Ageing, Dementia and TBI Data Analysis

ZHAO Kai

Here is a brief introduction of analyzing the results of our proposed approach on Ageing, Dementia and TBI dataset.

Preparations

```
suppressPackageStartupMessages({
  require(ggplot2)
  require(formatR)
  require(knitr)
  require(cluster)
  require(factoextra)
  require(dplyr)
  require(RColorBrewer)
  require(clusterProfiler)
  require(org.Hs.eg.db)
  require(enrichplot)
  require(stringr)
  require(forcats)
  require(ggplot2)
  require(ggdendro)
  require(graphics)
  require(gridExtra)
  require(extrafont)
  require(viridis)
  require(hrbrthemes)
})

truncated_var <- function(x){
    remove_idx <- c(which.max(x), which.min(x))
    var(x[-remove_idx])
}

wrap_labal <- function(x, width = 60){
    str_wrap(x, width=60)
}

simes.test <- function(x, returnstat = FALSE){
    r = rank(x,  ties.method = "random")
    t = min(length(x) * x / r)
    if (returnstat) c(t, t) else t
}
```

```
setwd("/Users/zhaokai/data/Results/ageing")

load("ageing_dataset_annotated_with_phenotypes_filtered.RData")
pheno <- dataset[,2]
tissue <- dataset[,3]
dataset <- dataset[, -(1:4)]
# our fitted model
load("insider_ageing_fitted_object_26.RData")
# load("gene_expression_iMF_L1_penalty_25v2.RData")
# attach(fitted_obj)
attach(object)
disease_factor <- cfd_matrices[[1]]
tissue_factor <- cfd_matrices[[2]]
donor_factor <- cfd_matrices[[3]]

# read meta information to facilitate our analyis
meta <- read.csv("meta_info.csv", header = TRUE, stringsAsFactors = F)
gene_info <- read.csv("rows-genes.csv", header = TRUE, stringsAsFactors = F)
structure_info <- unique(read.csv("structure_id_mapping.csv", stringsAsFactors = F)[, c(14, 15)])

structure_info$snames[c(4,7)] <- c("hippocampus_right", "hippocampus_left")

# match gene_info with genes included in the study
gene_included <- data.frame(gene_id = as.numeric(gsub("X", "", colnames(data))))
gene_info_inc <- inner_join(gene_included, gene_info, by = "gene_id")

row.names(tissue_factor) <- structure_info$snames
row.names(disease_factor) <- c("ctrl", "case")
# head to get a sense of our results and meta information
# str(fitted_obj)
head(meta)
```

```
##    donor_id       name age sex apo_e4_allele education_years age_at_first_tbi
## 1 326765665 H14.09.078  87   M             N              16                0
## 2 326765656 H14.09.069  97   M             N              17               12
## 3 326765654 H14.09.067  85   M             Y              10               72
## 4 467056391 H15.09.103  92   F             N              11               87
## 5 309335447 H14.09.010 100   M             Y              16                0
## 6 309335457 H14.09.020  97   F             N              18                0
##   longest_loc_duration cerad num_tbi_w_loc dsm_iv_clinical_diagnosis
## 1       Unknown or N/A     0             0                No Dementia
## 2              1-2 min     2             1                No Dementia
## 3            < 10 sec      3             1                   Vascular
## 4            < 10 sec      0             1                No Dementia
## 5       Unknown or N/A     3             0  Alzheimer's Disease Type
## 6       Unknown or N/A     2             0                No Dementia
##   control_set        nincds_arda_diagnosis ever_tbi_w_loc      race
## 1          31                  No Dementia              N     White
## 2          26                  No Dementia              Y     White
## 3          25       Dementia, Type Unknown              Y     White
## 4          52                  No Dementia              Y     White
## 5          28 Possible Alzheimer'S Disease              N     White
## 6           1                  No Dementia              N Non-white
##       hispanic act_demented braak nia_reagan
```

```
## 1 Not Hispanic  No Dementia     1          1
## 2 Not Hispanic  No Dementia     5          2
## 3 Not Hispanic     Dementia     4          2
## 4 Not Hispanic  No Dementia     4          0
## 5 Not Hispanic     Dementia     4          2
## 6 Not Hispanic  No Dementia     3          2
```

```
head(gene_info)
```

```
##     gene_id chromosome gene_entrez_id  gene_symbol
## 1 499304660          1      100287102      DDX11L1
## 2 499304661          1         653635       WASH7P
## 3 499304662          1      102466751    MIR6859-1
## 4 499304663          1      100302278    MIR1302-2
## 5 499304664          1         645520      FAM138A
## 6 499304665          1      105379212 LOC105379212
##                                           gene_name
## 1 DEAD/H (Asp-Glu-Ala-Asp/His) box helicase 11 like 1
## 2             WAS protein family homolog 7 pseudogene
## 3                                     microRNA 6859-1
## 4                                     microRNA 1302-2
## 5       family with sequence similarity 138, member A
## 6                         uncharacterized LOC105379212
```

