Capstone Project Analysis Pipeline

Syeda Zaki

10/7/2021

Overview

In this report we describe RNASeq Analysis pipeline to determine the differentially expressed genes between fetal and adult human brains.We aligned the reads to the human genome (b37hg19) with spliced aligner HISAT2. We generated counts with featureCounts() of subread package and create a phenotype table. Then we explore the data using edgeR package and do the differential expression analysis using DESeq2 package of Bioconductor. Finally, we do gene set analysis to determine if the genes that are differentially expressed between fetal and adult brain are associated with changes in H3K4me3 in their promoters, between the fetal and adult brains and com[pare it with liver tissue as a control.

Dependencies

This document depends on the following packages:

```
library(gplots)
library(devtools)
library(Biobase)
library(SummarizedExperiment)
library(org.Hs.eg.db)
library(EnhancedVolcano)
library(AnnotationDbi)
library(AnnotationHub)
library(dplyr)
library(tibble)
library(edgeR)
library("TxDb.Hsapiens.UCSC.hg19.knownGene")
library(DESeq2)
```

Alignment

Download the raw data

Only files with SRR1* accessions for both fetal and adult samples(1st run/replicate for each biosample) were used for downstream analysis.

Fetal Samples: SRR1554537, SRR1554538, SRR1554541, SRR1554566, SRR1554567, SRR1554568
Adult Samples: SRR1554535, SRR1554536, SRR1554539, SRR1554556, SRR1554561, SRR1554534

Data (SRA files) downloaded with the help of command line code from the AWS location of SRA link on NCBI website for each biosample listed in Week2 (https://www.coursera.org/learn/genomic-data-science-project/supplement/0HhAY/which-samples-to-use (https://www.coursera.org/learn/genomic-data-science-project/supplement/0HhAY/which-samples-to-use)). (`https://sra-pub-run-odp.s3.amazonaws.com/sra/SRR1*/SRR1*`)

Extract FASTQ files:

The tool fasterq-dump was used to extract FASTQ files from the SRA accessions above using Command Line.
(`~Desktop/GenomicsCapstone$sratoolkit.2.11.0-ubuntu64/bin/fasterq-dump/SRR1*`)

Check quality of reads:

Run fastqc on reads from command line(interactive mode)

Align the reads to the Human Reference Genome

- The reference human genome (b37)hg19 was downloaded from (`http://hgdownload.cse.ucsc.edu/goldenpath/hg19/bigZips/`), download file: hg19.fa.gz 2018-08-21 12:56 905M
- file was unzipped (`gunzip.hg19.fa.gz`) and saved as hg19.fa
- hisat2 built in command for creating index was used to build an index for the reference human genome. (`hisat2-build hg19.fa hg19index.*.ht2`)
- hisat2 was used to align mate1 files SRR1*_1.fastq and mate2 files SRR1*_2.fastq to the reference human genome index and the output was saved to SRR1*.sam files. (`hisat2 -p10 -x hg19index -1 SRR1*_1 -2 SRR1*_2 -S SRR1*.sam`)
- Alignment Summary/Results were reported for each sample
- Phenotype Table: Information was collected for each sample and stored in a phenotype table.

FastQC the Alignments

Run FastQC from command line in interactive mode and then do fastQc on each alignment file and determine the trend in average quality scores of all mapped reads. (`fastqc`) Use Excel to compute the average mapping rate for the fetal and adult samples

Get Feature Counts

Gene counts were obtained using featureCounts from subread package in command line for all sample alignment files * Sort alignment files (`samtools sort SRR1*.sam SRR1*.sorted.sam`) * Get Annotation file from Genecode and unzip and save as hg19.gtf
(`wget http://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_3c/gencode.v3c.annotation.GRCh37.gtf.gz`) * Get counts using featureCounts() from subread package for all the *.sorted.bam files*
(`featureCoutnts -T 20 -p -F 'GTF' -a hg19.gtf -o bam.featureCounts *.sorted.bam`) Get the feature counts from the file (`cut -f1,7-19 bam.featureCounts > gene_counts`) Save the gene expression table in “feature_counts.txt” tab-delimited text file

