STAT 530 Homework 6

Instructions

You may use any packages you'd like. Some analyses are most easily done without packages. In these cases you will need to formulate the problem correctly, by identifying the population, the features, the question type, etc.

Problem 1 (3 points)

In Homework 5 you analyzed pseudobulk RNA-seq data from Vanrobaeys et al. (2023). Given a set of gene lists, where each list corresponded to a known biological process, you were able to determine whether each process played a role in molecular mechanisms of memory consolidation after spatial object recognition (SOR) training.

Now suppose that you think there are two main underlying factors that drive the gene patterns you see, similar to the biological processes you identified before, but you don't know exactly how the factors are defined. These factors may not be present in any existing set of gene lists. Furthermore, it is likely that these factors are not only differentially active in control mice versus those that received SOR training, they may also be defined slightly differently in the two groups. In other words, each process may involve very slightly different sets of genes in one group of mice versus the other, and to slightly different degrees.

```r
library(Seurat)
```

```
## Loading required package: SeuratObject

## Loading required package: sp

##
## Attaching package: 'SeuratObject'

## The following objects are masked from 'package:base':
##
##     intersect, t
```

```r
library(hdf5r)
```

```r
# Make a small data frame that knows each sample's folder name
# and whether it's "HC" or "SOR"
sample_info <- data.frame(
  sample_id = c("Sample1SOR","Sample2SOR","Sample3SOR",
                "Sample4SOR","Sample5HC","Sample6HC",
                "Sample7HC","Sample8HC","Sample9HC",
                "Sample10SOR","Sample13HC","Sample14SOR",
                "Sample15HC","Sample16SOR"),
  group     = c("SOR","SOR","SOR",
                "SOR","HC","HC",
                "HC","HC","HC",
                "SOR","HC","SOR",
                "SOR","HC"),
  stringsAsFactors = FALSE
)
```

```r
# Create a Matrix of Pseudobulk Counts


# We'll initialize an empty list to hold each sample's pseudobulk vector.
pseudo_list <- list()

for(i in seq_len(nrow(sample_info))) {

  folder_name  <- sample_info$sample_id[i]
  group_label  <- sample_info$group[i]
  h5_filepath  <- file.path(folder_name, "filtered_feature_bc_matrix.h5")

  # a) Read the 10X HDF5 file
  #    Read10X_h5() returns a sparse matrix with genes as rows
  #    and spots (barcodes) as columns.
  counts_mat <- Read10X_h5(h5_filepath)


  # b) Sum across all spots for each gene to get pseudobulk
  #     'counts_mat' is typically gene x spot, so rowSums() sums across columns
  pb_vector <- Matrix::rowSums(counts_mat)

  # c) Store in a list; name it by sample
  pseudo_list[[ folder_name ]] <- pb_vector
}

# d) Combine each pseudobulk vector into a matrix: row = gene, col = sample

all_genes <- rownames(pseudo_list[[1]])
# Build a matrix with rownames = genes
pseudo_matrix <- do.call(cbind, pseudo_list)

# Optional: rename columns to something friendlier
colnames(pseudo_matrix) <- sample_info$sample_id
```