Cluster analysis of donor_factor

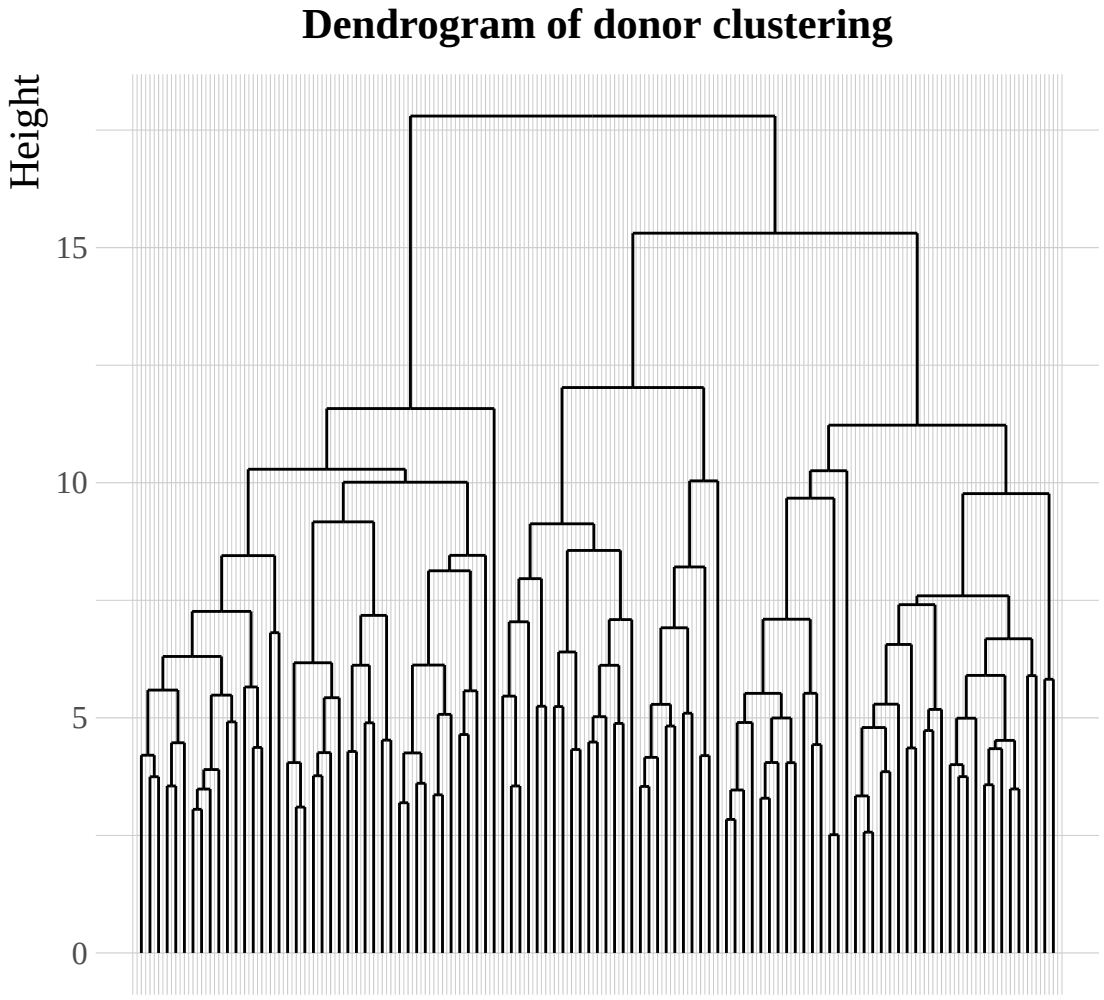
The donor_factor is a low rank representation of genetic information from the dementia dataset, which is a matrix of N rows and K columns. N is the number of donors and K is the number of latent features representing gene expression information in low dimensions. In this section, the cluster analysis basically follows this tutorial.

```
# for detail, see https://uc-r.github.io/hc_clustering methods to assess
m <- c( "average", "single", "complete", "ward")
names(m) <- c( "average", "single", "complete", "ward")

# function to compute coefficient
ac <- function(x) {
  agnes(donor_factor, method = x)$ac
}
ac_vec <- sapply(m, function(x) ac(x))
ac_vec
```

