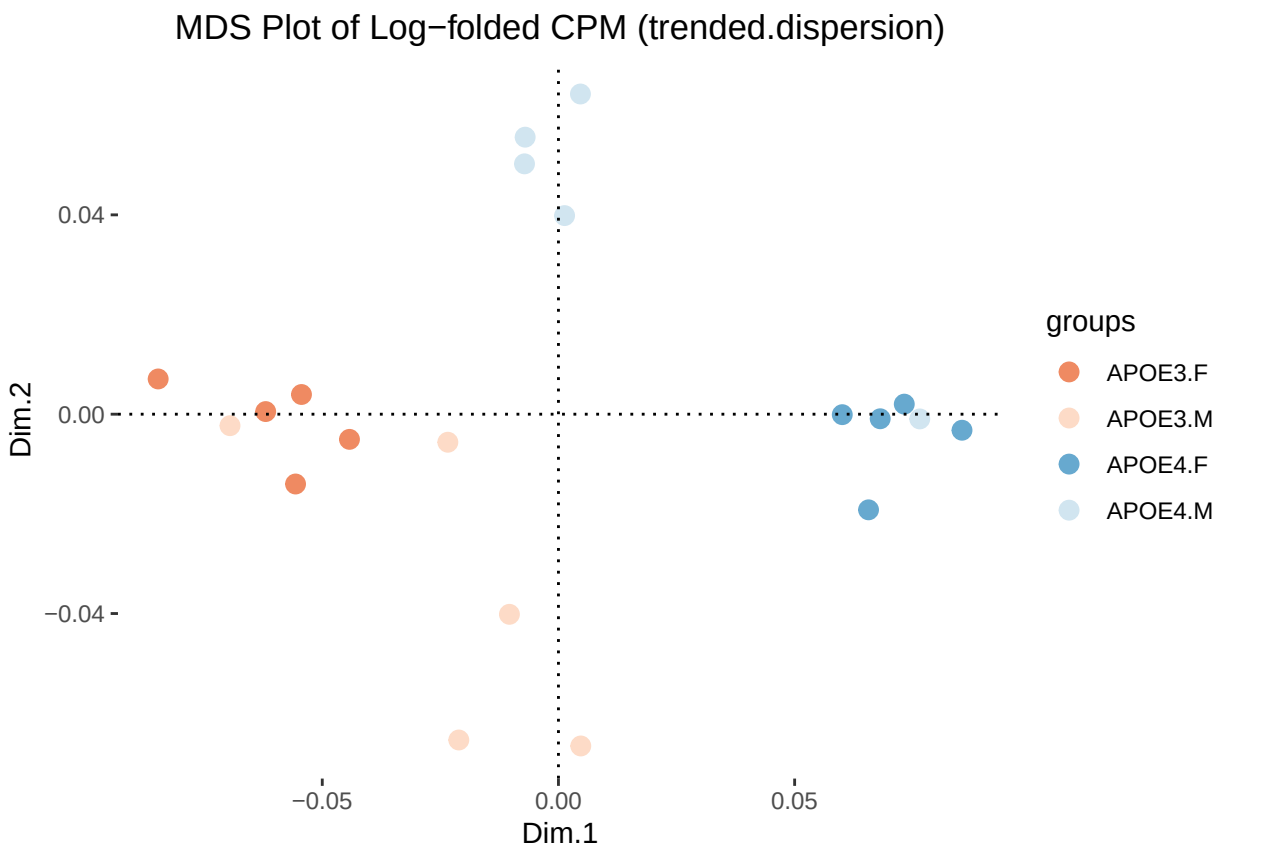



```
# Differential Expression of APOE4 over APOE3 in Males/Females"
```

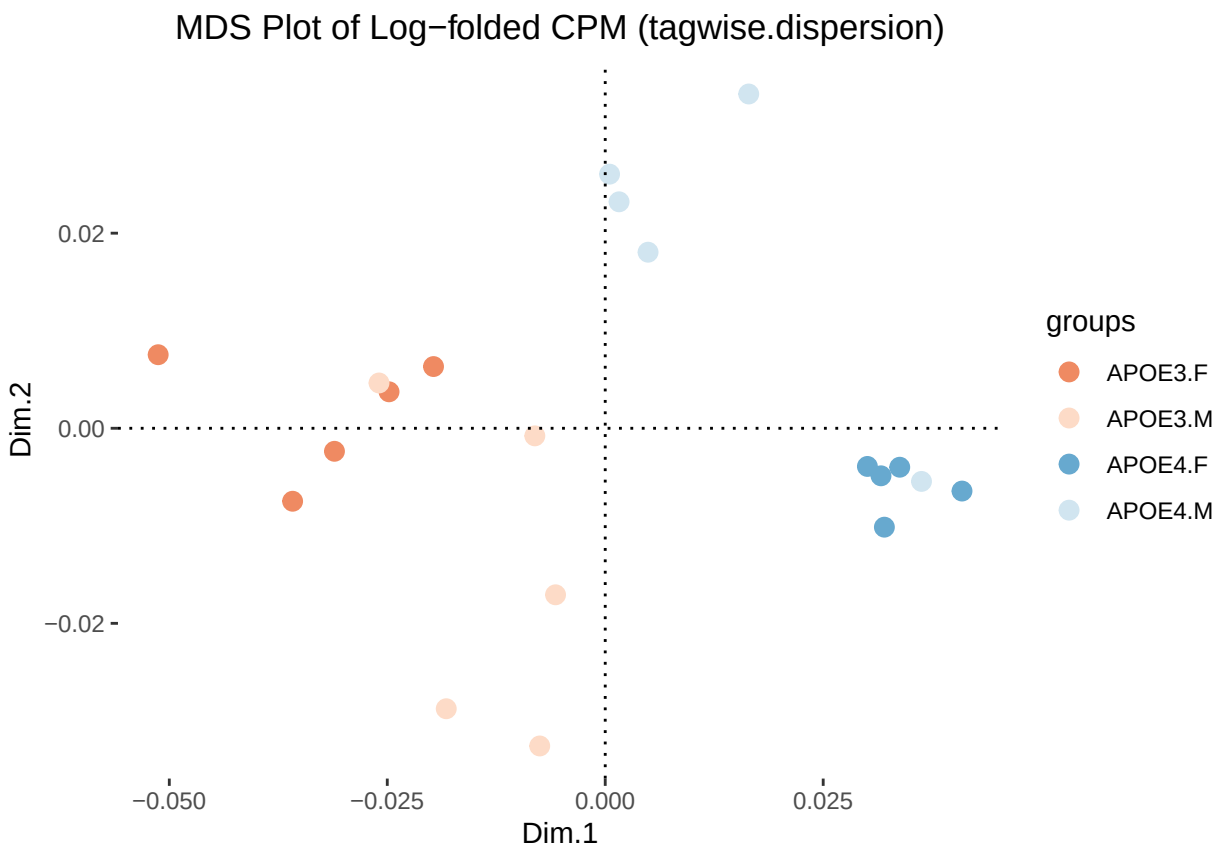
```
my.glm = function(expr, my.contrast, dispersion, p.value=0.05, n=2000) {

  fit <- glmQLFit(expr, design, dispersion)
  lrt <- glmQLFTest(fit, contrast=my.contrast)
  #print(summary(decideTests(lrt, p.value=p.value)))
  degs <- rownames(topTags(lrt, n=n, p.value=p.value)$table)
  return (degs)
}

# Fit glm model to see the effect the different dispersion
# degs.commom = myTest(expr, my.contrast[,'APOE3vs4.F'], common.dispersion)
# degs.trended = myTest(expr, my.contrast[,'APOE3vs4.F'], trended.dispersion)
degs = my.glm(expr, my.contrast[,'APOE3vs4.F'], expr$tagwise.dispersion)

# Estimate dispersion (within-group variation)
expr <- estimateDisp(expr, design, robust=T)

# Fit glm model to see the effect the different dispersion
degs.commom = my.glm(expr, my.contrast[,'APOE3vs4.F'], expr$common.dispersion)
