Ardalan Ahmadvand

Introduction

In this project, I would like to answer the following question: is the gene expression level different between fetals and adults? To achieve this goal, I will re-perform the analysis which described in this paper: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4281298/. In this research, authors collected 48 different samples and studied 6 different age groups from fetal to old. Since the data is quite big and some following steps are very time-consuming, I only collect 3 samples for each fetal (<0 years) and adult (20-50 years) group.

Getting the raw data

I downloaded all 6 datasets from European Nucleotide Archive: https://www.ebi.ac.uk/ena/browser/home. Three of them were fetal datasets: SRR1554537, SRR1554566, SRR1554568. The rest three datasets were adult datasets: SRR1554536, SRR1554539, SRR1554534. Each file was paired-end library, and there were two fastq files for each sample. For example, SRR1554537 contains SRR1554537_1 and SRR1554537_2, each file was in fastq.gz format.

Align the samples to reference genome

I used galaxy (https://usegalaxy.org/) to align the samples to its reference genome. First, the six datasets were uploaded to the server with the following settings: filetype was fastqsanger.gz; reference genome was hg19. Next, I used HISAT2 (Version 2.1.0+galaxy5) to make the alignment: the reference genome was built-in genome Human (Homo Sapiens)(b37) hg19; Paired-end two files for each sample. When the program was finished, there were two files: a BAM file contained alignment results, and a summary file described the alignment quality.

Quality control on the alignments

I used FastQC (Version 0.72+galaxy1) on galaxy server to perform the quality control. Number of reads were in the range of 21,450,348 to 55,133,946. Percentage of GC were in range 46 to 51. All the 6 alignment rates were close to 99.7%, average quality per read were 37 or 38, which indicate dthe alignment results were good and the quality of reads were good.

Next I would like to answer the questions: is the mapping rates similar for fetal and adult samples? First, I collected some information for each sample from: https://www.ncbi.nlm.nih.gov/sra?linkname=bioproject_sra_all&from_uid=245228. I selected “Send to - File - Download Full XML”, then I used python to parse the file to a csv file “Week5_QC_data.csv”.

```
f = read.csv("Week5_QC_data.csv")

phenotype_table = f

rownames(phenotype_table) = phenotype_table[,1]

phenotype_table[,1] = NULL

head(phenotype_table)
```

```
##                          SAMPLE     Age age.group RIN    sex race
```

```
## SRR1554534 R2857_DLPFC_polyA+_transcriptome 40.4200     adult 8.4   male   AA
## SRR1554536   R3098_DLPFC_polyA_RNAseq_total 44.1700     adult 5.3 female   AA
## SRR1554539   R3467_DLPFC_polyA_RNAseq_total 36.5000     adult 9.0 female   AA
## SRR1554537   R3452_DLPFC_polyA_RNAseq_total -0.3836     fetal 9.6 female   AA
## SRR1554566   R4706_DLPFC_polyA_RNAseq_total -0.4986     fetal 8.3   male HISP
## SRR1554568   R4708_DLPFC_polyA_RNAseq_total -0.4986     fetal 8.0   male   AA
##           Total.sequences alignment.rate Average.Quality.per.read X.GC
## SRR1554534        60234973         99.71                    37   51
## SRR1554536        45172764         99.86                    37   46
## SRR1554539        70679196         99.72                    38   48
## SRR1554537       118410185          99.73                    37   48
## SRR1554566       115556403          99.75                    37   48
## SRR1554568       103480309          99.75                    37   47
write.table(phenotype_table, file="phenotype.txt", col.names=TRUE, row.names=TRUE)
```

