Brainspan Data Analysis

ZHAO Kai

Here is a brief introduction of analyzing the results of our proposed approach on Brainspan 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(DOSE)
  require(ggplot2)
  require(hrbrthemes)
  require(viridis)
  require(reshape2)
  require(gridExtra)
  require(extrafont)
})

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)
}
```

```
glm_interaction <- function(object, inc_cfd){

  residual <- object[['data']]

  confounder_num <- ncol(object[['confounder']])
  for(i in 1:confounder_num){
    sub_predictions <- object[['cfd_matrices']][[i]] %*% object[['column_factor']]
    residual <- residual - sub_predictions[object[['confounder']][,i], ]
  }
```

```
  column_factor <- object[['column_factor']]
  train_indicator <- object[['train_indicator']]

  confounder <- object[['confounder']][, inc_cfd]
  unique_cfd <- unique(confounder)

  interaction_indicator <- rep(0, nrow(confounder))
  for(k in 1:nrow(unique_cfd)){
    selected <- apply(confounder, 1, function(x) all(x == unique_cfd[k,]))
    interaction_indicator[selected] <- k
  }

  unique_ita <- unique(interaction_indicator)
  coeff_matrix <- matrix(0, nrow = length(unique_ita), ncol = nrow(column_factor))
  pval_matrix <- matrix(0, nrow = length(unique_ita), ncol = nrow(column_factor))

  for(i in unique_ita) {

    ids <- which(interaction_indicator == i);

    st_idx <- 1; ed_idx <- 1
    nonzero_num <- length(ids) * ncol(column_factor);
    outcomes = rep(0,nonzero_num);
    features = matrix(0, nrow = nonzero_num, ncol = nrow(column_factor))

    for(k in ids){
      ed_idx = st_idx + ncol(column_factor) - 1;
      features[st_idx:ed_idx, ] = t(column_factor);
      outcomes[st_idx:ed_idx] = residual[k,];
      st_idx = ed_idx + 1
    }

    data <- data.frame(response = outcomes, features)
    fit <- glm(response ~ . - 1, family = gaussian(), data = data)
    coeff_matrix[i,] <- unname(coefficients(fit))
    pval_matrix[i,] <- coef(summary(fit))[,4]
  }
  return(list(unique_cfd, coeff_matrix, pval_matrix))
}
```

```
opts_chunk$set(tidy.opts=list(width.cutoff=80),tidy=TRUE)

setwd("~/data/multidimensional_datasets/brainspan_genes_matrix_csv/")
# load results for brain span
# load("~/data/Results/brainspan/insider_brainspan_fitted_object.RData")
load("~/data/Results/brainspan/insider_brainspan_R23_fitted_object.RData")
attach(object) # attach it for easy syntax
str(object) # show the structure of our result
```

```
## List of 7
##  $ data           : num [1:524, 1:43411] 5.23 4.66 4.35 4.84 4.39 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:524] "V2" "V3" "V4" "V5" ...
##   .. ..$ : chr [1:43411] "ENSG00000000003" "ENSG00000000005" "ENSG00000000419" "ENSG00000000457" ...
```

```
##  $ confounder     : num [1:524, 1:3] 1 1 1 1 1 1 1 1 1 1 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:524] "V2" "V3" "V4" "V5" ...
##   .. ..$ : chr [1:3] "preriod_id" "sid" "interaction_indicator"
##  $ train_indicator: int [1:524, 1:43411] 1 0 1 1 1 1 1 1 1 1 ...
##  $ params         :List of 4
##   ..$ global_tol : num 1e-10
##   ..$ sub_tol    : num 1e-05
##   ..$ tuning_iter: num 30
##   ..$ max_iter   : num 50000
##  $ cfd_matrices   :List of 3
##   ..$ factor0: num [1:13, 1:23] -0.0708 -1.1885 -0.9303 -1.3672 -2.0438 ...
##   ..$ factor1: num [1:26, 1:23] 0.0646 0.2225 0.6459 0.1011 -0.1981 ...
##   ..$ factor2: num [1:205, 1:23] 0.0646 -0.0855 -0.6158 0.1011 -0.3536 ...
##  $ column_factor  : num [1:23, 1:43411] -0.0409 -0.0624 0.1557 -0.0132 0.0476 ...
##  $ test_rmse      : num 4.68e-310
##  - attr(*, "class")= chr "insider"
```

```r
stage_factor <- cfd_matrices[[1]]
tissue_factor <- cfd_matrices[[2]]
interactions <- cfd_matrices[[3]]

# read meta information
dic <- read.csv("~/data/Results/brainspan/dictionary.csv", stringsAsFactors = F)
# obtain ensemble genes included in our study
load("brainspan_dataset_annotated_fitered.RData")
gene_id <- data.frame(ensembl_gene_id = colnames(data), stringsAsFactors =F)
# match the included genes with meta information
row_meta <- read.csv('rows_metadata.csv', stringsAsFactors = F)
meta <- inner_join(gene_id, row_meta, by = "ensembl_gene_id")

# prepare struture and stage names for naming corresponding latent factors
structure <- unique(dic[,c(6, 9)])
structure <- structure[order(structure[,2]),]
stage <- unique(dic[,c(11, 12)])
r_names <- apply(stage, 1, function(x) paste0(x[2], "_", trimws(x[1])))

# name tissue_factor and stage_factor
rownames(tissue_factor) <- structure[,1]
rownames(stage_factor) <- r_names
```

Explore development trajectory across entire lifespan

Here we explore the development trajectory with the latent representations of development stage factors. In the below exmaple, we selected the metagene with the largest variation across entire lifespan, visualized the trajectory of this metagene across all development stages, and explore the pathway enriched for top 2.5% genes that contribute to this metagene.

In this part of analysis, I only demonstrate with the metagens with greatest and smallest variance. In order to expand, analysis of other metagenes with a single for loop is fine to generate results.