degs.trended = my.glm(expr, my.contrast[,'APOE3vs4.F'], expr$trended.dispersion)
degs.tagwise = my.glm(expr, my.contrast[,'APOE3vs4.F'], expr$tagwise.dispersion)
myMDS(expr[degs.commom,], "pearson", title='(common.dispersio)')
```




```
myMDS(expr[degs.trended,], "pearson", title='(trended.dispersion)')
```



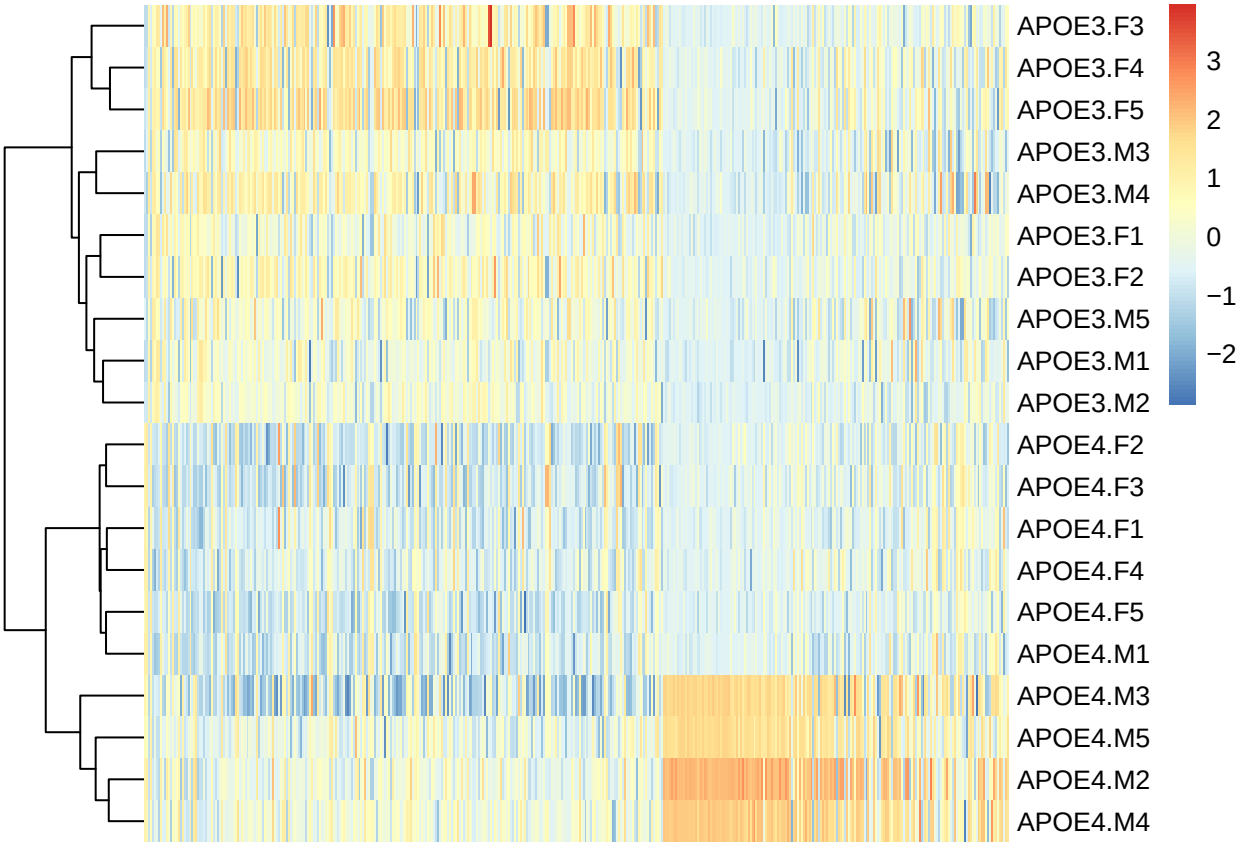

```
myMDS(expr[degs.tagwise,], "pearson", title='(tagwise.dispersion)')
```




```
# show top 5 genes with counts and external_gene_name
show_top_gene = function(degs) {
  ## Get mapping ensembl_gene_id to external_gene_name
  mm = useMart("ensembl", dataset = "mmusculus_gene_ensembl")

  top_gene = degs[1:5]
  top_gene_name = getBM(filters= "ensembl_gene_id",
                attributes=c('ensembl_gene_id', 'external_gene_name'),
                values=top_gene,mart=mm)

  top_expr = expr$counts[which(rownames(expr) %in% top_gene), ]