The summary of the adult and fetal group was shown below:

```
adult = f[1:3,]
fetal = f[4:6,]


summary(adult)
##         RUN                            SAMPLE       Age        age.group
##  SRR1554534:1   R2857_DLPFC_polyA+_transcriptome:1   Min.   :36.50   adult:3
##  SRR1554536:1   R3098_DLPFC_polyA_RNAseq_total  :1   1st Qu.:38.46   fetal:0
##  SRR1554537:0   R3452_DLPFC_polyA_RNAseq_total  :0   Median :40.42
##  SRR1554539:1   R3467_DLPFC_polyA_RNAseq_total  :1   Mean   :40.36
##  SRR1554566:0   R4706_DLPFC_polyA_RNAseq_total  :0   3rd Qu.:42.30
##  SRR1554568:0   R4708_DLPFC_polyA_RNAseq_total  :0   Max.   :44.17
##       RIN           sex      race   Total.sequences    alignment.rate
##  Min.   :5.300   female:2   AA  :3   Min.   :45172764   Min.   :99.71
##  1st Qu.:6.850   male  :1   HISP:0   1st Qu.:52703868   1st Qu.:99.72
##  Median :8.400                      Median :60234973   Median :99.72
##  Mean   :7.567                      Mean   :58695644   Mean   :99.76
##  3rd Qu.:8.700                      3rd Qu.:65457084   3rd Qu.:99.79
##  Max.   :9.000                      Max.   :70679196   Max.   :99.86
```

```
##  Average.Quality.per.read      X.GC
##  Min.   :37.00            Min.   :46.00
##  1st Qu.:37.00            1st Qu.:47.00
##  Median :37.00            Median :48.00
##  Mean   :37.33            Mean   :48.33
##  3rd Qu.:37.50            3rd Qu.:49.50
##  Max.   :38.00            Max.   :51.00
```

```
summary(fetal)
```

```
##         RUN                             SAMPLE       Age
##  SRR1554534:0   R2857_DLPFC_polyA+_transcriptome:0   Min.   :-0.4986
##  SRR1554536:0   R3098_DLPFC_polyA_RNAseq_total  :0   1st Qu.:-0.4986
##  SRR1554537:1   R3452_DLPFC_polyA_RNAseq_total  :1   Median :-0.4986
##  SRR1554539:0   R3467_DLPFC_polyA_RNAseq_total  :0   Mean   :-0.4603
##  SRR1554566:1   R4706_DLPFC_polyA_RNAseq_total  :1   3rd Qu.:-0.4411
##  SRR1554568:1   R4708_DLPFC_polyA_RNAseq_total  :1   Max.   :-0.3836
##  age.group      RIN            sex      race    Total.sequences
##  adult:0   Min.   :8.000   female:1   AA  :2   Min.   :103480309
##  fetal:3   1st Qu.:8.150   male  :2   HISP:1   1st Qu.:109518356
##            Median :8.300                       Median :115556403
##            Mean   :8.633                       Mean   :112482299
##            3rd Qu.:8.950                       3rd Qu.:116983294
##            Max.   :9.600                       Max.   :118410185
##  alignment.rate  Average.Quality.per.read      X.GC
##  Min.   :99.73   Min.   :37              Min.   :47.00
##  1st Qu.:99.74   1st Qu.:37              1st Qu.:47.50
##  Median :99.75   Median :37              Median :48.00
##  Mean   :99.74   Mean   :37              Mean   :47.67
##  3rd Qu.:99.75   3rd Qu.:37              3rd Qu.:48.00
##  Max.   :99.75   Max.   :37              Max.   :48.00
```

To determine whether the mapping rates and average quality score were different betweem adult and fetal group, I perfromed the student's t-test:

```
t.test(fetal$alignment.rate, adult$alignment.rate)
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  fetal$alignment.rate and adult$alignment.rate
## t = -0.4092, df = 2.0758, p-value = 0.7208
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2231072  0.1831072
## sample estimates:
## mean of x mean of y
##  99.74333  99.76333
```

```
t.test(fetal$Average.Quality.per.read, adult$Average.Quality.per.read)
```

```
##
##  Welch Two Sample t-test
##
## data:  fetal$Average.Quality.per.read and adult$Average.Quality.per.read
## t = -1, df = 2, p-value = 0.4226
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1.767551  1.100884
## sample estimates:
## mean of x mean of y
##  37.00000  37.33333
```

The p-values were 0.7208 and 0.4226 for mapping rate and average quality score between the two groups, which indicate there was no signifiant different between the two groups.

Get feature counts

To calculate the abundance of every gene in every sample, I used featureCounts (Version 1.6.4+galaxy1) on galaxy server, the gene annotation genome was hg19. The results were tables that was formatted with one gene per row with correponding counts. After performed on each sample, I merged the all 6 tabulars into one table by their gene_ids, and converted them to gene names.

```
library('tidyverse')
library(org.Hs.eg.db)
library(annotate)
# read feature count files
tabular_files = list.files(path = "/Users/miotomato/Documents/Genomics Data Science Specialization/8. Genomic Data Science Capstone/Data/FeatureCount", pattern = "tabular$", full.names = TRUE)
```

```
tabular_list = lapply(tabular_files, read.table)


# merge the files by gene_id
feature_count_files = Reduce(function(x, y) merge(x, y, by="V1"), tabular_list)
colnames(feature_count_files) = c("gene_id", "SRR1554534", "SRR1554536", "SRR1554537",  "SRR1554539", "SRR1554566", "SRR1554568")
feature_count_files = feature_count_files[c("gene_id", "SRR1554534", "SRR1554536", "SRR1554539", "SRR1554537", "SRR1554566", "SRR1554568")]


# convert gene_id to gene_name
for (i in 1:nrow(feature_count_files)){
  feature_count_files[i,1] = lookUp(toString(feature_count_files[i,1]), 'org.Hs.eg', 'SYMBOL')
}
rownames(feature_count_files) = make.names(feature_count_files[,1], unique=TRUE)
feature_count_files[,1] = NULL
feature_table = feature_count_files
head(feature_table)
##          SRR1554534 SRR1554536 SRR1554539 SRR1554537 SRR1554566 SRR1554568
## A1BG            219         48        133        242        304        149
## A2M            4496       5577       6446       3753       5866       3439
## NAT1             10         12         21         25         40         22
## NAT2              6          0          4          2          4          2
## SERPINA3        156         38         22          6          6          0
## AADAC             0          0          0          0          0          0
write.table(feature_table, file="feature_counts.txt", sep='\t', row.names=TRUE, col.names=TRUE)
```