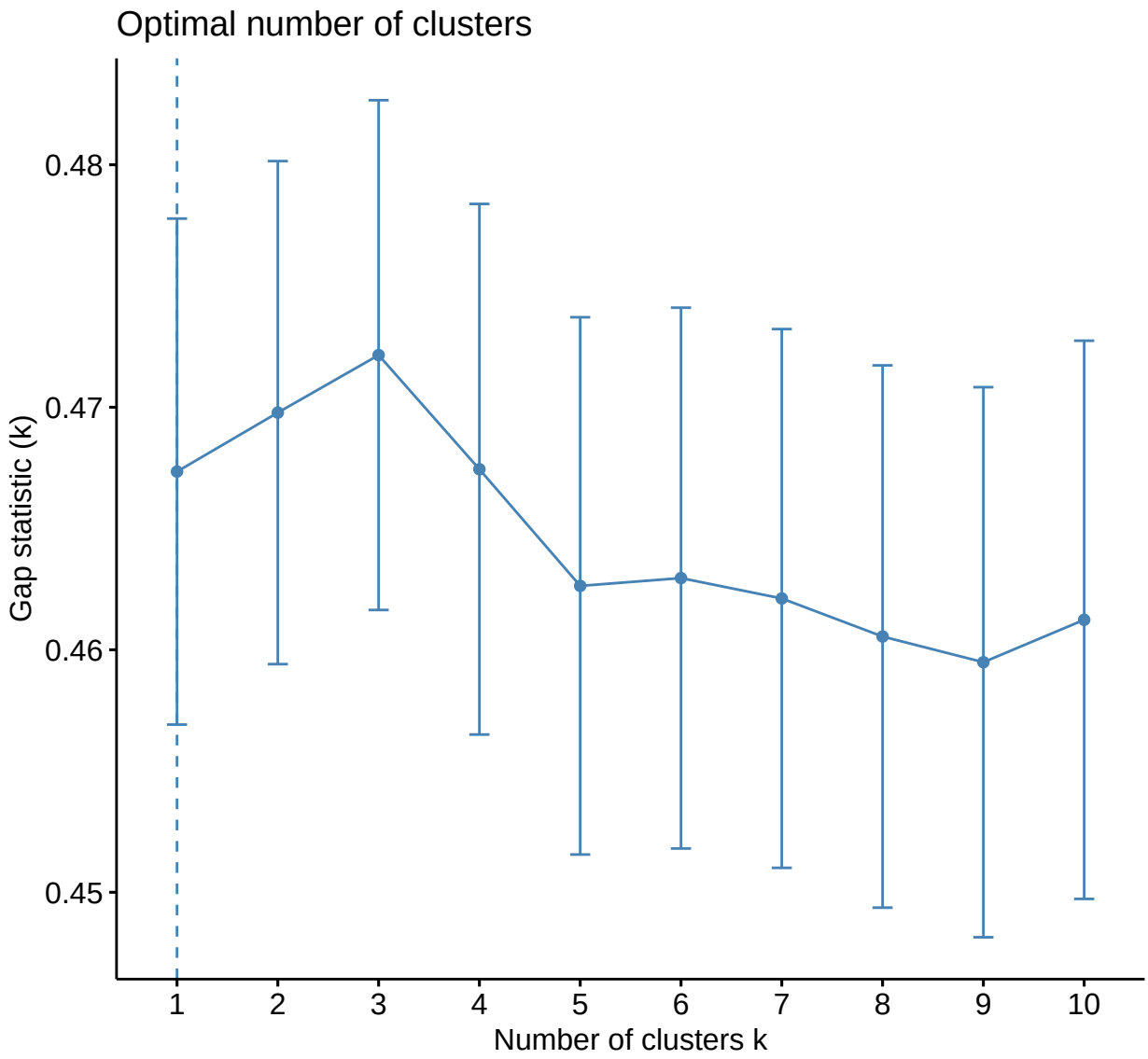
```
##   average    single  complete      ward
## 0.5360925 0.4549641 0.6314432 0.7452917
```

```
# carry out hierarchical cluster analysis using the method with the greatest coefficient
# hc3 <- agnes(donor_factor, method = "ward")
hc3 <- agnes(donor_factor, method = unname(m[which(ac_vec == max(ac_vec))]))
# pltree(hc3, cex = 0.6, hang = -1, main = "Dendrogram of donor clustering", labels = NULL)
hc <- as.hclust(hc3)
ggdendrogram(hc, theme_dendro = FALSE) +
  ggtitle("Dendrogram of donor clustering") + ylab("Height") +
  theme_ipsum(base_family = "Times New Roman", base_size= 12, plot_title_face = "bold", axis_title_size
  theme(plot.title = element_text(hjust = 0.5, size=16,face="bold"),
        text=element_text(family="Times New Roman"),
```

```
        axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())
```




```
# par(cex=0.4, mar=c(5, 8, 4, 1))
# plot(hc, ylab = "Height", main = "Dendrogram of donor clustering", label = F, hang = -1, xlab = "", s
# select cluster number using Gap statistics
gap_stat <- clusGap(donor_factor, FUN = hcut, nstart = 25, K.max = 10, B = 200)
fviz_gap_stat(gap_stat)
```




```
# choose 3 clusters by investigating Dendrogram and Gap statistics plots
sub_grp <- cutree(hc3, k = 3)
result <- cbind(meta, cluster_id = sub_grp)
aov_res <- aov(age ~ cluster_id, data = result)
unname(unlist(summary(aov_res)))[9]
```

```
## [1] 0.01310313
```

```
kruskal.test(age ~ cluster_id, data = result)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  age by cluster_id
## Kruskal-Wallis chi-squared = 7.9266, df = 2, p-value = 0.019
```

```
fisher.test(table(result$cluster_id, result$braak), simulate.p.value=TRUE)
```

```
##
##  Fisher's Exact Test for Count Data with simulated p-value (based on
```

```
##  2000 replicates)
##
## data:  table(result$cluster_id, result$braak)
## p-value = 0.2054
## alternative hypothesis: two.sided
```