```r
# compute the variance for each metagene
matagene_var <- apply(stage_factor, 2, var)
ord <- order(matagene_var, decreasing = TRUE)
```

```
stage_factor[, ord[1:3]]
```

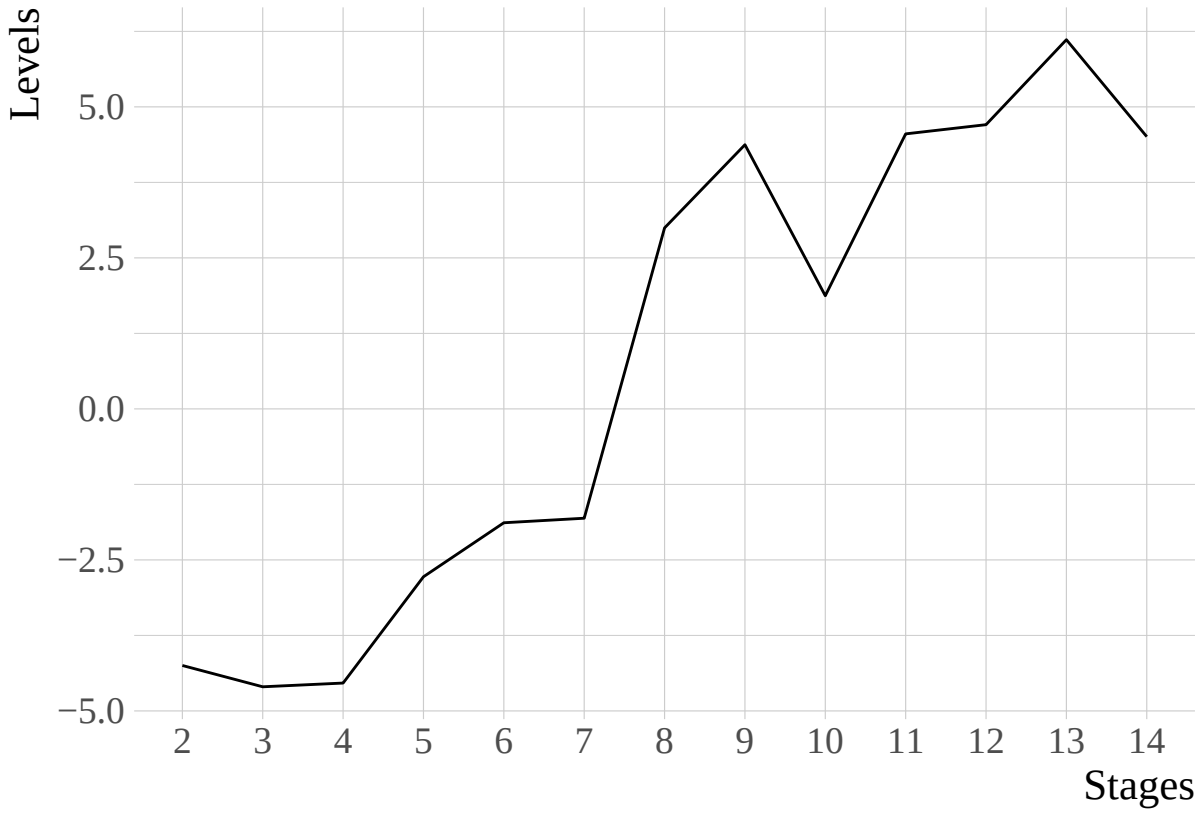
```
##                                    [,1]       [,2]       [,3]
## Early fetal_2                -4.247888 -1.2303545  0.2714118
## Early fetal_3                -4.600395 -2.5115801 -0.5359160
## Early mid-fetal_4            -4.538718 -2.8555911 -0.3498056
## Early mid-fetal_5            -2.779214 -1.5564732 -0.6081612
## Late mid-fetal_6             -1.885085 -1.8780150 -1.0547725
## Late fetal_7                 -1.810105  3.4581501 -0.8537005
## Neonatal and early infancy_8  2.996511  4.8831190 -1.5279612
## Late infancy_9                4.373334  1.4912690 10.2505781
## Early childhood_10            1.873393  7.0089109 -0.4774560
## Middle and late childhood_11  4.552819  3.3429458 -0.1978148
## Adolescence_12                4.705181  5.0872292 -1.2346552
## Young adulthood_13            6.111791  3.0808813 -0.9593607
## Middle adulthood_14           4.507083  0.8578252 -2.8051892
```

```
# use the most variably metagene as an example
metagene_id <- ord[1]
cat("Column_id:", metagene_id, "\n")
```

```
## Column_id: 17
```

The plot below show the trajectory of the selected metagene cross all development stages.

```
loadfonts(quiet = T)
result <- data.frame(stage = r_names, levels = stage_factor[, metagene_id], stringsAsFactors = F)
result$stage <- factor(r_names, levels = r_names)
# ggplot(data = result, aes(x = stage, y = levels, group = 1)) +
# geom_line(linetype = 'dashed') + geom_point() + xlab('Stages') +
# ylab('Levels') + theme(plot.title = element_text(size=12, face = 'bold',
# hjust = 0.5), axis.title.y = element_text(size=10),
# text=element_text(size=10, family='Times New Roman'), axis.text.x =
# element_text(size=10,angle = 45, vjust = 1, hjust=1))
ggplot(data = result, aes(x = stage, y = levels, group = 1)) + scale_color_viridis(discrete = T) +
    scale_x_discrete(labels = 2:14) + geom_line() + theme_ipsum(base_family = "Times New Roman",
    base_size = 14, axis_title_size = 16) + xlab("Stages") + ylab("Levels")
```




