

# Homework 2

Almog Angel

- Load packages and set working directory

```
library(DESeq2)

## Warning: package 'DESeq2' was built under R version 4.3.3

## Loading required package: S4Vectors

## Loading required package: stats4

## Loading required package: BiocGenerics

##
## Attaching package: 'BiocGenerics'

## 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, setdiff, sort,
##      table, tapply, union, unique, unsplit, which.max, which.min

##
## Attaching package: 'S4Vectors'

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

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

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

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

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb
```

```
## Loading required package: SummarizedExperiment
```

```
## Loading required package: MatrixGenerics
```

```
## Loading required package: matrixStats
```

```
##  
## Attaching package: 'MatrixGenerics'
```

```
## The following objects are masked from 'package:matrixStats':  
##  
##   colAlls, colAnyNAs, colAnys, colAvgPerRowSet, colCollapse,  
##   colCounts, colCummaxs, colCummins, colCumprods, colCumsums,  
##   colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,  
##   colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,  
##   colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,  
##   colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,  
##   colWeightedMeans, colWeightedMedians, colWeightedSds,  
##   colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgPerColSet,  
##   rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,  
##   rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,  
##   rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,  
##   rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,  
##   rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,  
##   rowWeightedMads, rowWeightedMeans, rowWeightedMedians,  
##   rowWeightedSds, rowWeightedVars
```

```
## 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")'.
```

```
##  
## Attaching package: 'Biobase'
```

```
## The following object is masked from 'package:MatrixGenerics':  
##  
##   rowMedians
```

```
## The following objects are masked from 'package:matrixStats':  
##  
##   anyMissing, rowMedians
```

```
library(ashr)  
library(EnhancedVolcano)
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
## Loading required package: ggrepel
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(clusterProfiler)
```

```
## Warning: package 'clusterProfiler' was built under R version 4.3.3
```

```
##
```

```
## clusterProfiler v4.10.1 For help: https://yulab-smu.top/biomedical-knowledge-mining-book/
##
## If you use clusterProfiler in published research, please cite:
## T Wu, E Hu, S Xu, M Chen, P Guo, Z Dai, T Feng, L Zhou, W Tang, L Zhan, X Fu, S Liu, X Bo, and G Yu. clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. The Innovation. 2021, 2(3):100141
```

```
##
## Attaching package: 'clusterProfiler'
```

```
## The following object is masked from 'package:IRanges':
##
## slice
```

```
## The following object is masked from 'package:S4Vectors':
##
## rename
```

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

```
library(msigdb)
library(tidyverse)
```

```
## — Attaching core tidyverse packages ————— tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ lubridate  1.9.3      ✓ tibble     3.2.1
## ✓ purrr      1.0.2      ✓ tidyr      1.3.1
```

```
## — Conflicts ————— tidyverse_conflicts() —
## ✗ lubridate::%within%() masks IRanges::%within%()
## ✗ dplyr::collapse()      masks IRanges::collapse()
## ✗ dplyr::combine()       masks Biobase::combine(), BiocGenerics::combine()
## ✗ dplyr::count()         masks matrixStats::count()
## ✗ dplyr::desc()          masks IRanges::desc()
## ✗ tidyr::expand()        masks S4Vectors::expand()
## ✗ dplyr::filter()        masks clusterProfiler::filter(), stats::filter()
## ✗ dplyr::first()         masks S4Vectors::first()
## ✗ dplyr::lag()           masks stats::lag()
## ✗ ggplot2::Position()    masks BiocGenerics::Position(), base::Position()
## ✗ purrr::reduce()        masks GenomicRanges::reduce(), IRanges::reduce()
## ✗ dplyr::rename()        masks clusterProfiler::rename(), S4Vectors::rename()
## ✗ lubridate::second()    masks S4Vectors::second()
## ✗ lubridate::second<-()  masks S4Vectors::second<-()
## ✗ purrr::simplify()      masks clusterProfiler::simplify()
## ✗ dplyr::slice()         masks clusterProfiler::slice(), IRanges::slice()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(GEOquery)
```

```
## Setting options('download.file.method.GEOquery'='auto')  
## Setting options('GEOquery.inmemory.gpl'=FALSE)
```

```
setwd("C:/technion/Bio/hw2")
```

- In this homework assignment, you will perform differential gene expression analysis
  - Read the abstract of the study “Temporal changes in postprandial blood transcriptomes reveal subject-specific pattern of expression of innate immunity genes after a high-fat meal” by Danielle G Lemay et al. (pdf file attached)
1. Explain in your own words the term “whole blood transcriptome” (up to 30 words)
  2. Explain in your own words the objective and design of the study (up to 250 words)

Write your answer here:

(1)

"Whole blood transcriptome" refers to the genome-wide expression of all genes, including different kinds of RNA (rRNA, mRNA and etc) from cells contained in the blood.

"Whole blood" is comprised of red blood cells, white blood cells and platelets.

(2)

The study objective aims to determine the changes in whole blood transcriptome – what changes in the gene expression caused by a high-fat meal. Mainly what are the effects related to white blood cells and innate immunity system.

The study design is composed by 2 groups.