  tbl = merge(top_gene_name,top_expr,by.x='ensembl_gene_id',by.y='row.names')
  rownames(tbl) = tbl$external_gene_name
  tbl = subset(tbl,select=-c(ensembl_gene_id,external_gene_name))
  kable(t(tbl))

}

# union all the DE genes
degs_union = Reduce(union, list(DEGinFemales$gene, DEGinMales$gene, DEGinteraction$gene))
```

```
# hierarchical clustering
# ------------------------------------------------
set.seed(0)
expr.log = cpm(expr, log = TRUE)

## select genes at 0.1 significance level
```

```
expr.top = expr.log[degs_union, ]
expr.top.scale = scale(t(expr.top))

res = pheatmap(expr.top.scale,
               cluster_rows = T,
               cluster_cols = F,
               cluster_distance_rows = "correlation",
               show_rownames = T,
               show_colnames = F)
```

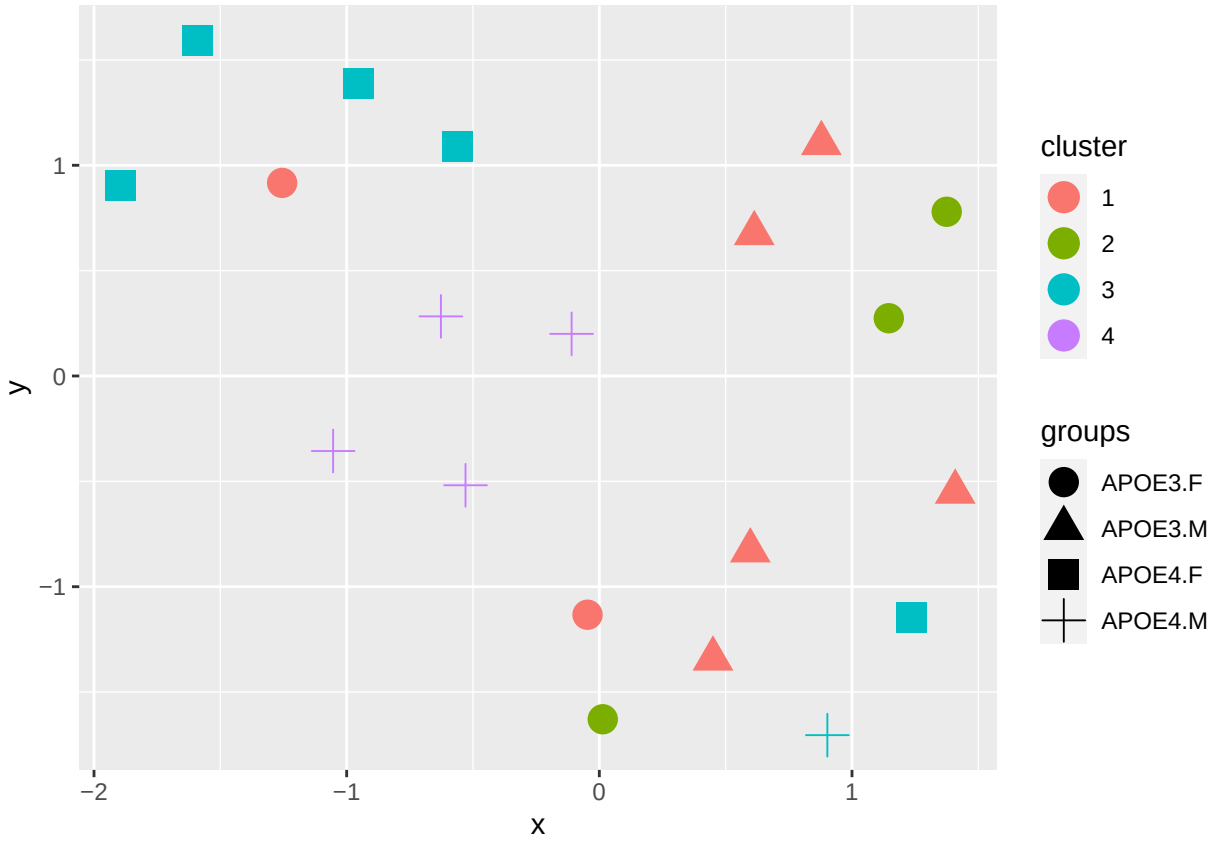



```
## cut tree
clust <- cutree(res$tree_row, k = 4)
table(clust, groups)
```

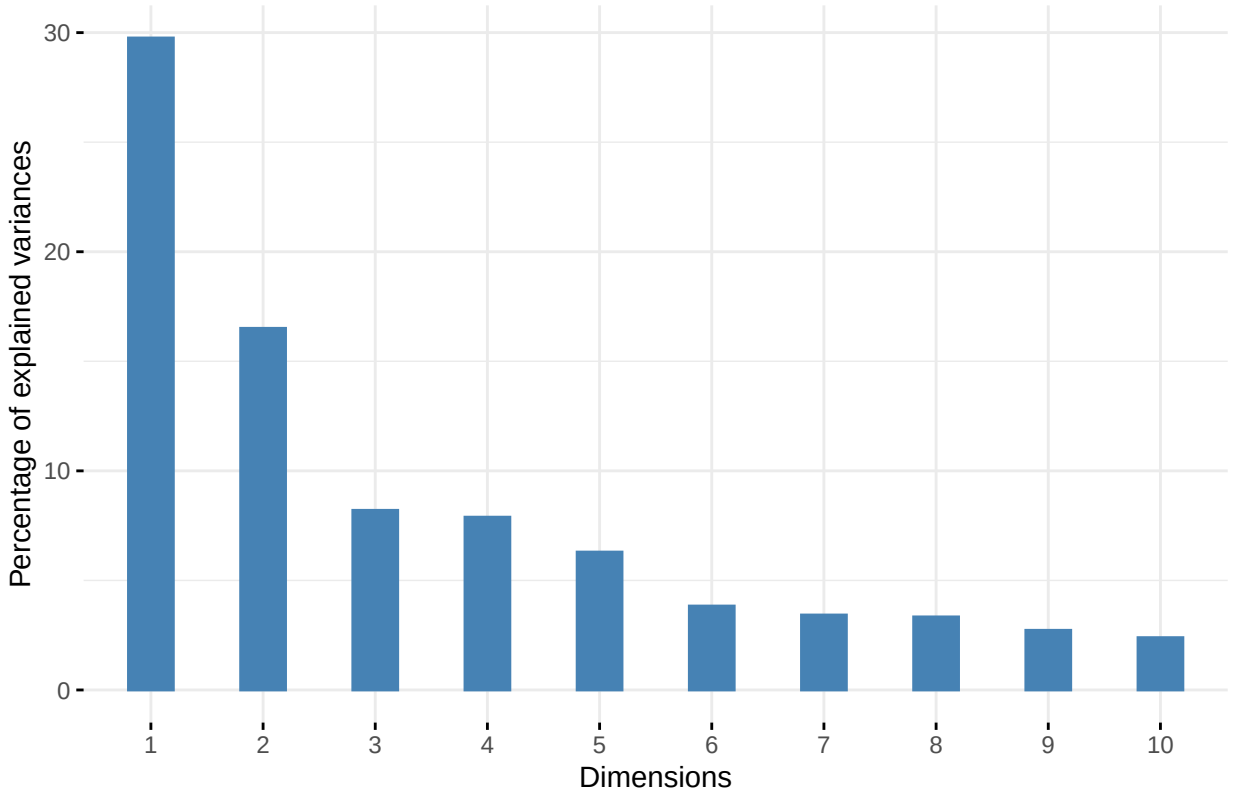