```
# axis.text.x = element_text(size=10,angle = 45, vjust = 1, hjust=1))
```

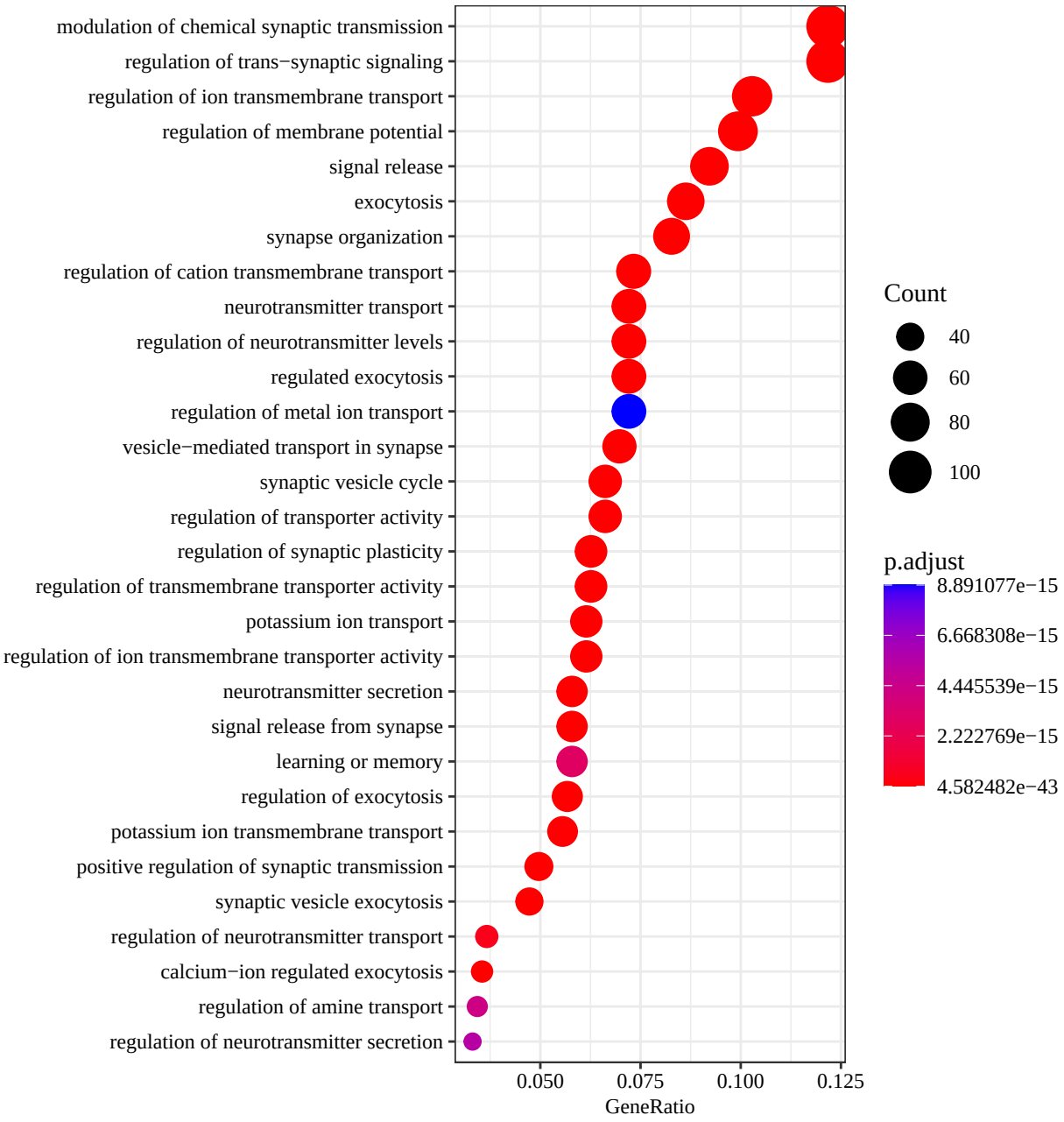
Then, we investigated the pathway enriched for top 2.5% genes that up-regulates and down-regulates this metagene.

```
cat("Column_id:", metagene_id, "\n")
```

```
## Column_id: 17
```

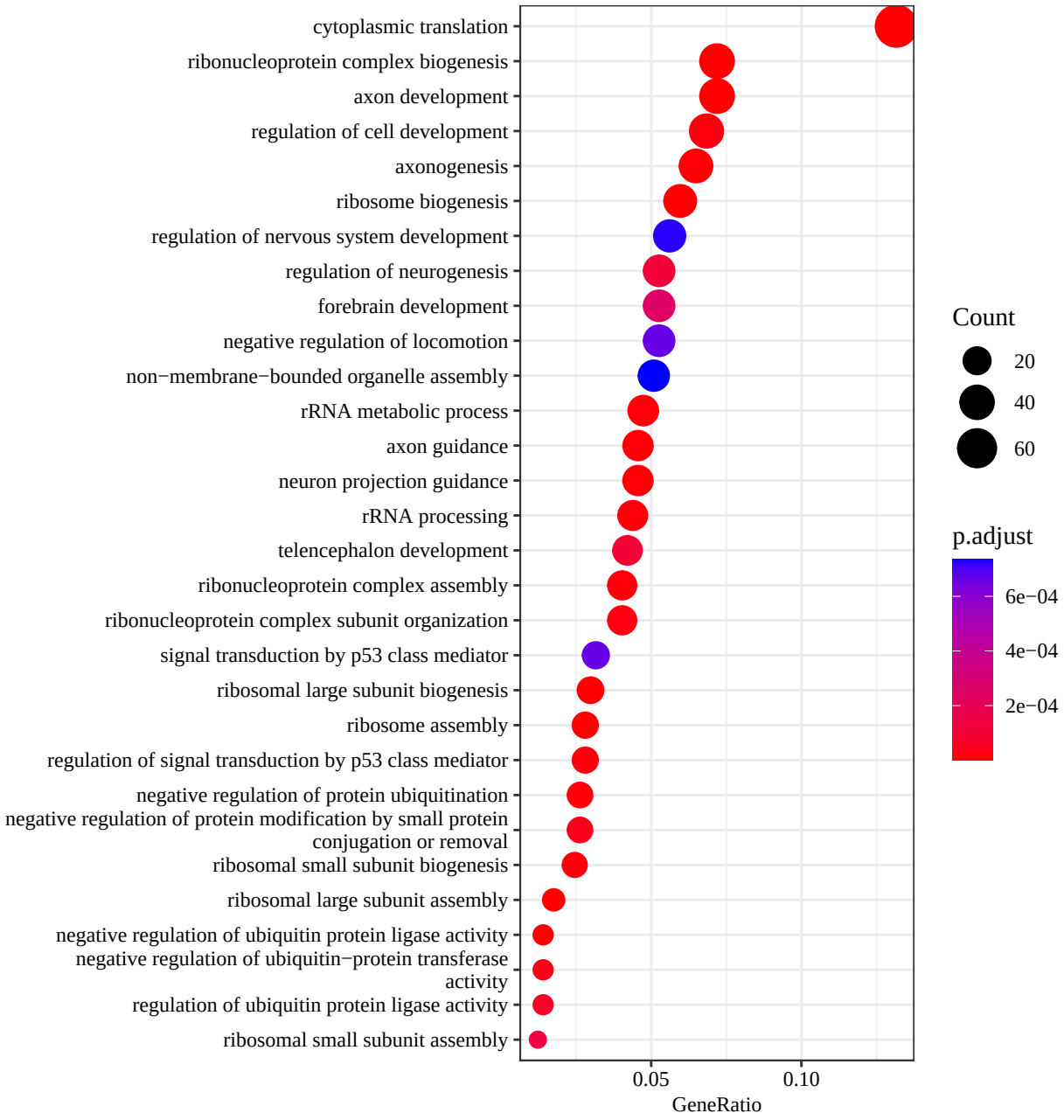
```
cutoffs <- quantile(column_factor[metagene_id, ], probs = seq(0, 1, 0.025))