The first, 5 subjects with dietary intervention. Each subject was fed a test breakfast, on 3 test days with 4 weeks washout between each test. The breakfast included high fat meal supplemented with placebo, blueberry powder and DHA (anti-inflammatory supplements). The blood tests from these subjects were taken in different time periods, 0 (before the meal – fasting), 3 and 6 hours after the meal. A total of 45 samples. The supplement's purpose is to determine whether consumption of rich anti-inflammatory polyphenols suppresses the high-fat-diet induced postprandial inflammation.

The second group included an addition of 15 subjects selected for the reverse-transcription polymerase chain reaction (RT-PCR) study. A total of 180 samples. The second group's purpose is a control group. The samples from the 20 other subjects is used to determine if the expression of innate genes in the postprandial period after a high fat meal is subject specific.

- Take a look at the manuscript, find the Gene Expression Omnibus (GEO) accession number
- Write it down here:

GEO accession number:  
GSE127530

- Go in to the GEO website and search for this accession number.
- Download the counts data (the second one: “fixed”)
- Download the Series Matrix File(s)
- Read the count matrix and metadata into the corresponding variables below

```
counts <- read.table("GSE127530_fixed_combinedCounts.txt")  
metadata <- getGEO(filename="./GSE127530_series_matrix.txt")
```

- Convert metadata to a tibble
- Select and rename the following columns and store it back into “metadata”: Sample = description Subject = study subject id #:ch1 Day = test day:ch1 Time = time of blood draw:ch1
- Make sure the “Time” column is in factor class

```

metadata <- as_tibble(metadata) %>%
  dplyr::select(Sample = description, Subject = study.subject.id...ch1,
               Day = test.day.ch1, Time = time.of.blood.draw.ch1) %>%
  mutate(Time = ifelse(Time == "3 hr postprandial", "3.hr.postprandial", Time)) %>%
  mutate(Time = ifelse(Time == "6 hr postprandial", "6.hr.postprandial", Time))
metadata$Time <- as.factor(metadata$Time)
metadata

```

```

## # A tibble: 45 × 4
##   Sample      Subject Day   Time
##   <chr>      <chr>  <chr> <fct>
## 1 S2-D1-Fast_S62 2      1    fasting
## 2 S2-D1-3hr_S63 2      1    3.hr.postprandial
## 3 S2-D1-6hr_S64 2      1    6.hr.postprandial
## 4 S2-D2-Fast_S65 2      2    fasting
## 5 S2-D2-3hr_S66 2      2    3.hr.postprandial
## 6 S2-D2-6hr_S67 2      2    6.hr.postprandial
## 7 S2-D3-Fast_S68 2      3    fasting
## 8 S2-D3-3hr_S69 2      3    3.hr.postprandial
## 9 S2-D3-6hr_S70 2      3    6.hr.postprandial
## 10 S9-D1-Fast_S71 9      1    fasting
## # i 35 more rows

```

- Make the reference level for Time be the “Fasting” condition

```

metadata$Time <- relevel(metadata$Time, ref="fasting")

```

- Remove genes with zero total counts

```

counts <- counts[rowSums(counts[])>0,]

```

- Make a DESeq2 object called “dds” and use the Time column for the design
- Don’t forget to check that samples in colData and countData match
- In this example tidy = FALSE (use ?DESeqDataSetFromMatrix to learn why)

```

# adjust the Sample column to the counts column names
metadata$Sample <- gsub("-", ".", metadata$Sample)

# making them in the same order
counts <- counts[, metadata$Sample]
all(metadata$Sample == colnames(counts))

```

```

## [1] TRUE

```

```

dds <- DESeqDataSetFromMatrix(countData=counts,
                              colData=metadata,
                              design= ~Time,
                              tidy=FALSE)

```

- Run DESeq2

```

dds <- DESeq(dds)

```

```

## estimating size factors

```

```

## estimating dispersions

```

```

## gene-wise dispersion estimates

```

```

## mean-dispersion relationship

```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
## -- replacing outliers and refitting for 12 genes  
## -- DESeq argument 'minReplicatesForReplace' = 7  
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

- We are now comparing the groups of fasting and time = 3 hours
- Use the function lfcShrink() to adjust the LFC values with the following arguments:
  1. coef="Time\_3.hr.postprandial\_vs\_fasting" for the name of the coefficient to shrink
  2. type="apeglm"
- Store the results back into "res"

```
res <- lfcShrink(dds, coef="Time_3.hr.postprandial_vs_fasting", type="apeglm")
```

```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:  
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for  
##     sequence count data: removing the noise and preserving large differences.  
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
## Warning in nbinomGLM(x = x, Y = YNZ, size = size, weights = weightsNZ, offset =  
## offsetNZ, : the line search routine failed, unable to sufficiently decrease the  
## function value
```

- Sort "res" by the adjusted p-values from lowest to highest and store in "resOrdered"

```
resOrdered <- res[order(res$padj),]  
length(which(resOrdered$padj < 0.05))
```

```
## [1] 85
```

- Did you find any significant genes (adjusted p-value < 0.05)? If so, how many?

```
Write your answer here:  
85
```

- Use the function vst() to extract the normalized counts from dds into "counts.vst"

```
# ?vst()  
counts.vst <- vst(dds)
```