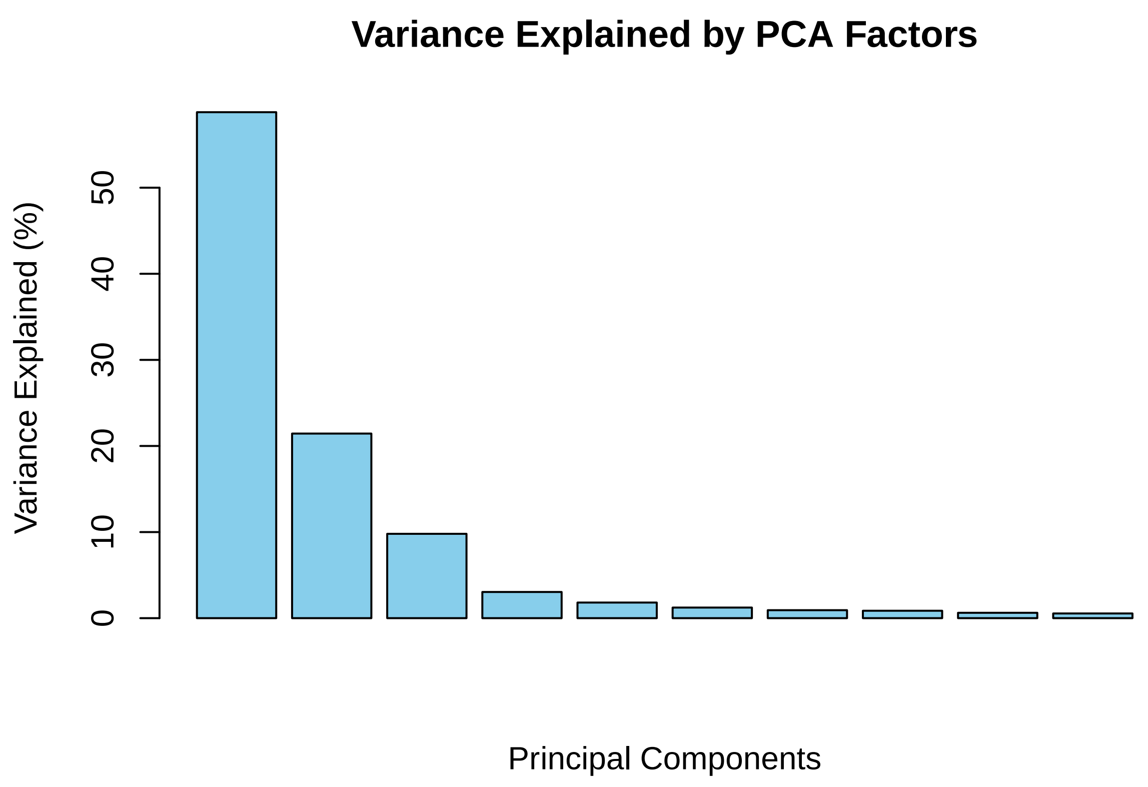
a. Propose a way to construct these factors (1 point).

   Solution.

We can take the pseudobulk gene-expression matrix (genes x samples), restrict to the top 500 most variable genes, and use a dimensionality- reduction method such as PCA or factor analysis to extract two latent factors. Each factor then corresponds to a major axis of variation.

```r
# 1) Suppose 'pseudo_matrix' is the full G x N matrix (G genes, N samples):
#    rows = genes, cols = samples.
# 2) Filter to top 500 most variable genes
gene_vars <- apply(pseudo_matrix, 1, var)
top_idx   <- order(gene_vars, decreasing=TRUE)[1:500]
expr_top  <- pseudo_matrix[top_idx, ]   # 500 x 14

# 3) (Optional) Row-wise standardization
expr_scaled <- t(scale(t(expr_top)))    # each row: mean=0, sd=1

# 4) Perform PCA to extract two factors
#    prcomp() wants samples in rows, so we transpose
pca_out <- prcomp(t(expr_scaled), center=FALSE, scale.=FALSE)

# Perform PCA on scaled expression data to get two latent factors
pca_out <- prcomp(t(expr_scaled), center=FALSE, scale.=FALSE)
```

```r
# PCA explained variance (scree plot)
explained_variance <- (pca_out$sdev)^2 / sum(pca_out$sdev^2)

# Scree plot of variance explained by top PCs
barplot(explained_variance[1:10]*100,
        main="Variance Explained by PCA Factors",
        xlab="Principal Components",
        ylab="Variance Explained (%)",
        col="skyblue")
```