In this part, we exclude a number of metagenes that is irrelevant to synapse function, cognition, and memory. The selection of metagenes is somewhat subjective. However, we observe that this strategy shows some interesting results.

```
# ord <- order(apply(disease_factor, 2, function(x) abs(x[2] - x[1])), decreasing = F) # (1, 5, 6, 9, 1

# result <- sapply(ord, function(i){
#   hc3 <- agnes(donor_factor[, -c(i, 10, 18, 7,6)], method = "ward")

#   sub_grp <- cutree(hc3, k = 3)
#   result <- cbind(meta, cluster_id = sub_grp)
#   aov_res <- aov(age ~ cluster_id, data = result)
#   print(unname(unlist(summary(aov_res)))[9])
# })



# hc <- as.hclust(hc3)
# ggdendrogram(hc, theme_dendro = FALSE) +
#   ggtitle("Dendrogram of donor clustering") + ylab("Height") +
#   theme_ipsum(base_family = "Times New Roman", base_size= 12, plot_title_face = "bold", axis_title_si
#   theme(plot.title = element_text(hjust = 0.5, size=16,face="bold"),
#         text=element_text(family="Times New Roman"),
#         axis.title.x=element_blank(),
#         axis.text.x=element_blank(),
#         axis.ticks.x=element_blank())

# sub_grp <- cutree(hc3, k = 3)
# result <- cbind(meta, cluster_id = sub_grp)
# aov_res <- aov(age ~ cluster_id, data = result)
# unname(unlist(summary(aov_res)))[9]

# fisher.test(table(result$cluster_id, result$nincds_arda_diagnosis)[,-1])
# fisher.test(table(result$cluster_id, result$dsm_iv_clinical_diagnosis)[,-c(4,5)])
```

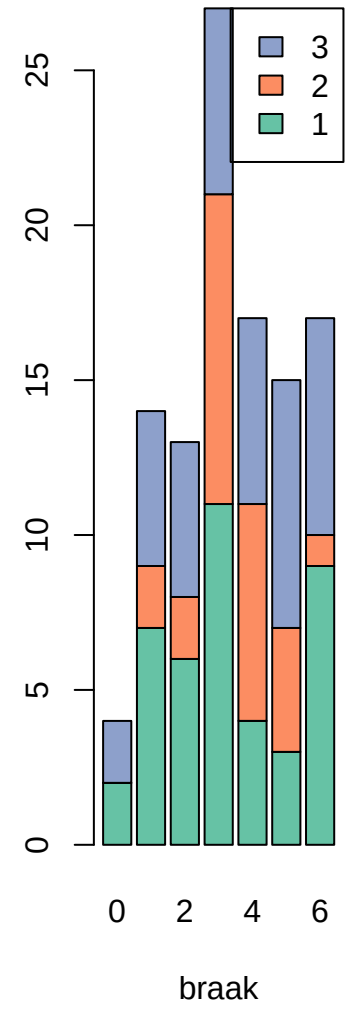
explore the relavance of the clustering

In this section, we only examined the association between the age of donors and their clusters. Explorations of the relavance of the clustering to other clinical variables can also be carried out. Pie charts and histograms can be drawn to visualize the association. Furthermore, some statistical tests can also be used to check the significance. The result below shows that there is a statistical significant association between the age and clustering.

```
# result <- cbind(meta, cluster_id = sub_grp)
# head(result)
# aov_res <- aov(age ~ cluster_id, data = result)
# summary(aov_res)
# kruskal.test(age ~ cluster_id, data = result)
```

The below histogram show the dribution of clusters cross different braak stages, which are clinical diagnoses of stage of dementia.