- Use the function plotPCA() to generate a ggplot2 object for the PCA visualization and store it into "pcaData":
  1. Use the top 1000 variable genes
  2. Use intgroup=c("Subject", "Day") for grouping
  3. Use returnData = TRUE

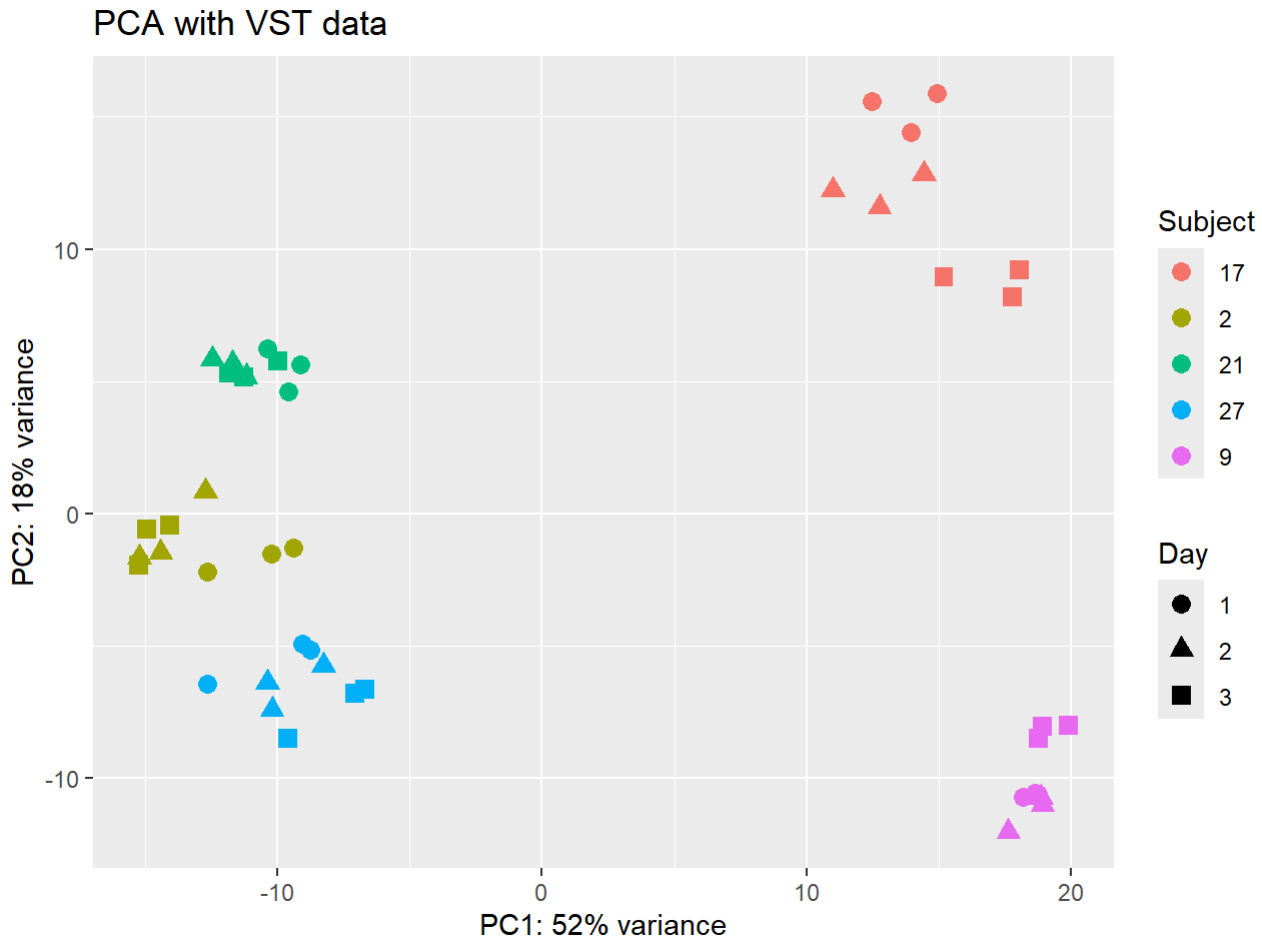
```
pcaData <- plotPCA(counts.vst, ntop=1000, c("Subject", "Day"), returnData = TRUE)
```

```
## using ntop=1000 top features by variance
```

- Use ggplot to plot the object you made above:
  1. Plot the percent of variance explained in PC1 and PC2 labels
  2. Color points by Subject