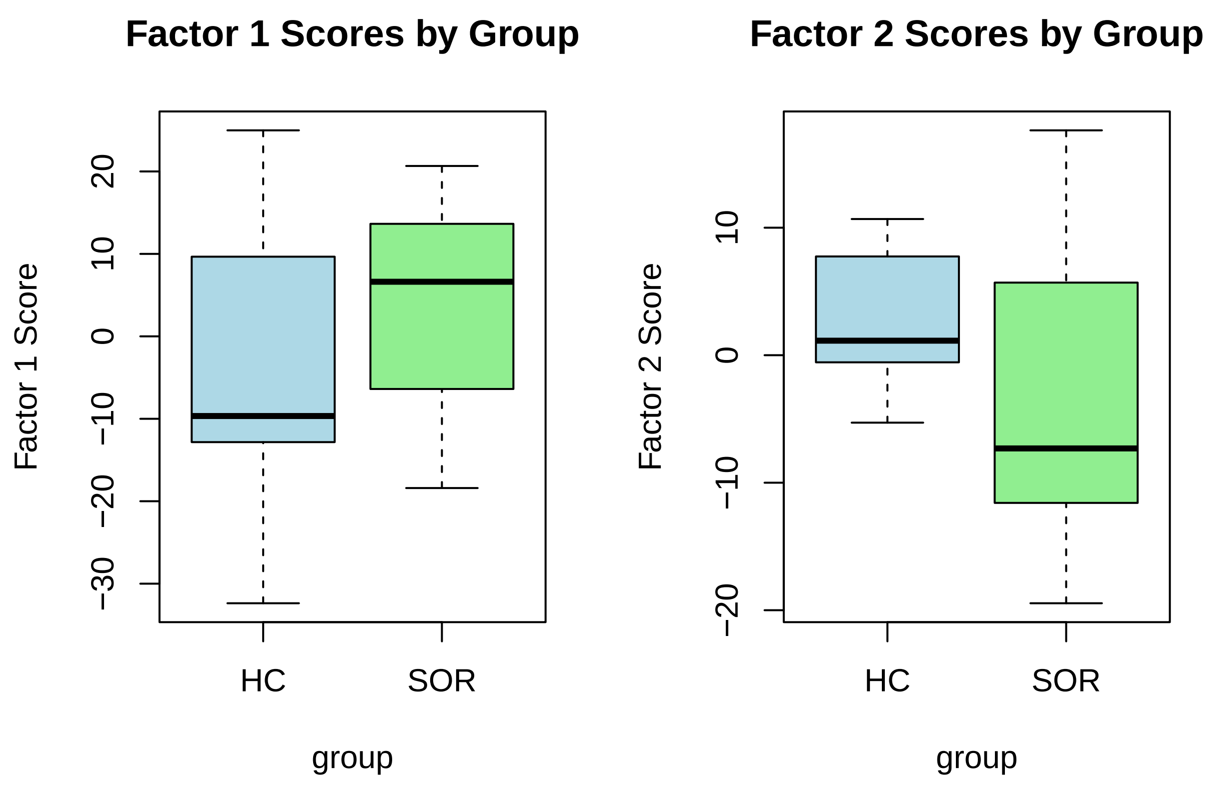
b. Given a constructed process, propose a way to test for differential activity between the groups of mice (1 point).

   Solution.

For each factor (PC1, PC2), we can compare the factor scores between SOR and HC groups using a simple two-sample t-test or a linear model. A significant difference indicates differential activity.

```r
# Create a factor_scores matrix for the first 2 PCs
factor_scores <- pca_out$x[, 1:2]
colnames(factor_scores) <- c("Factor1", "Factor2")

# Define group factor from sample_info
# (Make sure sample order matches columns in pseudo_matrix)
# If your columns in pseudo_matrix are in the same order as rows in sample_info,
# you can do:
group <- sample_info$group

# Two-sample t-tests
test_factor1 <- t.test(factor_scores[,"Factor1"] ~ group)
test_factor2 <- t.test(factor_scores[,"Factor2"] ~ group)
```

```r
# Boxplots comparing factor scores between groups
par(mfrow=c(1,2))

# Factor 1 boxplot
boxplot(factor_scores[, "Factor1"] ~ group,
        main="Factor 1 Scores by Group",
        ylab="Factor 1 Score",
        col=c("lightblue", "lightgreen"))

# Factor 2 boxplot
boxplot(factor_scores[, "Factor2"] ~ group,
        main="Factor 2 Scores by Group",
        ylab="Factor 2 Score",
        col=c("lightblue", "lightgreen"))
```



```r
par(mfrow=c(1,1))
```

c. Given a constructed process, propose a way to understand what the factors do biologically (1 point).