# up-regulation, select the highest quantile
selected <- (column_factor[metagene_id, ] >= cutoffs[length(cutoffs) - 1])  #
upreg <- enrichGO(gene = unique(meta[selected, 5]), OrgDb = "org.Hs.eg.db", ont = "BP",
    readable = TRUE)
dotplot(upreg, font = 9, showCategory = 30) + scale_y_discrete(labels = function(x) str_wrap(x,
    width = 60)) + theme(text = element_text(family = "Times New Roman"))
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```




```
# the object from enrichGO can be converted to data frame with the following.
# result <- data.frame(upreg) save(upreg, file = paste0('metagene',
# metagene_id, 'upreg_dev_pathway.RData'))

# down-regulation, select the lowest quantile
selected <- (column_factor[metagene_id, ] <= cutoffs[2])
downreg <- enrichGO(gene = unique(meta[selected, 5]), OrgDb = "org.Hs.eg.db", ont = "BP",
    readable = TRUE)
dotplot(downreg, font = 9, showCategory = 30) + scale_y_discrete(labels = function(x) wrap_labal(x)) +
    theme(text = element_text(family = "Times New Roman"))
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which will
## replace the existing scale.
```




```
# save(upreg, file = paste0('metagene', metagene_id,
# 'downreg_dev_pathway.RData'))
```

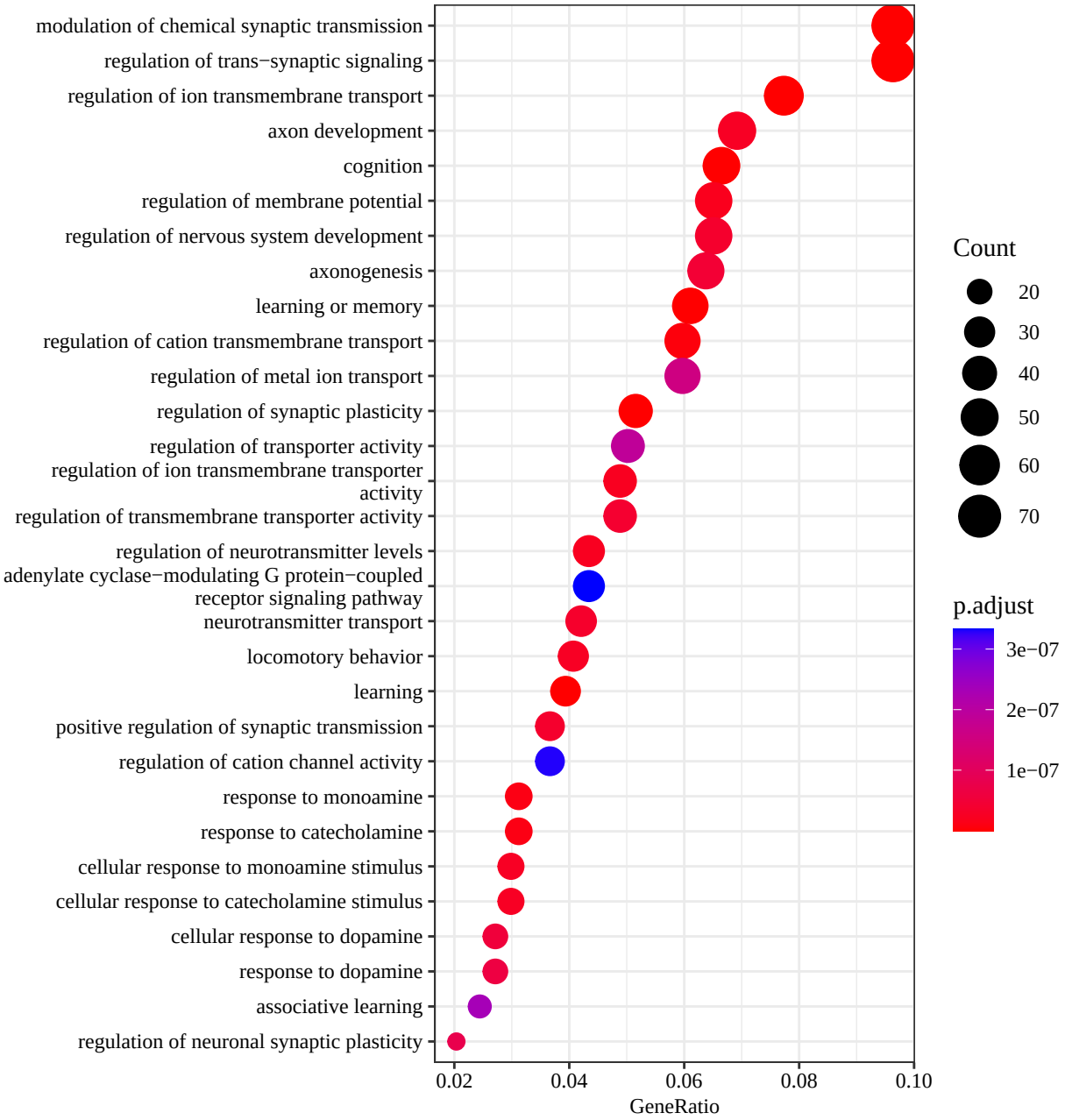
Then, we explore the metagene with the least variance.

```
# metagene_id <- ord[length(ord)]
metagene_id <- 14
cat("Column_id:", metagene_id, "\n")
```

```
## Column_id: 14
```

```
cutoffs <- quantile(column_factor[metagene_id, ], probs = seq(0, 1, 0.025))

# up-regulation, select the highest quantile
selected <- (column_factor[metagene_id, ] <= cutoffs[2])
```