Explorartory Analysis

```
#Gene Expression Data
#########################
#Read in the gene expression file with counts and preprocess
gene_counts <- read.delim("feature_counts.txt", stringsAsFactors = FALSE)
rownames(gene_counts) <- gene_counts[,1]
gene_counts <- gene_counts[,2:13]
new_genes <- rownames(gene_counts)
new_genes <- sub('\\.[0-9]*$', '', new_genes)
rownames(gene_counts) <- new_genes
keep <- rowSums(gene_counts) >= 10
gene_counts <- gene_counts[keep,]
```

```
#Phenotype Data
###############
#Read in the file with sample data
p_data <- read.delim("PhenotypeData2.txt", header=TRUE, stringsAsFactors = FALSE)
cols <- c("sample.id","type","isolate","age","sex","tissue","disease","race","RIN","fraction","BioSamp
le","avg_per_base_seq_qual")
p_data[cols] <- lapply(p_data[cols],factor)
```

```
#Feature Data
#############
#Create feature data frame and add annotation
feature_data <- data.frame(rownames(gene_counts))
#Add annotation to feature data
rownames(feature_data) <- rownames(gene_counts)
cols = c("GENENAME","ENSEMBL","SYMBOL")
feature_annots <- AnnotationDbi::select(org.Hs.eg.db, keys=new_genes, columns=cols, keytype="REFSEQ")
feature_annots <- feature_annots[!duplicated(feature_annots[,1]), ]
feature_df <- merge(feature_data, feature_annots, by.x=0,by.y="REFSEQ")
feature_df <- select(feature_df, -1)
```

Create SummarizedExperiment

```
p_data$type <- relevel(p_data$type, "fetal")
brain_data <- SummarizedExperiment(assays=list(counts=gene_counts),  colData = p_data)
rownames(colData(brain_data)) <- p_data$sample.id
```

Explore the data using edgeR

```
brain_dge <- DGEList(counts = assay(brain_data, "counts"), group=brain_data$type)
brain_dge$samples <- merge(brain_dge$samples, as.data.frame(colData(brain_data)),by=0)
brain_dge$genes <- data.frame(name = rownames(brain_dge$counts))
keep2 <- filterByExpr(brain_dge)
filt_brain_dge <- brain_dge[keep2,keep.lib.sizes=FALSE]
filt_brain_dge <- estimateDisp(filt_brain_dge)
```

Counts per million

```
#Compute counts per million
cpm <- cpm(filt_brain_dge)
cpm_df <- as.data.frame(cpm)
filt_cpm <- filter(cpm_df,rowMeans(cpm) > 1000)
coloramp=colorRampPalette(c(2,"white",4))(9)
heatmap.2(as.matrix(filt_cpm),col=coloramp,Rowv=NA,Colv=NA,dendogram="none",scale="row",trace="none",
 main="CPM")
```

Plots

```
#Do the log Transform
filtdata_transpose <- log2(filt_brain_dge$counts + 1)
par(mfrow=c(1,4))
#With Rotated Axis labels
boxplot(filt_brain_dge$counts,col=3,main="Without Transform",xaxt="n")
axis(side=1, labels=FALSE)
text(x=1:(length(p_data)-1),y=par("usr")[3]-0.45,labels=p_data$sample.id,xpd=NA,srt=35,adj=1,cex=1.2)
boxplot(filtdata_transpose,col=4,pch=19,main="Log2 Transformed",xaxt="n")
axis(side=1, labels=FALSE)
text(x=1:(length(p_data)-1),y=par("usr")[3]-0.45,labels=p_data$sample.id,xpd=NA,srt=35,adj=1,cex=1.2)
plotMDS(filt_brain_dge,col=c(rep("blue",6),rep("red",6)),main="MDS Plot by Sample Type")
plotBCV(filt_brain_dge,main="BCV plot")
```