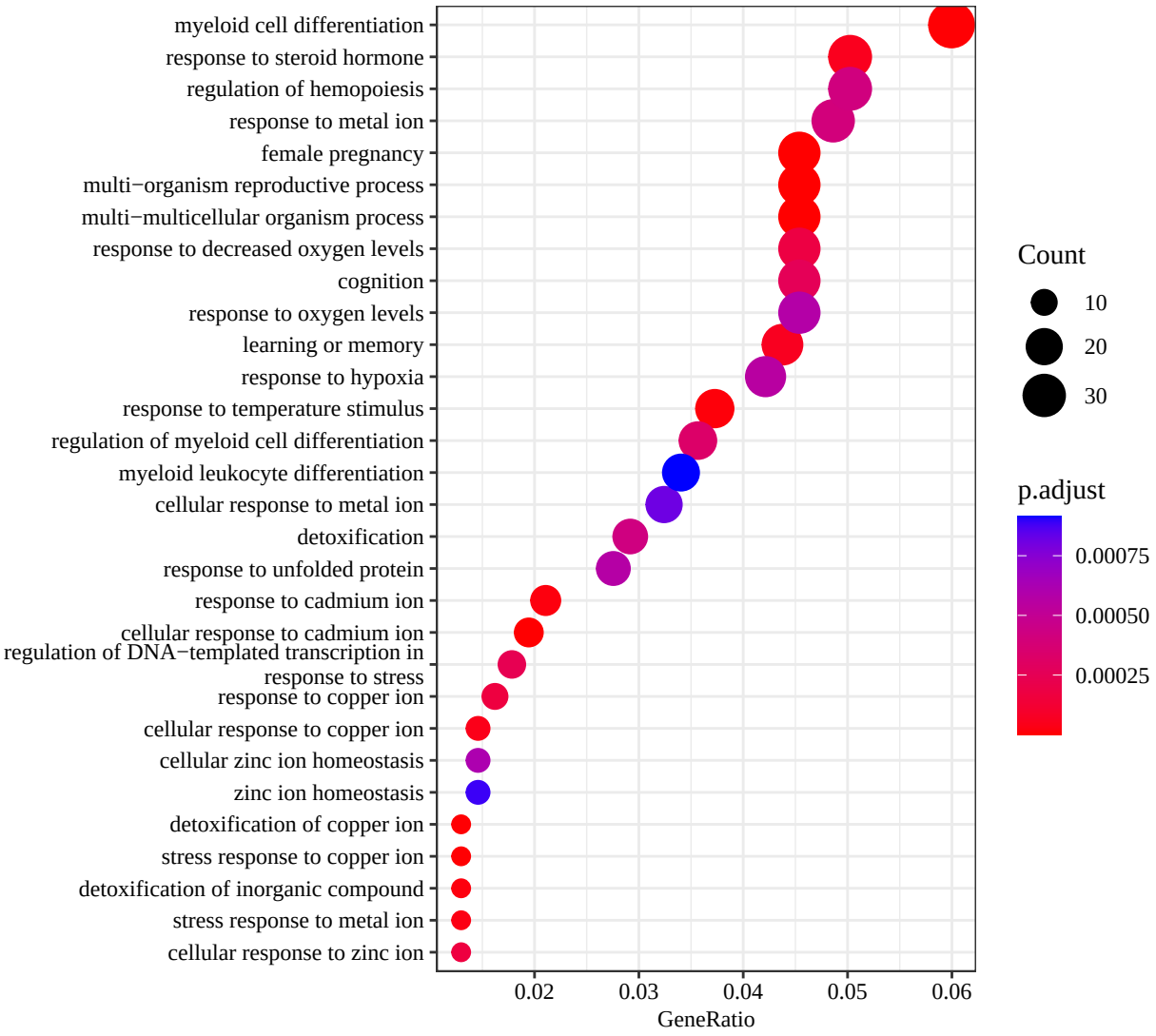
```
# cluster_sex <- table(result$cluster_id, result$sex)
# par(mar=c(5, 13, 3, 13))
# barplot(cluster_sex, main="number of observations",
#   xlab="SEX", col= brewer.pal(3, "Set2") ,
#   legend = rownames(cluster_sex), space = 0.2, width = 0.2,
#   args.legend = list(x = "topleft"))

cluster_tbi <- table(result$cluster_id, result$braak)
par(mar=c(5, 13, 3, 13))
barplot(cluster_tbi, main="number of observations",
  xlab="braak", col= brewer.pal(3, "Set2") ,
  legend = rownames(cluster_tbi), space = 0.2, width = 0.2,
  args.legend = list(x = "toprigh"))
```