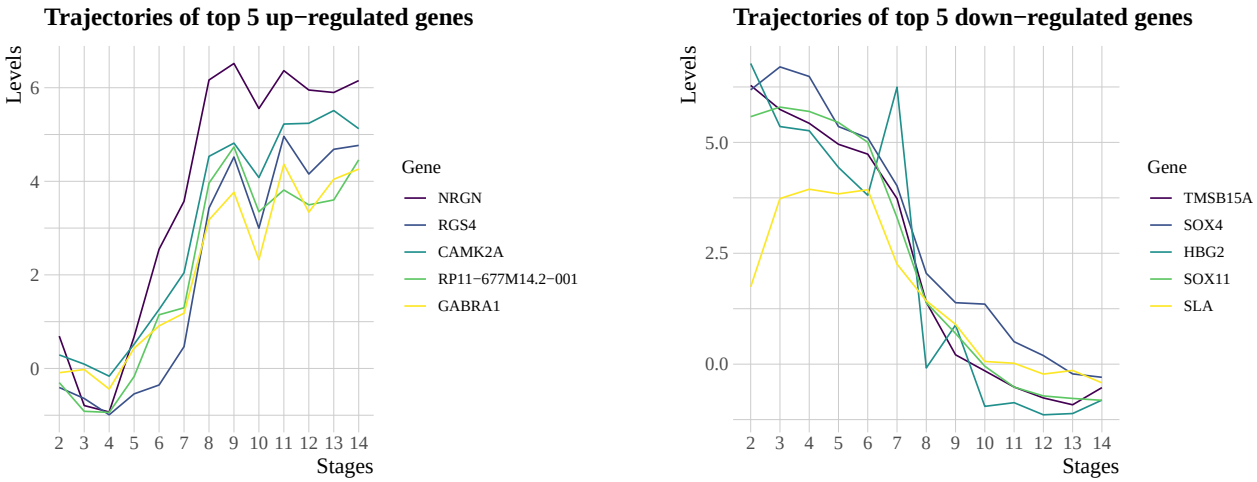
```
upreg <- enrichGO(gene = unique(meta[selected, 5]), OrgDb = "org.Hs.eg.db", ont = "BP",
    readable = TRUE)
dotplot(upreg, font = 9, showCategory = 30, label_format = 50) + theme(text = element_text(family = "Ti
```




```
# save(upreg, file = paste0('metagene', metagene_id,
# 'upreg_dev_pathway.RData'))

# down-regulation, select the lowest quantile selected <-
# (column_factor[metagene_id,] <= cutoffs[2]) downreg <- enrichGO(gene =
# unique(meta[selected,5]), OrgDb = 'org.Hs.eg.db', ont = 'BP', readable =
# TRUE) dot_plot(downreg, font = 8, showCategory=50) save(upreg, file =
# paste0('metagene', metagene_id, 'downreg_dev_pathway.RData'))
```

```r
# DCX expressed in neuronal progenitor cells and immature migrating neurons
# SYP, SYPL1, SYN1 for synapse development MAP1A, MAPT, CAMK2A for dendrite
# development selected_genes <- c('DCX', 'SYP', 'SYPL1', 'SYN1', 'SYPL2',
# 'MAPT', 'MAP1A', 'CAMK2A', 'NRXN1', 'POU2F3', 'SYNPO', 'PRNP', 'RASGRF1')
metagene_id <- ord[1]
gene_order <- order(column_factor[metagene_id, ], decreasing = TRUE)
selected_genes <- gene_id[gene_order[1:5], ]
col_ids <- sapply(selected_genes, function(x) which(meta[[1]] == x))

stage_profiles <- stage_factor %*% column_factor
selected <- stage_profiles[, col_ids]
colnames(selected) <- meta[[4]][col_ids]
rownames(selected) <- r_names
result1 <- melt(selected)
colnames(result1) <- c("Stage", "Gene", "Levels")
result1$Stage <- factor(r_names, levels = r_names)

selected_genes <- gene_id[gene_order[(length(gene_order) - 4):length(gene_order)],
    ]
col_ids <- sapply(selected_genes, function(x) which(meta[[1]] == x))

selected <- stage_profiles[, col_ids]
colnames(selected) <- meta[[4]][col_ids]
rownames(selected) <- r_names
result2 <- melt(selected)
colnames(result2) <- c("Stage", "Gene", "Levels")
result2$Stage <- factor(r_names, levels = r_names)

# par(mfrow = c(1, 2))

p1 <- ggplot(data = result1, aes(x = Stage, y = Levels, group = Gene, color = Gene)) +
    scale_color_viridis(discrete = TRUE) + geom_line(size = 0.5) + ggtitle("Trajectories of top 5 up-reg
    theme_ipsum(base_family = "Times New Roman", base_size = 12, axis_title_size = 14,
        plot_title_size = 16) + scale_x_discrete(labels = 2:14) + xlab("Stages") +
    ylab("Levels")

p2 <- ggplot(data = result2, aes(x = Stage, y = Levels, group = Gene, color = Gene)) +
    scale_color_viridis(discrete = TRUE) + geom_line(size = 0.5) + ggtitle("Trajectories of top 5 down-
    theme_ipsum(base_family = "Times New Roman", base_size = 12, axis_title_size = 14,
        plot_title_size = 16) + scale_x_discrete(labels = 2:14) + xlab("Stages") +
    ylab("Levels")

# p3 <- ggplot(data = result[result$Gene %in% c('MAP1A', 'MAPT', 'CAMK2A'), ],
# aes(x=Stage, y=Levels, group = Gene, color=Gene)) +
# scale_color_viridis(discrete = TRUE) + geom_line(size=1) +
# ggtitle('Trajectories of genes involved in dendrite development') +
# theme_ipsum(base_family = 'Times New Roman', base_size= 12, plot_title_face =
# 'bold', axis_title_size = 16, plot_title_size = 18) +
# scale_x_discrete(labels=2:14)+ xlab('Stages') + ylab('Levels')

# p4 <- ggplot(data = result[result$Gene %in% c('SYNPO', 'PRNP', 'RASGRF1'), ],
# aes(x=Stage, y=Levels, group = Gene, color=Gene)) +
# scale_color_viridis(discrete = TRUE) + geom_line(size=1) +
```

```
# ggtitle('Trajectories of genes enriched for cognition') +
# theme_ipsum(base_family = 'Times New Roman', base_size= 12, plot_title_face =
# 'bold', axis_title_size = 16, plot_title_size = 18) +
# scale_x_discrete(labels=2:14)+ xlab('Stages') + ylab('Levels')
grid.arrange(p1, p2, ncol = 2)
```