Principal Component Analysis

```
#Center the data and compute principal components and color by different variables
filtdata_centered <- filtdata_transpose - rowMeans(filtdata_transpose)
svd <- svd(filtdata_centered)
par(mfrow=c(1,4))
plot(svd$v[,1],svd$v[,2],ylab="2nd PC", xlab="1st PC",pch=19,lwd=2,col=as.numeric(p_data$sex),main='PC
colored by Sex of the Sample',cex.main=1.25)
legend("topright",legend=levels(p_data$sex),pch=19,col=1:2)
plot(svd$v[,1],svd$v[,2],ylab="2nd PC", xlab="1st PC",pch=19,lwd=2,col=as.numeric(p_data$race),main="P
Cs colored by Race of the Sample")
legend("topright",legend=levels(p_data$race),pch=19,col=1:2)
plot(svd$v[,1],svd$v[,2],ylab="2nd PC", xlab="1st PC",pch=19,lwd=2,col=as.numeric(p_data$RIN),main="PC
s colored by RIN of the Sample")
legend("topright",legend=levels(p_data$RIN),pch=19,col=1:11)
plot(svd$v[,1],svd$v[,2],ylab="2nd PC", xlab="1st PC",pch=19,lwd=2,col=as.numeric(p_data$type),main="P
Cs colored by Type of Sample")
legend("topright",legend=levels(p_data$type),pch=19,col=1:2)
```

Statistical Analysis using DESeq2

Hypothesis

H0 - Null hypothesis is that the relationship between sample type or age (fetal vs adult) and expression is exactly zero.

H1 - Alternative Hypothesis is that the relationship between sample type or age is not zero

We use the un-normalzed count data matrix which we obtained from feature counts as input to DESeq2.This matrix gives the number of reads that were assigned to each gene for each sample.

```
#Create DESeq object from counts matrix obtained from feature counts in week 6.
brain_dds <- DESeqDataSetFromMatrix(gene_counts, p_data, design = ~ type)
```

Differential Expression Analysis

```
#Filter to remove lowly expressed genes
keep3 <- rowSums(counts(brain_dds)) >= 10
brain_dds <- brain_dds[keep3]
brain_dds$type
#Differential Expression Analysis
brain_dds <- DESeq(brain_dds)
brain_res <- results(brain_dds)
brain_res05 <- results(brain_dds,alpha = 0.05)
sum(brain_res05$padj < 0.05, na.rm = TRUE)
brain_res05Ordered <- brain_res05[order(brain_res05$pvalue), ]
keep4 <- !is.na(brain_res05Ordered$padj)
brain_05padj <- brain_res05Ordered[keep4,]
brain_05padj <- brain_05padj[brain_05padj$padj < 0.05,]
#Get the list of genes
brain_05genes <- rownames(brain_05padj)
#Write the genes differentially expressed at FDR 0.05 to file
write.csv(brain_05genes, file="diffexp_05FDRgenes")
```

We save the list of differentially expressed genes for downstream analysis

Visualization with Enhanced Volcano Plot

```
#Annotate genes with symbols
genes <- rownames(brain_dds)
symbols <- mapIds(org.Hs.eg.db, keys = genes, column=c("SYMBOL"), keytype = "REFSEQ")
symbols <- symbols[!is.na(symbols)]
brain_dds2 <- brain_dds
rownames(brain_dds2) <- symbols
keep5 <- !is.na(rownames(brain_dds2))
brain_dds2 <- brain_dds2[keep5,]
#LFC Shrink to visualize and rank
brain_resLFC <- lfcShrink(brain_dds2, coef="type_adult_vs_fetal", type="apeglm")
#Enhanced Volcano Plot
EnhancedVolcano(brain_resLFC, lab = rownames(brain_resLFC),x = 'log2FoldChange', y = 'pvalue', title =
'logFold change vs -log 10 p-value')
```