Enrichment analysis of biological processes involved

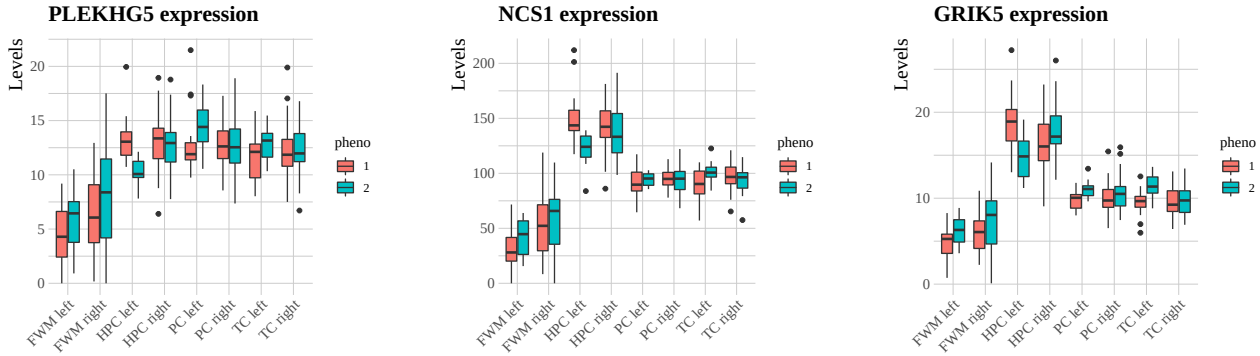
We can also investigate biological processes we are interested in with our results. For example, we explored the mechanism of dementia with the disease and gene factors. We obtained the expression profiles for the dementia and control, extracted the genes with greatest postive difference between them, and examine the biological processes enriched by those genes.

```
idx <- order(apply(disease_factor, 2, var), decreasing = T)[1:2]  # 1, 4, 5?, 8, 9,14,15,18,20,23,26
disease_matrix <- disease_factor[,idx, drop = F] %*% column_factor[idx, , drop = F]
diff <- disease_matrix[2,] - disease_matrix[1,]
cutoffs <- quantile(diff, probs = seq(0, 1, 0.025))
selected <- (diff <= cutoffs[2]) # greatest negative difference
# selected <- (diff >= cutoffs[length(cutoffs)-1])  # greatest positive difference

# head(gene_info[selected,3])
upreg <- enrichGO(gene      = unique(gene_info_inc[selected,3]),
                  OrgDb     = 'org.Hs.eg.db',
                  ont       = "BP",
                  readable  = TRUE)

dotplot(upreg, font.size = 9, showCategory=30, label_format = 50) +
  theme(text=element_text(family="Times New Roman"))
```




```
loadfonts(quiet = T)

structure_names <- c("TC left", "FWM right", "FWM left", "HPC right", "PC right", "TC right", "HPC left

# result <- as.data.frame(upreg)
# gene_names <- unlist(strsplit(result[1,8], split = "/"))

selected <- c("PLEKHG5", "NCS1", "GRIK5", "CACNG3")

row_ids <- unlist(sapply(selected, function(x) which(gene_info_inc[[4]] == x)))
data <- cbind(pheno, structure_names[tissue], dataset[,row_ids])
colnames(data) <- c("pheno", "tissue", names(row_ids))
data$pheno <- as.factor(data$pheno)

# boxplots and t-tests for the 4 variables at once
test_results <- sapply(3:ncol(data), function(j){
    pvalues <- sapply(structure_names, function(i) t.test(data[data$tissue == i, j] ~ data$pheno[data$t
})
```

```
simes_pvalues <- apply(test_results, 2, function(x)simes.test(x))
names(simes_pvalues) <- names(row_ids)

colnames(test_results) <- names(row_ids)
rownames(test_results) <- structure_names

# selected <- c("PLEKHG5", "NCS1", "GRIK5", "CACNG3")
# par(mfrow = c(1, 3))
p1 <- ggplot(data, aes(x=tissue, y=PLEKHG5, fill=pheno)) +
    scale_color_viridis(discrete = TRUE) +
    geom_boxplot() +
    ggtitle("PLEKHG5 expression") +
    xlab("") + ylab("Levels") +
    theme_ipsum(base_family = "Times New Roman", base_size= 12, plot_title_face = "bold", axis_title_si
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))

p2 <- ggplot(data, aes(x=tissue, y=NCS1, fill=pheno)) +
    scale_color_viridis(discrete = TRUE) +
    geom_boxplot() +
    ggtitle("NCS1 expression") +
    xlab("") + ylab("Levels") +
    theme_ipsum(base_family = "Times New Roman", base_size= 12, plot_title_face = "bold", axis_title_si
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))

p3 <- ggplot(data, aes(x=tissue, y=GRIK5, fill=pheno)) +
    scale_color_viridis(discrete = TRUE) +
    geom_boxplot() +
    ggtitle("GRIK5 expression") +
    xlab("") + ylab("Levels") +
    theme_ipsum(base_family = "Times New Roman", base_size= 12, plot_title_face = "bold", axis_title_si
    theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))

# p4 <- ggplot(data, aes(x=tissue, y=CACNG3, fill=pheno)) +
#     scale_color_viridis(discrete = TRUE) +
#     geom_boxplot() +
#     ggtitle("NCS1 expression") +
#     xlab("") + ylab("Levels") +
#     theme_ipsum(base_family = "Times New Roman", base_size= 12, plot_title_face = "bold", axis_title_
#     theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))

grid.arrange(p1, p2, p3, nrow=1)
```