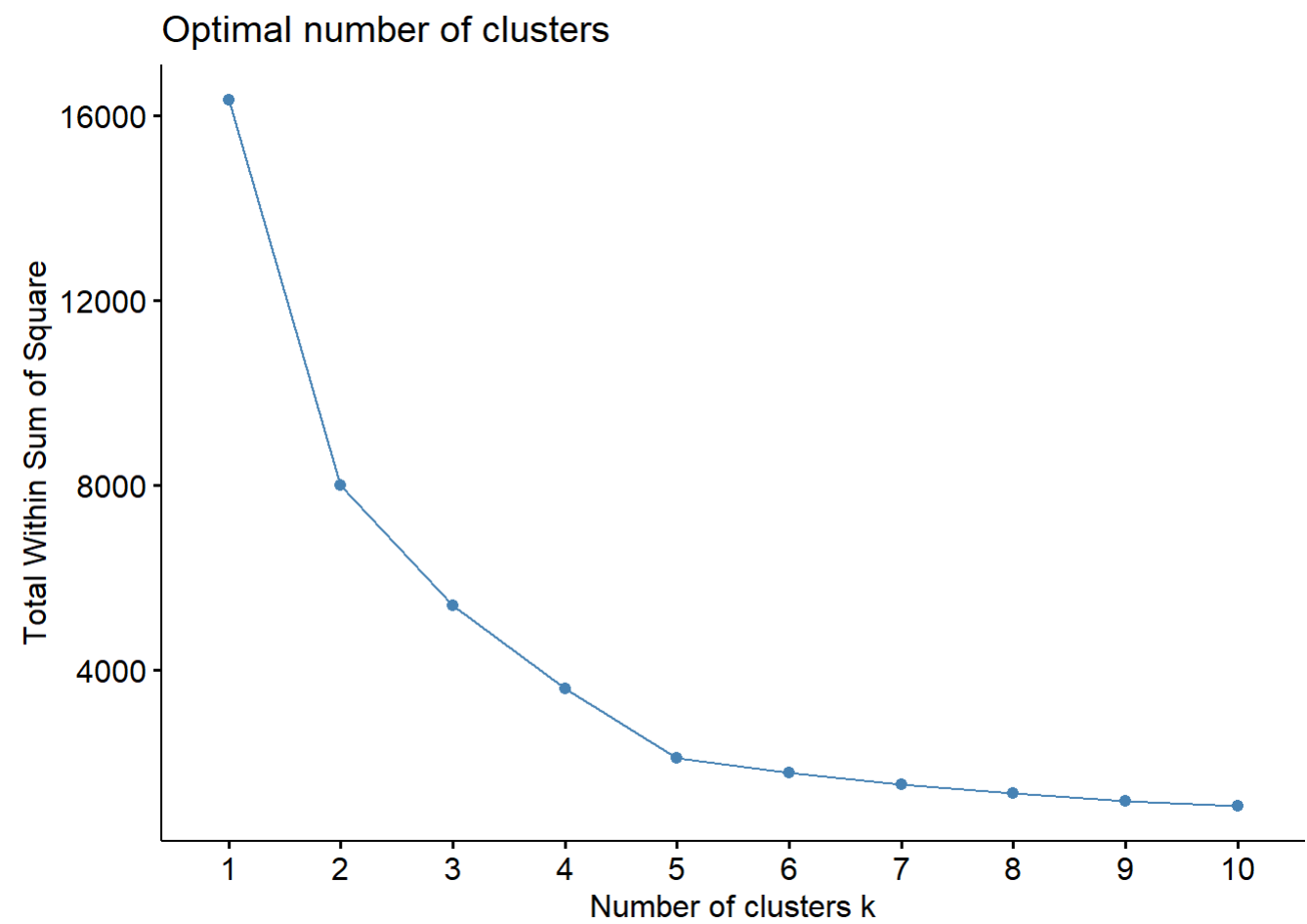
### 3. Shape points by Day

```
percentVar <- round(100 * attr(pcaData, "percentVar"))
ggplot(pcaData, aes(x = PC1, y = PC2, color = Subject, shape = Day)) +
  geom_point(size = 3) +
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance")) +
  coord_fixed() +
  ggtitle("PCA with VST data")
```