Explore pathways that contribute to the brain structure development

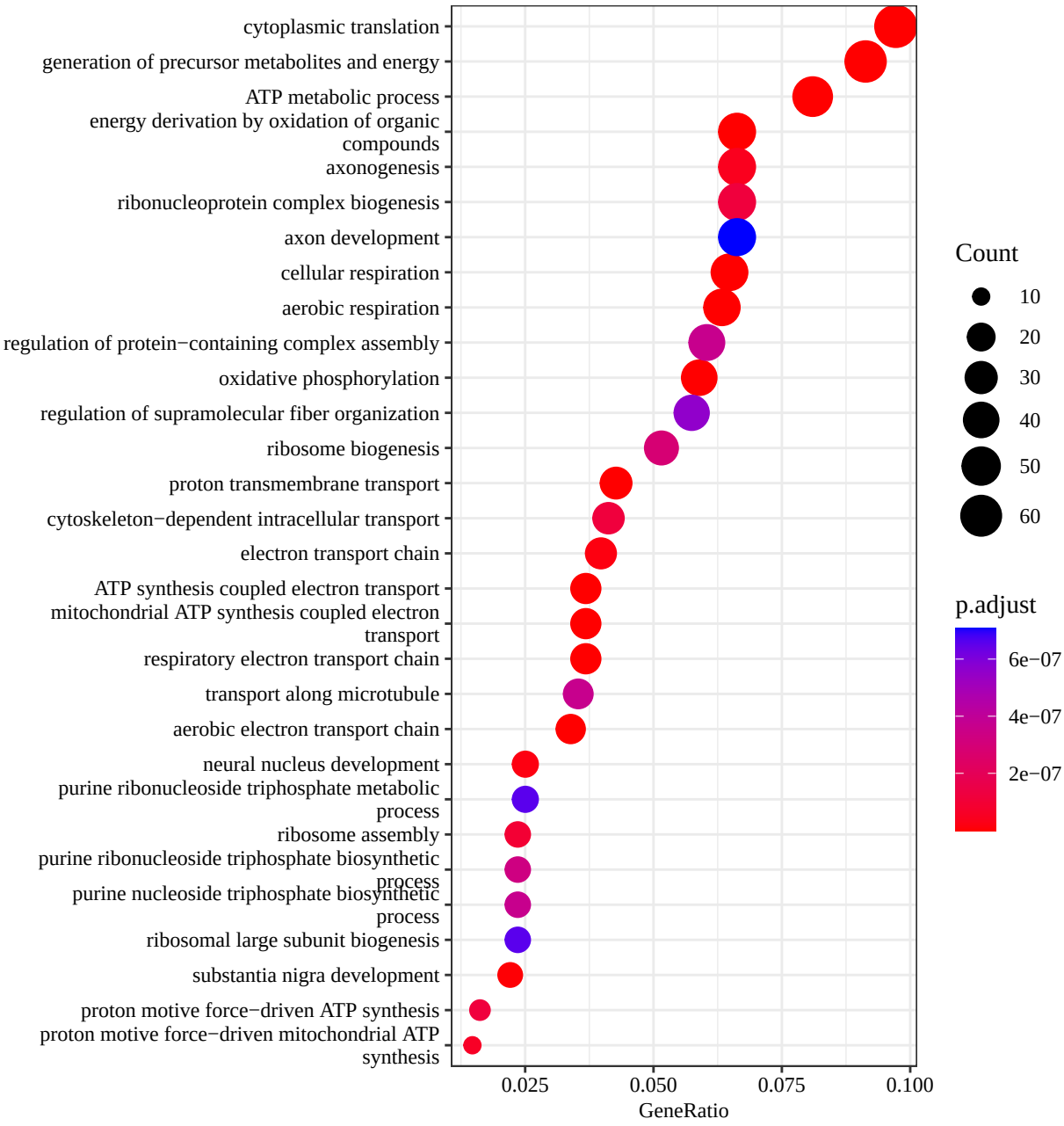
First, we obtain general expression profiles for different tissues, and analyze the functional pathways for each tissue.

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 results for all tissues.

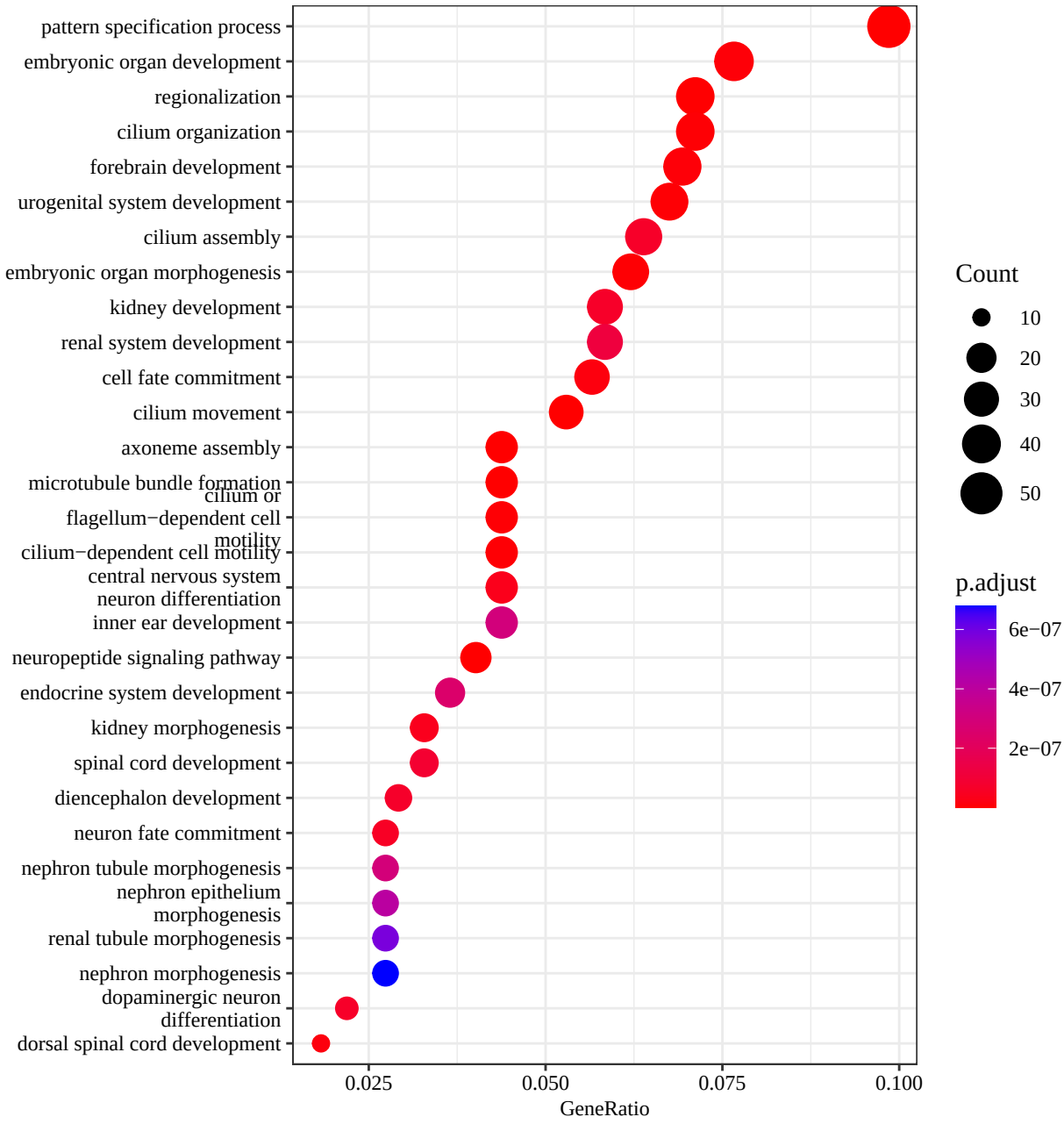
```
# tissue_matrix <- tissue_factor[,-c(14)] %*% column_factor[-c(14),]
tissue_matrix <- tissue_factor %*% column_factor
rownames(tissue_matrix) <- rownames(tissue_factor)

# use the second brain region as example
tissue_id <- 2
cat("Tissue name:", rownames(tissue_factor)[tissue_id], "\n")
```

```
## Tissue name: M1C-S1C
```

```
cutoffs <- quantile(tissue_matrix[tissue_id, ], probs = seq(0, 1, 0.025))

# up_regulation, select the highest quantile
selected <- (tissue_matrix[tissue_id, ] >= cutoffs[length(cutoffs) - 1])
upreg <- enrichGO(gene = unique(meta[selected, 5]), OrgDb = "org.Hs.eg.db", ont = "BP",
    readable = TRUE)
# result <- data.frame(upreg)
dotplot(upreg, font = 9, showCategory = 30, label_format = 50) + theme(text = element_text(family = "Ti
```