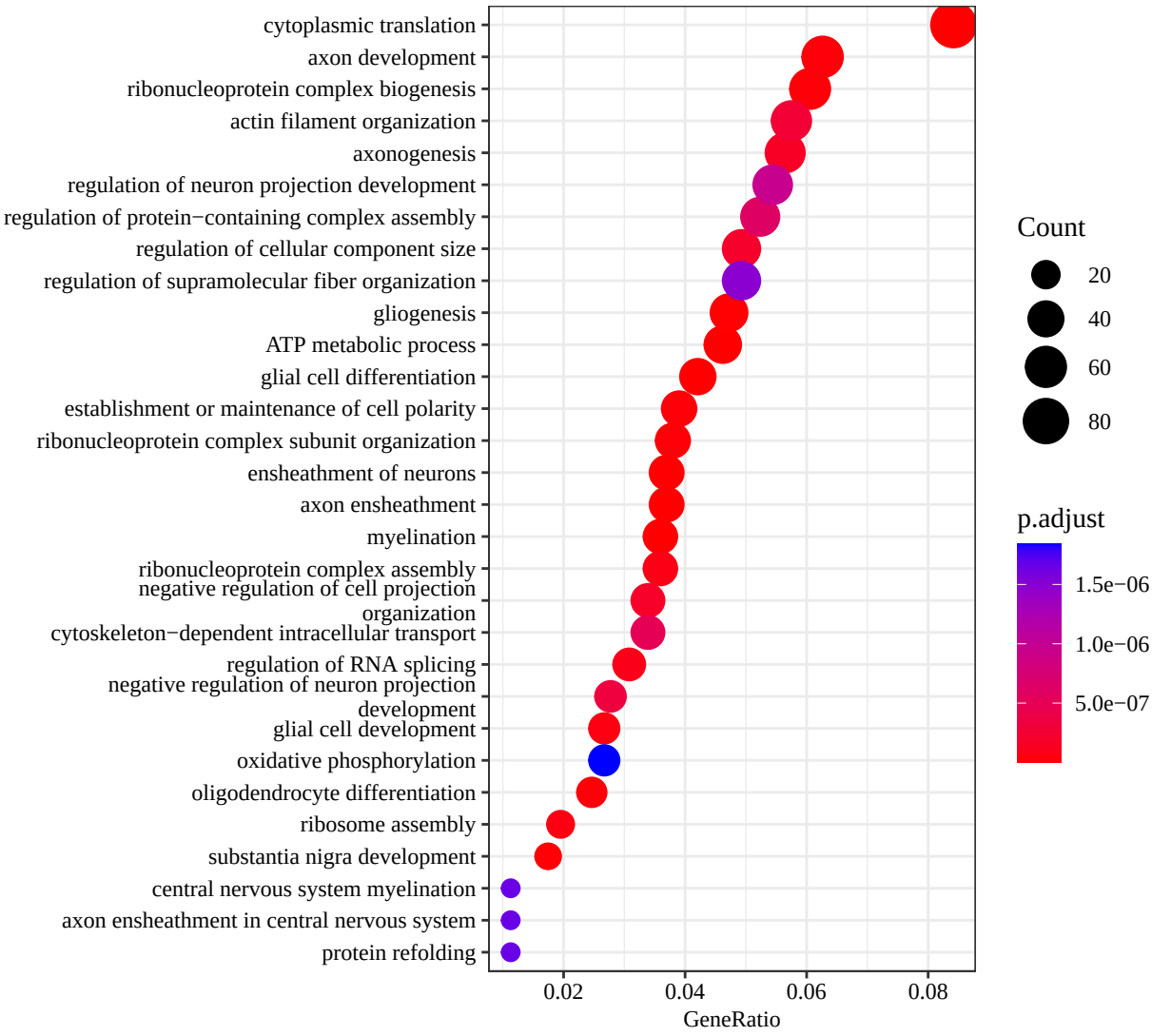
Here is a another pieces of code to demenstrate molecular functions of genes with the largest effects in different brain structures.

In this part of analysis, I only demonstrate with the second tissue. In order to expand, analysis of other metagenes with a single for loop is fine to generate all results.