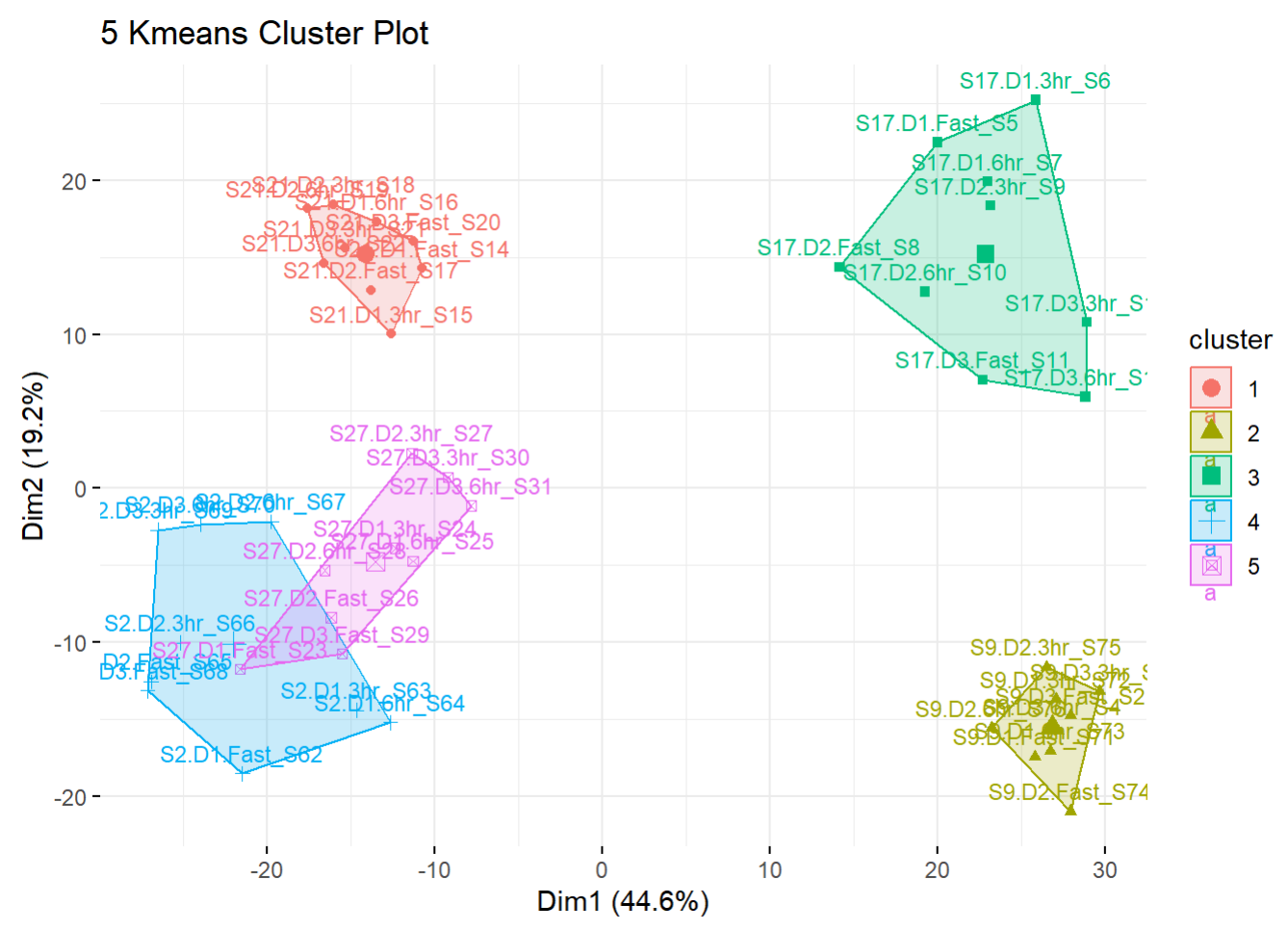


- Write a short script that perform hierarchical or kmeans clustering (choose one):
  1. Extract the data from “counts.vst” using the assay(function)
  2. Use only top 1,000 variable genes
  3. Calculate distance matrix
  4. Use the Elbow method to decide the optimal number of clusters
  5. Plot the results in a dendrogram for the hierarchical clustering or use fviz\_cluster() for kmeans

```
kmeans_data <- assay(counts.vst)
var_per_gene <- apply(kmeans_data, 1, var) # Calculate the variance per gene
selectedGenes <- names(var_per_gene[order(var_per_gene, decreasing = T)][1:1000]) # Take the top
1000 variable genes
kmeans_data.top1Kvar <- t(kmeans_data[selectedGenes,]) # Construct a new matrix only for the top
1000 genes
dist_mat <- dist(kmeans_data.top1Kvar, method = 'euclidean') # set distance map with euclidean distance
library(factoextra)
fviz_nbclust(kmeans_data.top1Kvar, FUN = hcut, method = "wss") # plot Elbow graph
```



```
kmeans_result <- kmeans(kmeans_data.top1Kvar, centers = 5, nstart = 123) # do kmeans algorithm
fviz_cluster(list(data = kmeans_data.top1Kvar, cluster = kmeans_result$cluster), labels = 9, main = "5 Kmeans Cluster Plot", ggtheme = theme_minimal()) # plot kmeans graph
```



- Looking at the PCA and the clustering you made above:
1. What is the main source of variation among samples in this study?
  2. What is the second source of variation?
  3. What led you to this conclusion?



Write your answer here:

(1)

The main source of variation among samples in the study is the subjects.

(2)

The second source of variance is the day of the experiment.

(3)

We can see in the PCA and by figure 1A in the manuscript that samples strongly cluster by subject. This implies that whole blood transcriptome samples differ more by individuals than they do by time or by diet.

Furthermore, K-means clustering visually supports this by grouping samples near each subject's centroid, reinforcing the notion that the most significant source of variability is the subject itself. Additionally, the Elbow method optimal five clusters, equal to the number of subjects.

The second significant source of variance identified is the day. This is evident from the patterns observed in PCA graph, where samples from the same subject on different test days exhibit a tendency to cluster more closely together compared to other samples from the same individual. Such a clustering pattern suggests that Time has big influence on variance.