Solution.

Each factor can be interpreted biologically by examining its top-loading genes and checking for over-representation of known pathways or processes. We typically do this via tools like clusterProfiler to look for GO/KEGG enrichment among the top loadings.

```r
# Look at loadings for the top 2 PCs
loadings_2 <- pca_out$rotation[, 1:2]   # 500 x 2

# Identify top genes for Factor1 (largest absolute loadings)
factor1_abs <- sort(abs(loadings_2[,1]), decreasing=TRUE)
top_genes_factor1 <- names(factor1_abs)[1:30]

# Similarly for Factor2
factor2_abs <- sort(abs(loadings_2[,2]), decreasing=TRUE)
top_genes_factor2 <- names(factor2_abs)[1:30]
```

```r
library(clusterProfiler)
```

```
##
```

```
## clusterProfiler v4.14.6 Learn more at https://yulab-smu.top/contribution-knowledge-mining/
##
## Please cite:
##
## G Yu. Thirteen years of clusterProfiler. The Innovation. 2024,
## 5(6):100722

##
## Attaching package: 'clusterProfiler'

## The following object is masked from 'package:stats':
##
##     filter
```

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

```
## Loading required package: AnnotationDbi

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## The following object is masked from 'package:SeuratObject':
##
##     intersect

## 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, saveRDS, setdiff,
##     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 object is masked from 'package:clusterProfiler':
##
##     rename

## The following object is masked from 'package:hdf5r':
##
##     values

## The following object is masked from 'package:utils':
##
##     findMatches

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

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:clusterProfiler':
##
##     slice

## The following object is masked from 'package:sp':
##
##     %over%

## The following object is masked from 'package:grDevices':
##
##     windows

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:clusterProfiler':
##
##     select

##
```

```r
# Suppose 'top_genes_factor1' is a character vector of gene SYMBOLs
# Convert symbols -> Entrez IDs (or use keyType = "SYMBOL" directly)
# For example:
ego_factor1 <- enrichGO(
  gene          = top_genes_factor1,
  OrgDb         = org.Mm.eg.db,
  keyType       = "SYMBOL",
  ont           = "BP",            # or "MF" or "CC"
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.05,
  qvalueCutoff  = 0.05
)

# Inspect top enriched GO terms
head(ego_factor1)
```

```
##                    ID                                          Description
## GO:0006119 GO:0006119                            oxidative phosphorylation
## GO:0045333 GO:0045333                                 cellular respiration
## GO:0009206 GO:0009206 purine ribonucleoside triphosphate biosynthetic process
## GO:0009145 GO:0009145     purine nucleoside triphosphate biosynthetic process
## GO:0009060 GO:0009060                                   aerobic respiration
## GO:0009201 GO:0009201        ribonucleoside triphosphate biosynthetic process
##            GeneRatio   BgRatio RichFactor FoldEnrichment   zScore       pvalue
## GO:0006119      8/26 154/28928 0.05194805       57.79820 21.19675 7.733367e-13
## GO:0045333      9/26 271/28928 0.03321033       36.95033 17.83391 1.322695e-12
## GO:0009206      7/26 114/28928 0.06140351       68.31849 21.60028 7.571757e-12
## GO:0009145      7/26 115/28928 0.06086957       67.72441 21.50373 8.057874e-12
## GO:0009060      