```
##      groups
## clust APOE3.F APOE3.M APOE4.F APOE4.M
##     1       2       5       0       0
##     2       3       0       0       0
##     3       0       0       5       1
##     4       0       0       0       4
```

```
# UMAP
# ------------------------------------------------
expr.log.t = t(expr.log)
um = umap(expr.log.t)
df = data.frame(x = um$layout[,1],
                y = um$layout[,2],
```

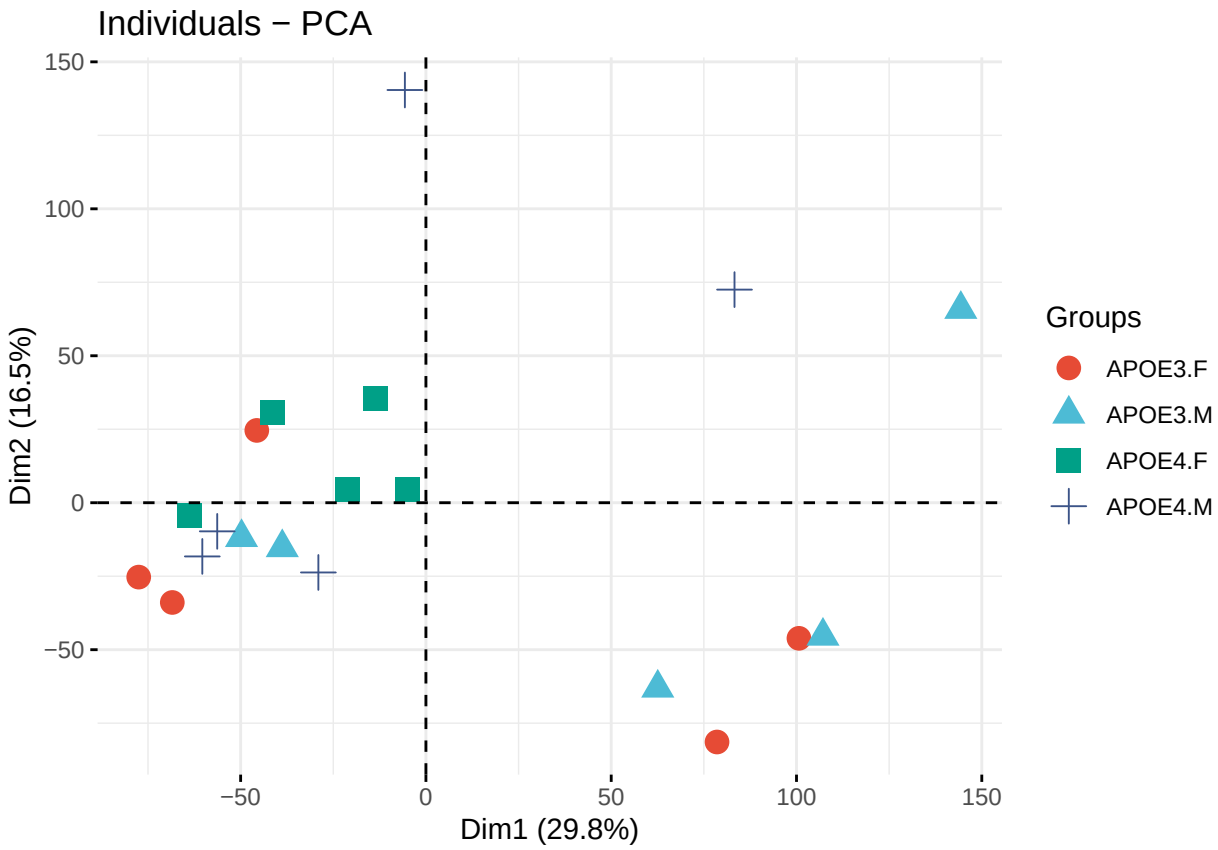
```
                groups = groups)

df$cluster = as.factor(clust)

ggplot(data = df,
       mapping = aes(x = x,
                     y = y,
                     shape = groups,
                     color = cluster)) +
      geom_point(size = 5)
```




```
# PCA
# ------------------------------------------------
pca_out <- prcomp(expr.log.t, center = TRUE, scale = TRUE, retx=TRUE)
fviz_eig(pca_out, geom = "bar", bar_width = 0.4) + ggtitle("")
```




```