- Hint: Look at Figure 1A in the manuscript.
- Make a second DESeq2 object called “dds2”
- This time make use of the information you gained from the PCA above to in the “design” argument

```
dds2 <- DESeqDataSetFromMatrix(countData=counts,
                                colData=metadata,
                                design= ~Subject+Day+Time,
                                tidy=FALSE)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

- How did your new object differ from the old one? Explain why your results should be better now.

Write your answer here:

The original design focused solely on the ~Time variable, implying that changes in gene expression were primarily influenced by time. This assumption overlooks other critical sources of variation, ignoring other potential sources of variability.

The new design formula includes Day, Time, and Subject as factors. We now acknowledge additional layers of complexity. By including these variables, the model accounts for day-to-day variations and individual differences among subjects, alongside the effects of time.

This change in the design helps us to control the natural variation in gene expression that occurs between individuals. In other words, we now control more of the source of variance.

- Run DESeq() and lfcShrink() again, save the results into “res2” and sort by adjusted p-value in “resOrdered2”

```
dds2 <- DESeq(dds2)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
res2 <- lfcShrink(dds2, coef='Time_3.hr.postprandial_vs_fasting' , type="apeglm")
```

```
## using 'apeglm' for LFC shrinkage. If used in published research, please cite:
##     Zhu, A., Ibrahim, J.G., Love, M.I. (2018) Heavy-tailed prior distributions for
##     sequence count data: removing the noise and preserving large differences.
##     Bioinformatics. https://doi.org/10.1093/bioinformatics/bty895
```

```
resOrdered2 <- res2[order(res2$padj),]
length(which(resOrdered2$padj < 0.05))
```

```
## [1] 2154
```

- How many significant genes (adjusted p-value < 0.05) did you get this time?

Write your answer here:  
2154

- Explain in short the meaning of each column in “resOrdered2”

Write your answer here:

"baseMean" - The average of the normalized count values across all samples.

"log2FoldChange" - The estimated fold change between the conditions coef in the lfcShrink function 'Time\_3.hr.postprandial\_vs\_fasting'.

Negative value indicates down regulation in the postprandial state after 3 hours compared to fasting, while a positive value indicates up regulation genes in the postprandial sample.

In log2 scale.

"lfcSE" - The standard error estimate for the log2 fold change estimate

"pvalue" - The p-value for the Wald test of whether the gene's expression changes significantly between comparison groups. A smaller p-value means stronger evidence against the null hypothesis.

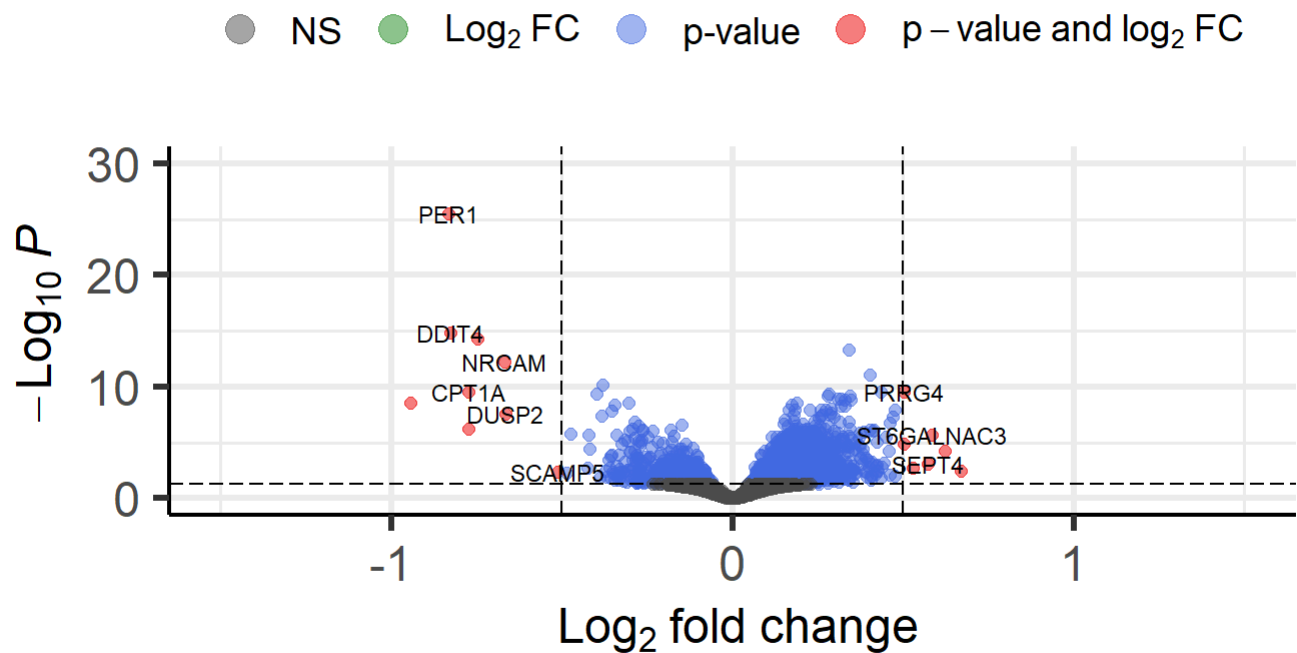
"padj" - Adjusted p-value, It adjusted the p-value to control the false rate. Lowering the chance to receive 'false-positive'.

- Visualize the results with a volcano plot as we did in the tutorial
- Change the following arguments to:
  1. FCcutoff=0.5
  2. pCutoff=0.05
  3. xlim=c(-1.5, 1.5)
  4. ylim=c(0, 30)