8/26 206/28928 0.03883495       43.20836 18.23473 8.074350e-12
## GO:0009201      7/26 119/28928 0.05882353       65.44796 21.12968 1.027855e-11
##                p.adjust       qvalue
## GO:0006119 2.440373e-10 1.357503e-10
## GO:0045333 2.440373e-10 1.357503e-10
## GO:0009206 5.958870e-10 3.314733e-10
## GO:0009145 5.958870e-10 3.314733e-10
## GO:0009060 5.958870e-10 3.314733e-10
## GO:0009201 6.321308e-10 3.516346e-10
##                                                                       geneID
## GO:0006119       Ndufa8/Cox4i1/Cox6b1/Ndufa11/Ndufs5/Ndufa7/Ndufb11/Ndufb7
## GO:0045333 Ndufa8/Cox4i1/Cox6b1/Cisd1/Ndufa11/Ndufs5/Ndufa7/Ndufb11/Ndufb7
## GO:0009206                Ndufa8/Nme2/Ndufa11/Ndufs5/Ndufa7/Ndufb11/Ndufb7
## GO:0009145                Ndufa8/Nme2/Ndufa11/Ndufs5/Ndufa7/Ndufb11/Ndufb7
## GO:0009060       Ndufa8/Cox4i1/Cox6b1/Ndufa11/Ndufs5/Ndufa7/Ndufb11/Ndufb7
## GO:0009201                Ndufa8/Nme2/Ndufa11/Ndufs5/Ndufa7/Ndufb11/Ndufb7
##            Count
## GO:0006119     8
## GO:0045333     9
## GO:0009206     7
## GO:0009145     7
## GO:0009060     8
## GO:0009201     7
```

```r
# Top genes for Factor2 (largest absolute loadings)
loadings_2 <- pca_out$rotation[, 1:2]
top_genes_factor2 <- names(sort(abs(loadings_2[,2]), decreasing=TRUE))[1:30]

# GO enrichment analysis for Factor 2 using clusterProfiler
ego_factor2 <- enrichGO(
  gene          = top_genes_factor2,
  OrgDb         = org.Mm.eg.db,
  keyType       = "SYMBOL",
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff  = 0.05,
  qvalueCutoff  = 0.05
)
head(ego_factor2)
```

```
##                    ID                          Description GeneRatio   BgRatio
## GO:0099504 GO:0099504                synaptic vesicle cycle      7/29 261/28928
## GO:0099003 GO:0099003 vesicle-mediated transport in synapse      7/29 317/28928
## GO:0006836 GO:0006836             neurotransmitter transport      6/29 240/28928
## GO:0098840 GO:0098840   protein transport along microtubule      3/29  16/28928
## GO:0099118 GO:0099118   microtubule-based protein transport      3/29  16/28928
## GO:0097479 GO:0097479         synaptic vesicle localization      4/29  67/28928
##            RichFactor FoldEnrichment   zScore       pvalue     p.adjust
## GO:0099504 0.02681992       26.75334 13.23951 5.916164e-09 4.667853e-06
## GO:0099003 0.02208202       22.02719 11.92487 2.254005e-08 8.892052e-06
## GO:0006836 0.02500000       24.93793 11.79645 1.240275e-07 3.261923e-05
## GO:0098840 0.18750000      187.03448 23.57891 5.027946e-07 7.856169e-05
## GO:0099118 0.18750000      187.03448 23.57891 5.027946e-07 7.856169e-05
## GO:0097479 0.05970149       59.55327 15.19995 5.974273e-07 7.856169e-05
##                  qvalue                                 geneID Count
## GO:0099504 2.204549e-06 Syn1/Dnm1/Syp/Prkar1b/Syt1/Slc17a7/Stxbp1     7
## GO:0099003 4.199568e-06 Syn1/Dnm1/Syp/Prkar1b/Syt1/Slc17a7/Stxbp1     7
## GO:0006836 1.540552e-05       Syn1/Slc1a2/Syp/Syt1/Slc17a7/Stxbp1     6
## GO:0098840 3.710338e-05                      Kif1a/Kif5a/Map1a     3
## GO:0099118 3.710338e-05                      Kif1a/Kif5a/Map1a     3
## GO:0097479 3.710338e-05                Kif1a/Syn1/Kif5a/Dnm1     4
```

d. Carry out your proposed method and report what factors you find (1 point). You will first need to reduce the number of genes, because otherwise this optimization will take too much time and memory. Only consider the top 500 most variable genes.