Gene Set Analysis

We take the differentially expressed genes list “diffexp_05FDRgenes” and go to UCSC Table Browser and download the start and end coordinates for the transcripts of the genes and other metadata and save it to file “diffexp_05genes”. We get the promoters for these genes and then examine whether these are marked by H3Kme3.

```
#Read in the file preprocess and convert to GRnages Object
diffexp_genes <- read.delim("diffexp_05genes",header = TRUE)
diffexp_genes_df <- DataFrame(diffexp_genes)
keep_genes <- !duplicated(diffexp_genes_df$name)
diffexp_genes_df <- diffexp_genes_df[keep_genes,]
#Convert into GRanges object
diff_exp_gr <- makeGRangesFromDataFrame(diffexp_genes_df, start.field = "txStart", end.field = "txEnd"
,keep.extra.columns = TRUE)
```

Fetal brain: Enrichment of promoters in differentially expressed genes of fetal samples

```
#Get the fetal brain peak data from roadmap project from Annotation Hub
ah <- AnnotationHub()
ah <- subset(ah, species == "Homo sapiens")
qhs <- query(ah, c("EpigenomeRoadMap","Fetal_Brain","H3K4me3","narrowPeak"))
# AH44720 | UCSF-UBC.Fetal_Brain.H3K4me3.HuFNSC02.narrowPeak.gz
fetal_gr <- qhs[['AH44720']]
fetal_peaks <- fetal_gr
#Find the promoters for diff_exp_gr
prom <- promoters(diff_exp_gr)
ov_fetal <- findOverlaps(prom,fetal_peaks)
length(unique(queryHits(ov_fetal)))
length(unique(subjectHits(ov_fetal)))
length(subsetByOverlaps(fetal_peaks, prom, ignore.strand = TRUE))/length(fetal_peaks)
length(subsetByOverlaps(prom, fetal_peaks, ignore.strand = TRUE))/length(prom)
sum(width(reduce(fetal_peaks, ignore.strand = TRUE)))/10^6
sum(width(reduce(prom, ignore.strand = TRUE)))/10^6
sum(width(intersect(fetal_peaks, prom, ignore.strand = TRUE)))/10^6
inOut_fetal <- matrix(0, ncol = 2, nrow = 2)
colnames(inOut_fetal) <- c("in", "out")
rownames(inOut_fetal) <- c("in", "out")
inOut_fetal[1,1] <- sum(width(intersect(fetal_peaks, prom, ignore.strand = TRUE)))
inOut_fetal[1,2] <- sum(width(setdiff(fetal_peaks, prom, ignore.strand = TRUE)))
inOut_fetal[2,1] <- sum(width(setdiff(prom, fetal_peaks, ignore.strand = TRUE)))
inOut_fetal[2,2] <- 3*10^9 - sum(inOut_fetal)
oddsRatio_fetal <- inOut_fetal[1,1] * inOut_fetal[2,2] / (inOut_fetal[2,1] * inOut_fetal[1,2])
```