```
EnhancedVolcano::EnhancedVolcano(resOrdered2,
                                  lab = rownames(resOrdered2),
                                  x = 'log2FoldChange',
                                  y = 'padj',
                                  labSize = 3,
                                  title = "Volcano Plot of Differential Expression Genes",
                                  subtitle = "3 hr postprandial vs fasting",
                                  FCcutoff=0.5,
                                  pCutoff=0.05,
                                  xlim=c(-1.5, 1.5),
                                  ylim=c(0, 30))
```

# Volcano Plot of Differential Expression Genes

3 hr postprandial vs fasting



- Look at the volcano plot and explain in short the meaning of each axis:

log2FoldChange (x-axis) - Represents the magnitude of change in gene expression between two groups, with positive values indicating upregulation and negative values indicating downregulation in the group of interest compared to the reference group.

padj (y-axis) - The adjusted p-value for each gene, indicating the significance of the gene's expression change between the groups.

- Now let's try to understand the meaning of the differential expressed genes (DEG).
- Take a look at the genes names (rownames)

```
View(data.frame(resOrdered2))
```

- Pick several genes you like from the most DEG
- Google their names, try to find if they have anything in common
- In your answer, mention the genes and what you found about them

Write your answer here:

PER1 - is a gene involved in the regulation of the circadian rhythm, which is the body's internal clock that governs sleep-wake cycles and other physiological processes. It plays a critical role in maintaining the timing of the circadian rhythms, influencing various aspects of physical, mental, and behavioral changes that follow a daily cycle.

padj = 3.38286e-26

DDIT4 - is a stress-response gene that plays a significant role in regulating cellular growth and survival under adverse conditions, such as DNA damage and hypoxia (low oxygen levels). It acts primarily by inhibiting the mTOR signaling pathway, which is crucial for cell growth, proliferation, and survival.

padj = 1.50714e-15

HIF1A - plays a critical role in how cells respond to hypoxia. It's a regulatory protein that helps cells adapt to oxygen deficiency. When oxygen levels are low, HIF1A activates the transcription of several genes to increase oxygen supply (such as by promoting the formation of new blood vessels) and to adjust metabolism (for example, by increasing glucose uptake and metabolism under anaerobic conditions). HIF1A is involved in various processes including development, metabolism, and the body's response to conditions like chronic diseases such as cancer, where it can influence tumor growth and survival.

padj = 0.0568201

PDK4 - plays a crucial role in regulating how our body uses energy. It helps decide whether the body should burn glucose (sugar) for energy or store it as fat. Specifically, PDK4 inhibits the activity of an enzyme complex involved in converting glucose to energy, thereby increasing the utilization of fats as an energy source. This process is particularly active during fasting, prolonged exercise, or starvation.

padj = 3.10937e-09

SLC25A20 - is involved in the transport of acylcarnitines into mitochondria in exchange for free carnitine. This process is critical for the beta-oxidation of long-chain fatty acids in the mitochondria. Mutations in the SLC25A20 gene can lead to metabolic disorders, such as carnitine-acylcarnitine translocase deficiency, which is a condition that prevents the body from converting certain fats into energy, especially during periods without food.

padj = 4.67264e-15

\* mTOR - The mechanistic Target of Rapamycin coordinates eukaryotic cell growth and metabolism with environmental inputs including nutrients and growth factors.

The genes are related to metabolism, oxygen supply or basic daily function.

- Now you will use functional enrichment analysis with the Hallmark pathways gene sets.
- We learned two ways to perform functional enrichment analysis - over representation and gene set enrichment analysis (GSEA). Here you will run GSEA.
- First we need to create an ordered vector by the log fold change with the gene symbols as names:

```
resOrdered2.fatDiet.nona <- resOrdered2[!is.na(resOrdered2$padj) & resOrdered2$log2FoldChange > 0,]  
genes_ordered <- sort(resOrdered2.fatDiet.nona$log2FoldChange, decreasing = T)
```

- We now need to get the Hallmarks pathways gene sets. We will use the msigdb package for that:

```
hallmarks <- msigdb(species = "Homo sapiens", category = "H") %>%  
  dplyr::select(gs_name, gene_symbol)
```

- Use the GSEA() function from the clusterProfiler package to run the analysis.
- You can find an awesome tutorial for the clusterProfiler package here: <http://yulab-smu.top/biomedical-knowledge-mining-book/universal-api.html> (<http://yulab-smu.top/biomedical-knowledge-mining-book/universal-api.html>)
- Save your results into "hm"