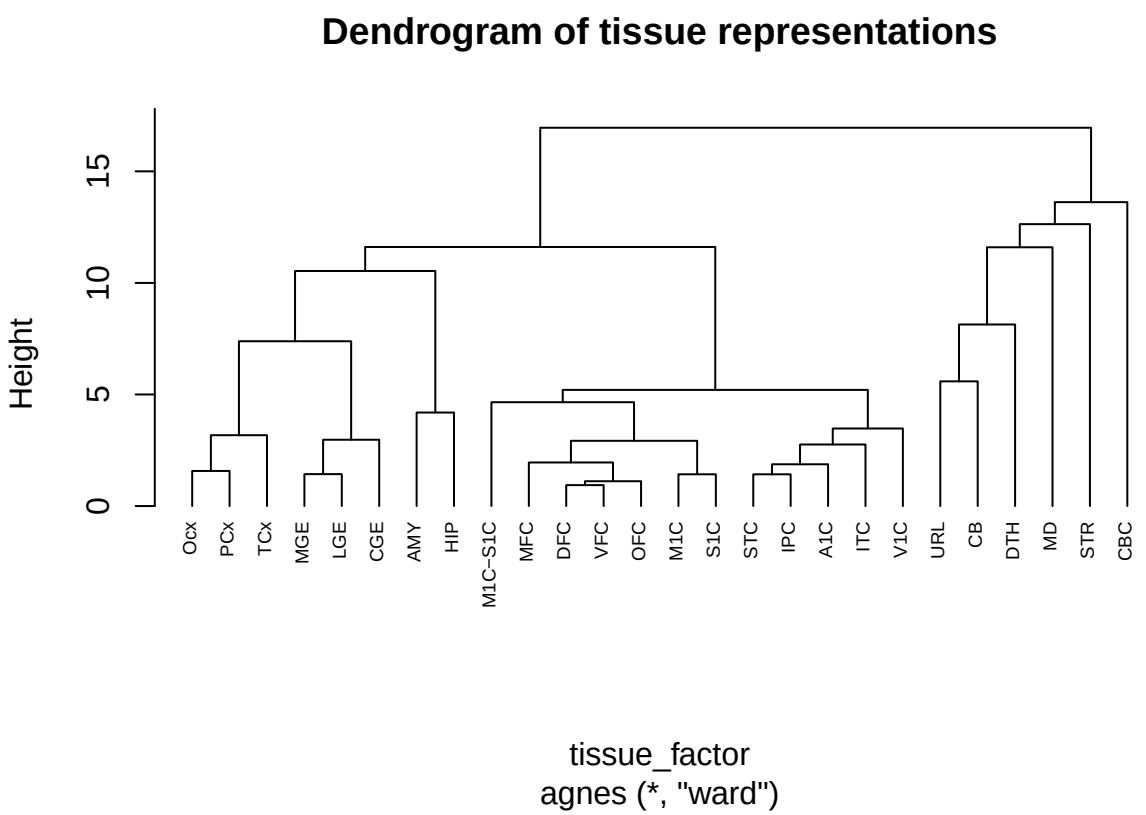

```
# down-regulation, select the lowest quantile
selected <- (tissue_matrix[tissue_id, ] <= cutoffs[2])
downreg <- enrichGO(gene = unique(meta[selected, 5]), OrgDb = "org.Hs.eg.db", ont = "BP",
    readable = TRUE)
dotplot(downreg, font = 9, showCategory = 30) + theme(text = element_text(family = "Times New Roman"))
```




```
m <- c("average", "single", "complete", "ward")
names(m) <- c("average", "single", "complete", "ward")

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

```
##   average    single  complete      ward
## 0.6999680 0.6426831 0.7233920 0.7706310
```