fviz_pca_ind(pca_out,
             geom = "point",
             pointsize = 4,
             habillage = groups,
             palette = "npg",
             mean.point = FALSE)
```




```
# Gene Ontology Analysis: Revigo
# ------------------------------------------------
Symbol <- mapIds(org.Mm.eg.db, keys=rownames(expr), keytype="ENSEMBL",column="ENTREZID")
```

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

```
Symbol[which(duplicated(Symbol))] = names(Symbol)[which(duplicated(Symbol))]
Symbol[which(is.na(Symbol))] = names(Symbol)[which(is.na(Symbol))]
fit <- glmFit(expr, design, expr$tagwise.dispersion)
lrt <- glmLRT(fit, contrast=my.contrast[,'APOE3vs4.FvsM'])
lrt.symbol = lrt
rownames(lrt.symbol) = Symbol
go <- goana(lrt.symbol, species = "Mm")
```

```
topgo = topGO(go, n=300, sort='Down', truncate=20)
revigo = data.frame(rownames(topgo),topgo$P.Down)
head(topgo,5)
```

```
##                            Term Ont   N Up Down P.Up       P.Down
## GO:0007076 mitotic chromosom...  BP  14  0    2    1 0.0007374755
## GO:0019827 stem cell populat...  BP 147  0    4    1 0.0008625649
## GO:0098727 maintenance of ce...  BP 151  0    4    1 0.0009533310
## GO:0007346 regulation of mit...  BP 437  0    6    1 0.0015721569
## GO:0045120           pronucleus  CC  21  0    2    1 0.0016799718
```

```
write.table(revigo, file='revigo.txt',row.names=F,col.names=F,quote=F)
```

```
# Choice of normalization
# ------------------------------------------------
```

```r
par(mfrow = c(3, 1))
hist(rowMeans(expr$count),
     main = "Histogram of Raw Mean Expression of Genes",
     xlab = "Expression")

hist(log(rowMeans(expr$count), 2),
     main = "Histogram of Log2-folded Mean Expression of Genes",
     xlab = "Log2(Expression)")

hist(log(rowMeans(expr$count)),
     main = "Histogram of Nature-log-folded Mean Expression of Genes",
     xlab = "Ln(Expression)")
```