```
hm <- GSEA(genes_ordered, TERM2GENE = hallmarks)
```

```
## preparing geneSet collections...
```

```
## GSEA analysis...
```

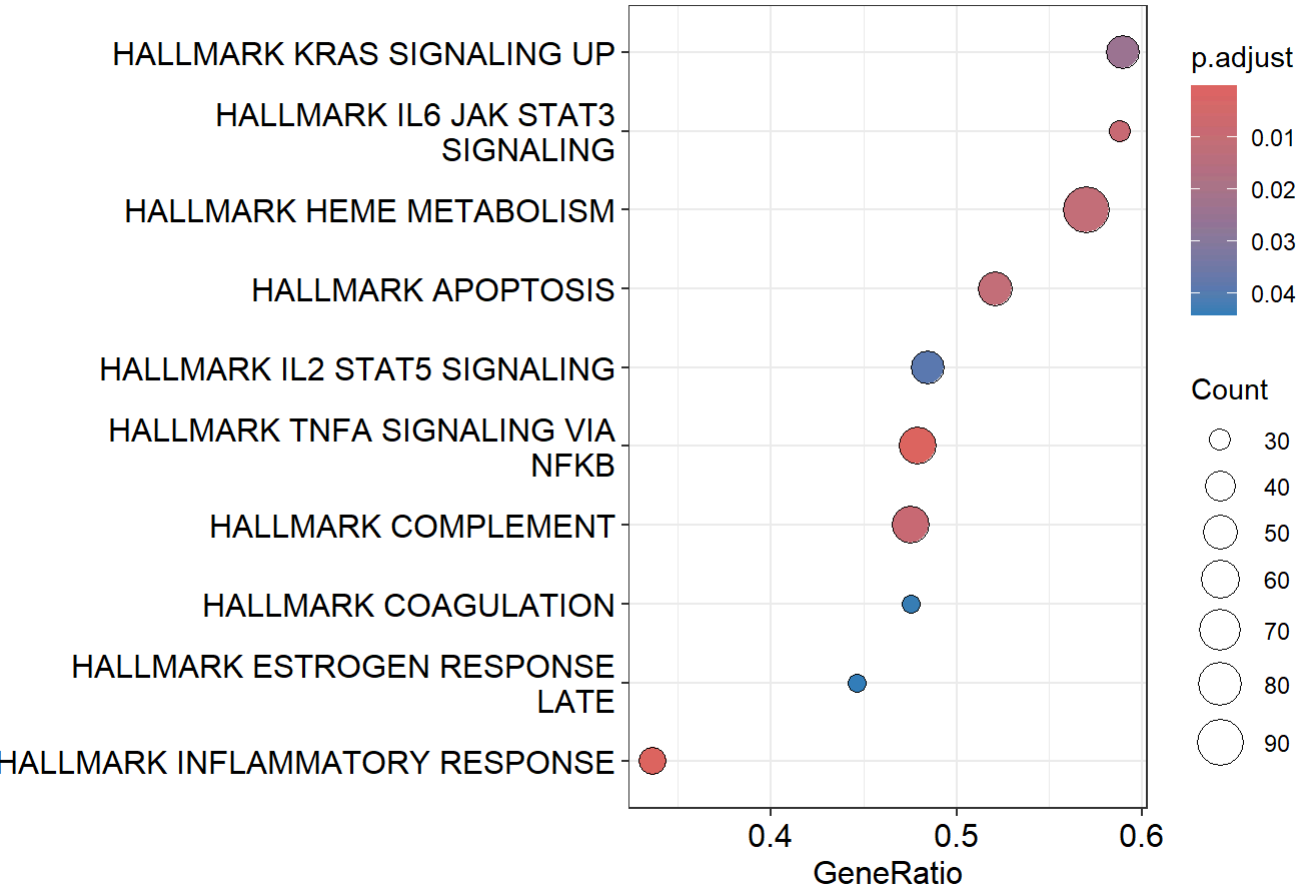
```
## Warning in preparePathwaysAndStats(pathways, stats, minSize, maxSize,  
## gseaParam, : All values in the stats vector are greater than zero and scoreType  
## is "std", maybe you should switch to scoreType = "pos".
```

```
## leading edge analysis...
```

```
## done...
```

- Finally, visualize the results of this analysis using the dotPlot function from the clusterProfiler database.

```
hallmarks <- msigdb(species = "Homo sapiens", category = "H")  
hallmarks <- hallmarks[,c('gs_name', 'gene_symbol')]  
dotplot(hm)
```



Summarize your conclusions from the analysis (up to 250 words) - Make sure you base your conclusions on the different steps you made (PCA, clustering, DEG, GSEA)

Write your answer here:

In summary, our analysis highlights that genes associated with innate immune responses exhibit differential expression during the postprandial period following a high-fat meal. From the results we concluded the main variance is by subject, suggesting that whole blood transcriptomes differ more among individuals than they do by time or by high-fat challenge meal. Though the most significant hallmark was inflammatory response, suggesting that all subjects had, in some expression level, some pattern of expression of innate immunity genes.

The conclusion relies on the PCA. With the PCA we saw what the main source of variance between the samples is. We concluded it is the Subject. The kmeans supported our finding by grouping samples near each subject's centroid. We also saw that the high-fat meal had a different time factor on each subject. Individuals have different times, what could lead to misleading conclusions as did past research.

From digging in the resOrdered2 and highlighting a few genes we saw a connection between genes related to metabolism, energy consumption and daily regulation. Some of the genes, like HIF1A have a strong connection to chronic diseases such as cancer.

The GSEA gave us the important notation that the most significant hallmarks are the inflammatory response and the TNFA signaling via NFkB, both refer to pathways that are central to the inflammatory response and immune system regulation. Showing us the impact of high-fat meals on innate immune system impact in the postprandial blood.

- [Save this HW as HTML](#)

Eat healthy :)