```
hc3 <- agnes(tissue_factor, method = unname(m[which(ac_vec == max(ac_vec))]))
pltree(hc3, cex = 0.6, hang = -1, main = "Dendrogram of tissue representations")
```



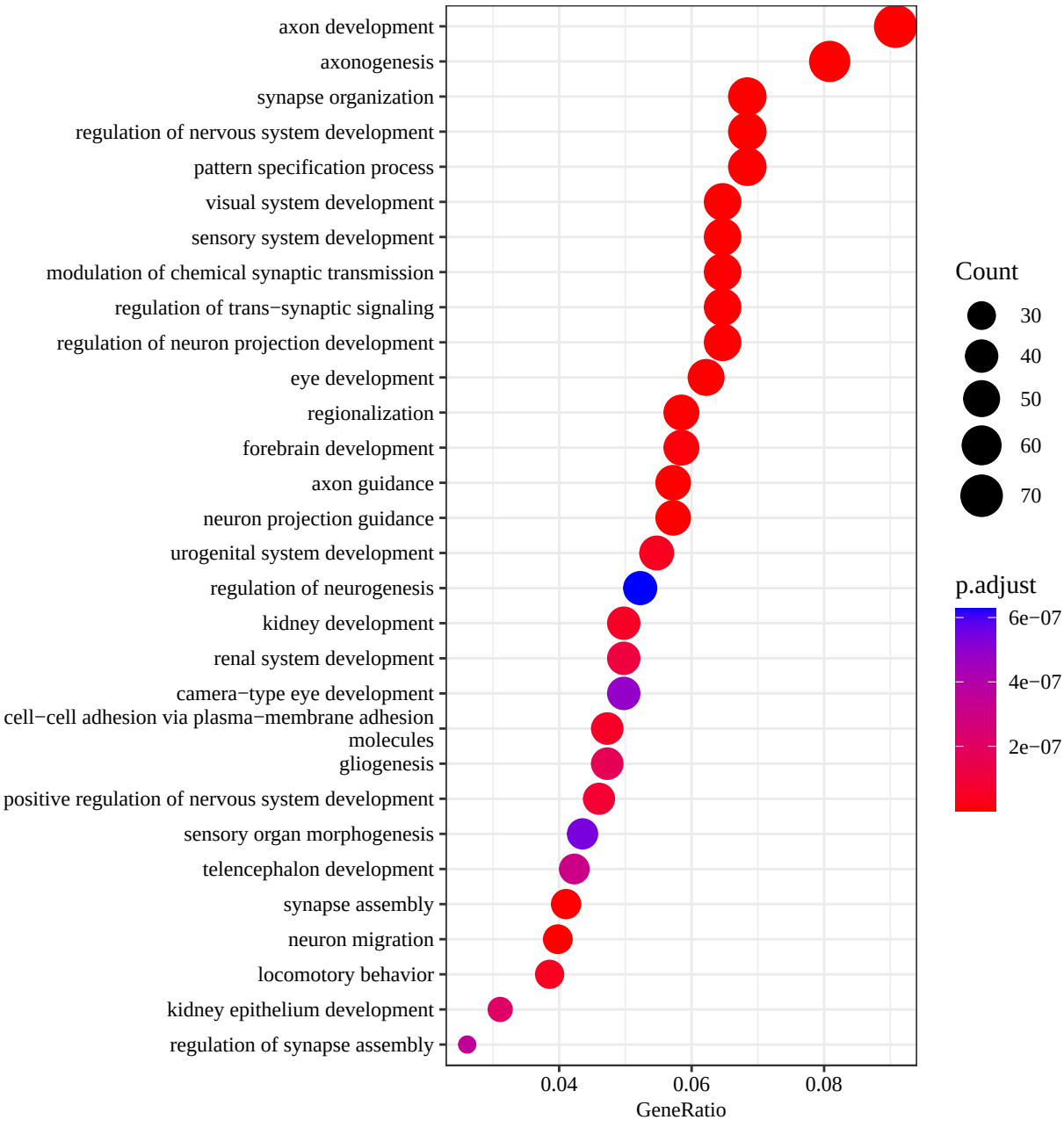
Second, we explore the pathways that contribute the most to the expression difference across different brain structures. This analysis can identify biological processes that partially contribute to the difference across different brain structures.

The second kind of analysis only can partial explain the pathways that signal differently across all tissues in our study, so expanding it with other two or three metagenes with high variance is enough.

```
# tiss_var <- apply(tissue_factor, 2, var)
tiss_var <- apply(tissue_factor, 2, function(x) truncated_var(x))

metagene_id <- which.max(tiss_var)
cat("Column_id:", metagene_id, "\n")
```

```
## Column_id: 7
```

```
cutoffs <- quantile(column_factor[metagene_id, ], probs = seq(0, 1, 0.025))

# up-regulation, select the highest quantile
selected <- (column_factor[metagene_id, ] >= cutoffs[length(cutoffs) - 1])
upreg <- enrichGO(gene = unique(meta[selected, 5]), OrgDb = "org.Hs.eg.db", ont = "BP",
    readable = TRUE)
dotplot(upreg, font = 9, showCategory = 30, label_format = 50) + theme(text = element_text(family = "Ti
```



Explore the interaction between development stages and brain regions

\textcolor{red}{Exploring the interaction is an important feature of our approach, so if possible we may carry out analysis on all possible combinations between brain regions and development stages and select reasonable results for interpretation.

```
# interactions <- glm_interaction(object, c(1, 2))

# unique_cfd <- interactions[[1]] coeff_matrix <- interactions[[2]] pval_matrix
# <- interactions[[3]]

# interaction_indicator <- rep(0, nrow(object[['confounder']])) for(k in
# 1:nrow(unique_cfd)){ selected <- apply(confounder, 1, function(x) all(x ==
```

```
# unique_cfd[k,])) interaction_indicator[selected] <- k }

# intersted_idx <- which(apply(pval_matrix, 1, function(x) sum(x < 1e-8)) ==
# 19)

# confound_values <- (confounder[which(interaction_indicator ==
# intersted_idx[3]), ])

# stage_id <- r_names[confound_values[1]] tissue_id <-
# structure[confound_values[2], 1]

# metagene_id <- which.max(coeff_matrix[intersted_idx[3],])

# # which.min(coeff_matrix[intersted_idx[3],])


# cutoffs <- quantile(column_factor[metagene_id,], probs = seq(0, 1, 0.025))

# # up-regulation, select the highest quantile selected <-
# (column_factor[metagene_id,] >= cutoffs[length(cutoffs) - 1])

# # # down-regulation, select the lowest quantile # selected <-
# (interaction_effect <= cutoffs[2]) upreg <- enrichGO(gene =
# unique(meta[selected,5]), OrgDb = 'org.Hs.eg.db', ont = 'BP', readable =
# TRUE)

# dotplot(upreg, font = 9, showCategory=30) + scale_y_discrete(labels =
# function(x) wrap_labal(x))+ theme(text=element_text(family='Times New
# Roman'))

# # up-regulation, select the highest quantile selected <- (interaction_effect
# <= cutoffs[2])

# # down-regulation, select the lowest quantile # selected <-
# (interaction_effect <= cutoffs[2]) downreg <- enrichGO(gene =
# unique(meta[selected,5]), OrgDb = 'org.Hs.eg.db', ont = 'BP', readable =
# TRUE)

# dotplot(downreg, font = 9, showCategory=30) + scale_y_discrete(labels =
# function(x) wrap_labal(x))+ theme(text=element_text(family='Times New
# Roman'))
```