Adult brain: Enrichment of promoters in differentially expressed genes

```
ahub <- AnnotationHub()
ahub <- subset(ahub, species == "Homo sapiens")
qhs_a <- query(ahub, c("EpigenomeRoadMap","H3K4me3","Brain","narrowPeak"))
#AH43565 | BI.Brain_Mid_Frontal_Lobe.H3K4me3.112.narrowPeak.gz
adult_peaks <- qhs_a[["AH43565"]]
ov_adult <- findOverlaps(prom,adult_peaks)
length(unique(queryHits(ov_adult)))
length(subsetByOverlaps(adult_peaks, prom, ignore.strand = TRUE))
length(subsetByOverlaps(adult_peaks, prom, ignore.strand = TRUE))/length(adult_peaks)
length(subsetByOverlaps(prom, adult_peaks, ignore.strand = TRUE))/length(prom)
sum(width(reduce(adult_peaks, ignore.strand = TRUE)))/10^6
sum(width(reduce(prom, ignore.strand = TRUE)))/10^6
sum(width(intersect(adult_peaks, prom, ignore.strand = TRUE)))/10^6
inOut_adult <- matrix(0, ncol = 2, nrow = 2)
colnames(inOut_adult) <- c("in", "out")
rownames(inOut_adult) <- c("in", "out")
inOut_adult[1,1] <- sum(width(intersect(adult_peaks, prom, ignore.strand = TRUE)))
inOut_adult[1,2] <- sum(width(setdiff(adult_peaks, prom, ignore.strand = TRUE)))
inOut_adult[2,1] <- sum(width(setdiff(prom, adult_peaks, ignore.strand = TRUE)))
inOut_adult[2,2] <- 3*10^9 - sum(inOut_adult)
oddsRatio_adult <- inOut_adult[1,1] * inOut_adult[2,2] / (inOut_adult[2,1] * inOut_adult[1,2])
```

Adult Liver: Enrichment of promoters in differentially expressed genes in adult liver

```
ah_l <- AnnotationHub()
ah_l <- subset(ah_l, species == "Homo sapiens")
qhs_liver <- query(ah_l, c("EpigenomeRoadMap","H3K4me3","Liver","narrowPeak"))
#AH43450 | BI.Adult_Liver.H3K4me3.3.narrowPeak.gz
liver_peaks<- qhs_liver[["AH43450"]]
ov_liver <- findOverlaps(prom,liver_peaks)
length(unique(queryHits(ov_liver)))
length(subsetByOverlaps(liver_peaks, prom, ignore.strand = TRUE))/length(liver_peaks)
length(subsetByOverlaps(prom, liver_peaks, ignore.strand = TRUE))/length(prom)
sum(width(reduce(liver_peaks, ignore.strand = TRUE)))/10^6
sum(width(reduce(prom, ignore.strand = TRUE)))/10^6
sum(width(intersect(liver_peaks, prom, ignore.strand = TRUE)))/10^6
inOut_liver <- matrix(0, ncol = 2, nrow = 2)
colnames(inOut_liver) <- c("in", "out")
rownames(inOut_liver) <- c("in", "out")
inOut_liver[1,1] <- sum(width(intersect(liver_peaks, prom, ignore.strand = TRUE)))
inOut_liver[1,2] <- sum(width(setdiff(liver_peaks, prom, ignore.strand = TRUE)))
inOut_liver[2,1] <- sum(width(setdiff(prom, liver_peaks, ignore.strand = TRUE)))
inOut_liver[2,2] <- 3*10^9 - sum(inOut_liver)
oddsRatio_liver <- inOut_liver[1,1] * inOut_liver[2,2] / (inOut_liver[2,1] * inOut_liver[1,2])
```

Odds Ratio: The odds ratio tells us how much more enriched the overlap between peaks and promoters is than we would expect.