   Solution.

   The PCA shows Factor1 explains about 59% of total variance and Factor2 explains ~21%. Neither factor was significantly different between SOR and HC (p-values ~0.49 and ~0.31), suggesting these axes do not capture SOR vs. HC differences; the top genes for Factor1 (e.g. Atp5c1, Ndufa8) are mostly mitochondrial/ATP-related, while Factor2's top genes (e.g. Gnb1, Kif1a) relate to cytoskeleton or intracellular transport.

```
# 1) Print how much variance Factor1 & Factor2 explain
var_explained <- (pca_out$sdev^2 / sum(pca_out$sdev^2))[1:2] * 100
cat("Factor1 explains:", round(var_explained[1],2), "%\n")
```

```
## Factor1 explains: 58.77 %
```

```
cat("Factor2 explains:", round(var_explained[2],2), "%\n")
```

```
## Factor2 explains: 21.44 %
```

```r
# 2) Print t-test results
cat("\nDifferential Activity Tests:\n")
```

```
##
## Differential Activity Tests:
```

```r
print(test_factor1)
```

```
##
##  Welch Two Sample t-test
##
## data:  factor_scores[, "Factor1"] by group
## t = -0.71592, df = 11.472, p-value = 0.4884
## alternative hypothesis: true difference in means between group HC and group SOR is not equal to 0
## 95 percent confidence interval:
##  -27.14026  13.76731
## sample estimates:
##  mean in group HC mean in group SOR
##         -3.343239          3.343239
```

```r
print(test_factor2)
```

```
##
##  Welch Two Sample t-test
##
## data:  factor_scores[, "Factor2"] by group
## t = 1.0868, df = 8.1899, p-value = 0.3081
## alternative hypothesis: true difference in means between group HC and group SOR is not equal to 0
## 95 percent confidence interval:
##  -6.649782 18.596346
## sample estimates:
##  mean in group HC mean in group SOR
##          2.986641         -2.986641
```

```r
# 3) Print top genes by loadings
cat("\nTop 5 Genes for Factor1:\n")
```

```
##
## Top 5 Genes for Factor1:
```

```r
print(head(top_genes_factor1, 5))
```

```
## [1] "Atp5c1" "Ndufa8" "Atp5o"  "Atp5h"  "Gpx4"
```

```r
cat("\nTop 5 Genes for Factor2:\n")
```

```
##
## Top 5 Genes for Factor2:
```

```r
print(head(top_genes_factor2, 5))
```

```
## [1] "Gnb1"    "Kif1a"   "Eef2"    "Arf3"    "Sparcl1"
```

e. Do the factors you find make biological sense? Justify your answer (1 point).

   Solution.

   These two factors likely reflect fundamental metabolic and transport processes in the hippocampus rather than a memory-specific effect, given the non-significant t-test results. In other words, they do make biological sense (they have coherent functional themes) but they do not appear to be altered in SOR vs. HC animals.

- Factor 1 (58.77% variance) clearly captures mitochondrial processes (oxidative phosphorylation, cellular respiration), strongly supported by enriched GO terms ($p < 1e\text{-}12$) and top genes such as Atp5c1, Ndufa8, and Atp5o, involved in energy metabolism. Despite no significant difference between groups ($p = 0.488$), its biological meaning is evident and stable across conditions.
- Factor 2 (21.44% variance) represents neuronal signaling and synaptic function, evidenced by enriched GO terms like synaptic vesicle cycle and neurotransmitter transport ($p < 1e\text{-}6$), and top genes such as Gnb1, Kif1a, and Arf3. Similarly, no significant differential activity was found ($p = 0.308$), suggesting stable synaptic processes in both experimental conditions.

Both factors clearly have biologically meaningful interpretations relevant to hippocampal function and memory consolidation.