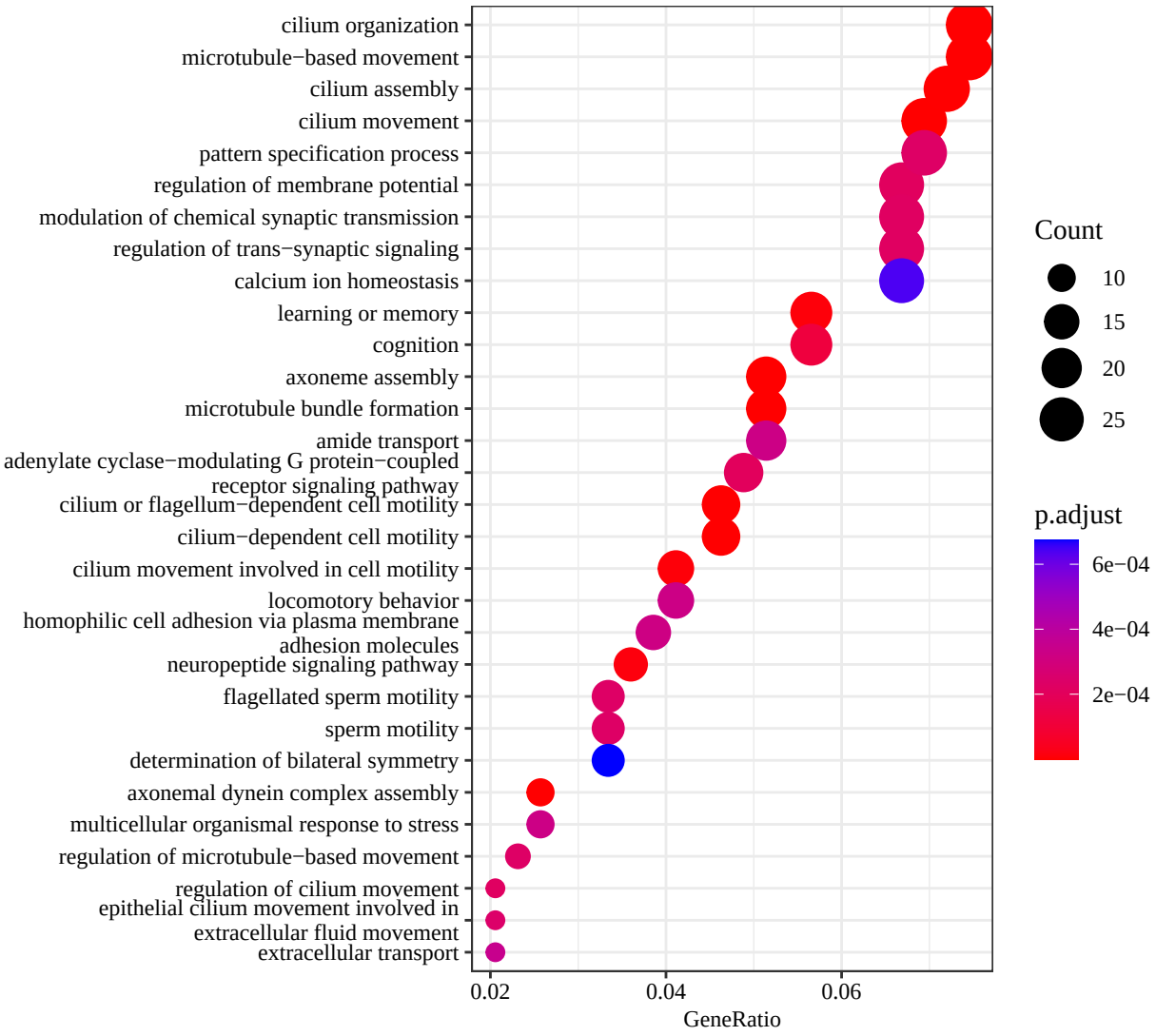
```
idx <- order(apply(tissue_factor, 2, function(x) truncated_var(x)))[1:4]
tissue_matrix <- tissue_factor[,-idx] %*% column_factor[-idx, ]
row.names(tissue_matrix) <- structure_info$snames
id <- 2
cat("tissue name:", row.names(tissue_matrix)[id], "\n")
```

```
## tissue name: white matter of forebrain_right
```

```
cutoffs <- quantile(tissue_matrix[id, ], probs = seq(0, 1, 0.025))
# selected <- (tissue_matrix[id,] <= cutoffs[2]) # greatest negative difference
selected <- (tissue_matrix[id, ] >= cutoffs[length(cutoffs)-1])  # greatest positive difference

# head(gene_info[selected,3])
up_reg <- enrichGO(gene      = unique(gene_info_inc[selected,3]),
                   OrgDb     = 'org.Hs.eg.db',
                   ont       = "BP",
                   readable  = TRUE)

dotplot(up_reg, font.size = 9, showCategory=30, label_format = 50) +
  theme(text=element_text(family="Times New Roman"))
```




```
selected <- (tissue_matrix[id,] <= cutoffs[2]) # greatest negative difference
down_reg <- enrichGO(gene     = unique(gene_info_inc[selected,3]),
                     OrgDb    = 'org.Hs.eg.db',
                     ont      = "BP",
                     readable  = TRUE)

dotplot(down_reg, font.size = 9, showCategory=30, label_format = 50) +
  theme(text=element_text(family="Times New Roman"))
```



Furthermore, we could also examine the interaction between tissue and disease factors. The code below explores the interaction between different tissues and dementia. Then, similar techniques can be employed to examine the contribution of underlying biological processes to the interaction.

In this part of analysis, I only demonstrate with the parietal neocortex(right). In order to expand, analysis of temporal neocortex(right) and white matter of forebrain(left) with a single for loop is enough.

```
# # since all latent vectors are restricted in the same space, we can compute the correlation between d
# row.names(tissue_factor) <- structure_info$snames
# scores <- cor(t(tissue_factor[,-idx[1]]), t(disease_factor[, -idx[1]]))
# print(scores)

# # then we can examine the tissue with the largest change in correlation
# tissue_id <- 3
# cat("Tissue name:", rownames(tissue_factor)[tissue_id], "\n")

# interaction <- t(apply(disease_factor[,-idx[1]], 1, function(x) x * tissue_factor[tissue_id, -idx[1]]
# interaction_matrix <- interaction %*% column_factor[-idx[1], ]
# diff <- interaction_matrix[2,] - interaction_matrix[1,]
```

```
# cutoffs <- quantile(diff, probs = seq(0, 1, 0.025))

# # up-regulation, greatest positive difference
# # selected <- (diff >= cutoffs[length(cutoffs)-1])
# # upreg <- enrichGO(gene      = unique(gene_info_inc[selected,3]),
# #                   OrgDb     = 'org.Hs.eg.db',
# #                   ont       = "BP",
# #                   readable  = TRUE)

# # dotplot(upreg, font.size = 9, showCategory=30, label_format = 50) +
# #   theme(text=element_text(family="Times New Roman"))

# # up-regulation, greatest negative difference
# selected <- (diff <= cutoffs[2])
# downreg <- enrichGO(gene    = unique(gene_info_inc[selected,3]),
#                     OrgDb     = 'org.Hs.eg.db',
#                     ont       = "BP",
#                     readable  = TRUE)

# dotplot(downreg, font.size = 9, showCategory=30, label_format = 50) +
#   theme(text=element_text(family="Times New Roman"))
```