Command Line tools

Ubuntu 20.04.1

fastqc version 0.11.9

sratoolkit version 2.11.0

HISAT2 version 2.1.0

samtools version 1.2

subread version 2.0.0

```
devtools::session_info()
```

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.1.0 (2021-05-18)
##  os       Windows 10 x64
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  English_United States.1252
##  ctype    English_United States.1252
##  tz       America/Indianapolis
##  date     2021-10-10
##
## - Packages -------------------------------------------------------------------
##  package                         * version  date       lib
##  annotate                          1.70.0   2021-05-19 [1]
##  AnnotationDbi                   * 1.54.1   2021-06-08 [1]
##  assertthat                        0.2.1    2019-03-21 [1]
##  Biobase                         * 2.52.0   2021-05-19 [1]
##  BiocFileCache                     2.0.0    2021-05-19 [1]
##  BiocGenerics                    * 0.38.0   2021-05-19 [1]
##  BiocIO                            1.2.0    2021-05-19 [1]
##  BiocManager                       1.30.16  2021-06-15 [1]
##  BiocParallel                      1.26.0   2021-05-19 [1]
##  biomaRt                           2.48.2   2021-07-01 [1]
##  Biostrings                        2.60.0   2021-05-20 [1]
##  bit                               4.0.4    2020-08-04 [1]
##  bit64                             4.0.5    2020-08-30 [1]
##  bitops                            1.0-7    2021-04-24 [1]
##  blob                              1.2.2    2021-07-23 [1]
##  bslib                             0.3.0    2021-09-02 [1]
##  cachem                            1.0.5    2021-05-15 [1]
##  callr                             3.7.0    2021-04-20 [1]
##  caTools                           1.18.2   2021-03-28 [1]
##  cli                               3.0.0    2021-06-30 [1]
##  codetools                         0.2-18   2020-11-04 [2]
##  colorspace                        2.0-1    2021-05-04 [1]
##  crayon                            1.4.1    2021-02-08 [1]
##  curl                              4.3.2    2021-06-23 [1]
##  DBI                               1.1.1    2021-01-15 [1]
##  dbplyr                            2.1.1    2021-04-06 [1]
##  DelayedArray                      0.18.0   2021-05-19 [1]
##  desc                              1.4.0    2021-09-28 [1]
##  DESeq2                          * 1.32.0   2021-05-19 [1]
##  devtools                        * 2.4.2    2021-06-07 [1]
##  digest                            0.6.27   2020-10-24 [1]
##  dplyr                           * 1.0.6    2021-05-05 [1]
##  edgeR                           * 3.34.0   2021-05-19 [1]
##  ellipsis                          0.3.2    2021-04-29 [1]
##  evaluate                          0.14     2019-05-28 [1]
##  fansi                             0.4.2    2021-01-15 [1]
##  fastmap                           1.1.0    2021-01-25 [1]
##  filelock                          1.0.2    2018-10-05 [1]
##  fs                                1.5.0    2020-07-31 [1]
##  genefilter                        1.74.0   2021-05-19 [1]
##  geneplotter                       1.70.0   2021-05-19 [1]
##  generics                          0.1.0    2020-10-31 [1]
##  GenomeInfoDb                    * 1.28.0   2021-05-19 [1]
```

```
##  GenomeInfoDbData                  1.2.6    2021-05-29 [1]
##  GenomicAlignments                 1.28.0   2021-05-19 [1]
##  GenomicFeatures                 * 1.44.0   2021-05-19 [1]
##  GenomicRanges                   * 1.44.0   2021-05-19 [1]
##  ggplot2                           3.3.5    2021-06-25 [1]
##  glue                              1.4.2    2020-08-27 [1]
##  gplots                          * 3.1.1    2020-11-28 [1]
##  gtable                            0.3.0    2019-03-25 [1]
##  gtools                            3.9.2    2021-06-06 [1]
##  highr                             0.9      2021-04-16 [1]
##  hms                               1.1.1    2021-09-26 [1]
##  htmltools                         0.5.2    2021-08-25 [1]
##  httr                              1.4.2    2020-07-20 [1]
##  IRanges                         * 2.26.0   2021-05-19 [1]
##  jquerylib                         0.1.4    2021-04-26 [1]
##  jsonlite                          1.7.2    2020-12-09 [1]
##  KEGGREST                          1.32.0   2021-05-19 [1]
##  KernSmooth                        2.23-20  2021-05-03 [2]
##  knitr                             1.36     2021-09-29 [1]
##  lattice                           0.20-44  2021-05-02 [2]
##  lifecycle                         1.0.1    2021-09-24 [1]
##  limma                           * 3.48.0   2021-05-19 [1]
##  locfit                            1.5-9.4  2020-03-25 [1]
##  magrittr                          2.0.1    2020-11-17 [1]
##  Matrix                            1.3-3    2021-05-04 [2]
##  MatrixGenerics                  * 1.4.3    2021-08-26 [1]
##  matrixStats                     * 0.61.0   2021-09-17 [1]
##  memoise                           2.0.0    2021-01-26 [1]
##  munsell                           0.5.0    2018-06-12 [1]
##  org.Hs.eg.db                    * 3.13.0   2021-07-21 [1]
##  pillar                            1.6.3    2021-09-26 [1]
##  pkgbuild                          1.2.0    2020-12-15 [1]
##  pkgconfig                         2.0.3    2019-09-22 [1]
##  pkgload                           1.2.2    2021-09-11 [1]
##  png                               0.1-7    2013-12-03 [1]
##  prettyunits                       1.1.1    2020-01-24 [1]
##  processx                          3.5.2    2021-04-30 [1]
##  progress                          1.2.2    2019-05-16 [1]
##  ps                                1.6.0    2021-02-28 [1]
##  purrr                             0.3.4    2020-04-17 [1]
##  R6                                2.5.1    2021-08-19 [1]
##  rappdirs                          0.3.3    2021-01-31 [1]
##  RColorBrewer                      1.1-2    2014-12-07 [1]
##  Rcpp                              1.0.6    2021-01-15 [1]
##  RCurl                             1.98-1.3 2021-03-16 [1]
##  remotes                           2.4.1    2021-09-29 [1]
##  restfulr                          0.0.13   2017-08-06 [1]
##  rjson                             0.2.20   2018-06-08 [1]
##  rlang                             0.4.11   2021-04-30 [1]
##  rmarkdown                         2.11     2021-09-14 [1]
##  rprojroot                         2.0.2    2020-11-15 [1]
##  Rsamtools                         2.8.0    2021-05-19 [1]
##  RSkittleBrewer                  * 1.1      2021-09-16 [1]
##  RSQLite                           2.2.7    2021-04-22 [1]
##  rtracklayer                       1.52.0   2021-05-19 [1]
##  S4Vectors                       * 0.30.0   2021-05-19 [1]
##  sass                              0.4.0    2021-05-12 [1]
```

```
##  scales                              1.1.1    2020-05-11 [1]
##  sessioninfo                         1.1.1    2018-11-05 [1]
##  stringi                             1.6.2    2021-05-17 [1]
##  stringr                             1.4.0    2019-02-10 [1]
##  SummarizedExperiment              * 1.22.0   2021-05-19 [1]
##  survival                            3.2-11   2021-04-26 [2]
##  testthat                            3.0.4    2021-07-01 [1]
##  tibble                            * 3.1.2    2021-05-16 [1]
##  tidyselect                          1.1.1    2021-04-30 [1]
##  TxDb.Hsapiens.UCSC.hg19.knownGene * 3.2.2    2021-09-30 [1]
##  usethis                           * 2.0.1    2021-02-10 [1]
##  utf8                                1.2.1    2021-03-12 [1]
##  vctrs                               0.3.8    2021-04-29 [1]
##  withr                               2.4.2    2021-04-18 [1]
##  xfun                                0.24     2021-06-15 [1]
##  XML                                 3.99-0.6 2021-03-16 [1]
##  xml2                                1.3.2    2020-04-23 [1]
##  xtable                              1.8-4    2019-04-21 [1]
##  XVector                             0.32.0   2021-05-19 [1]
##  yaml                                2.2.1    2020-02-01 [1]
##  zlibbioc                            1.38.0   2021-05-19 [1]
##  source
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.0)
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.0)
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
```

```
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.0)
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.1)
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.1)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
```

```
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  Github (alyssafrazee/RSkittleBrewer@3bd1c98)
##  CRAN (R 4.1.0)
##  Bioconductor
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  CRAN (R 4.1.0)
##  Bioconductor
##  CRAN (R 4.1.0)
##  Bioconductor
##
## [1] C:/Users/syeda/Documents/R/win-library/4.1
## [2] C:/Program Files/R/R-4.1.0/library
```