Here is a workflow for the entire galaxy jobs:

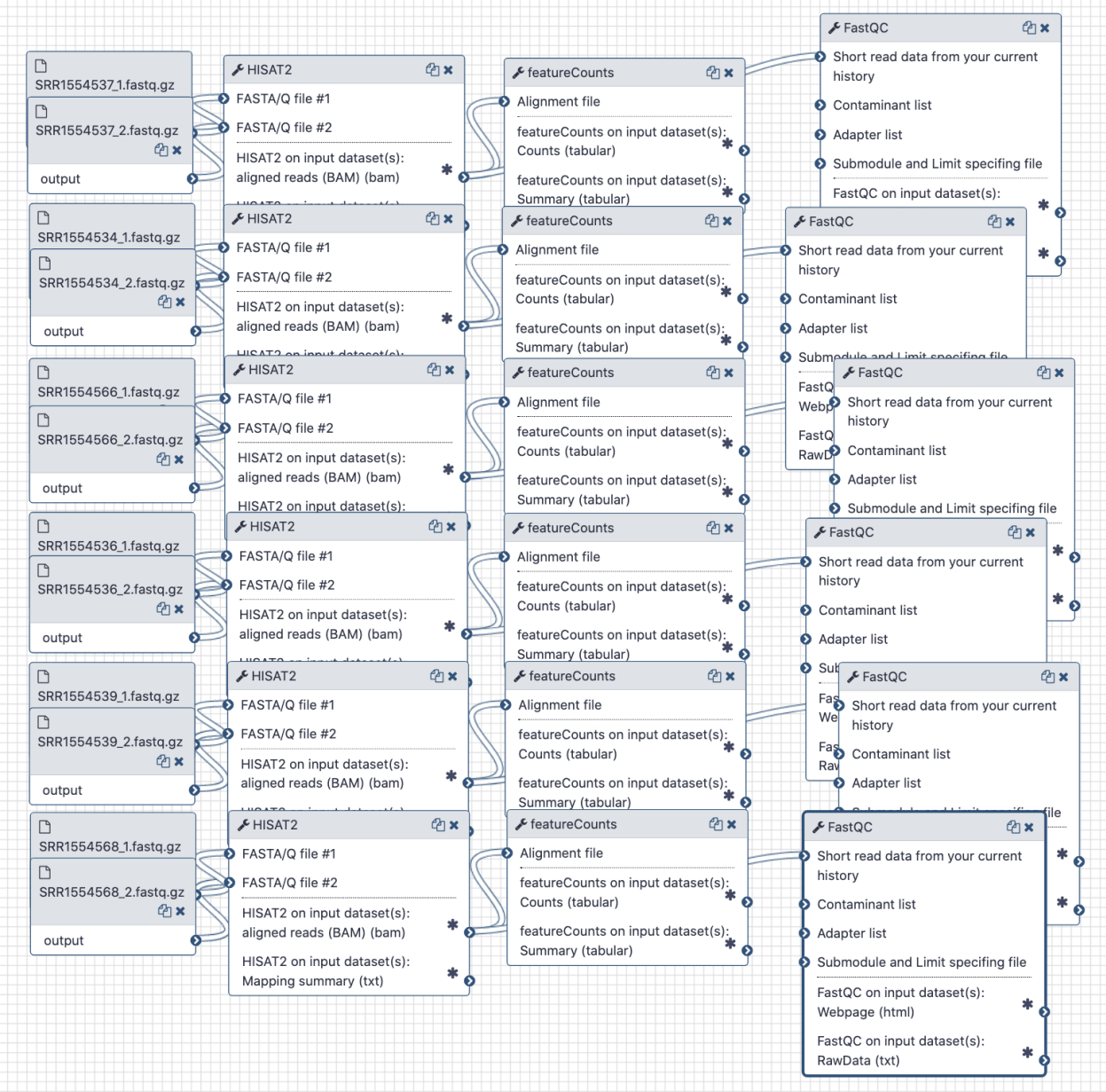


Figure 1. Galaxy workflow for alignment, QC and feature counting.

For the further analysis, I will use R only.

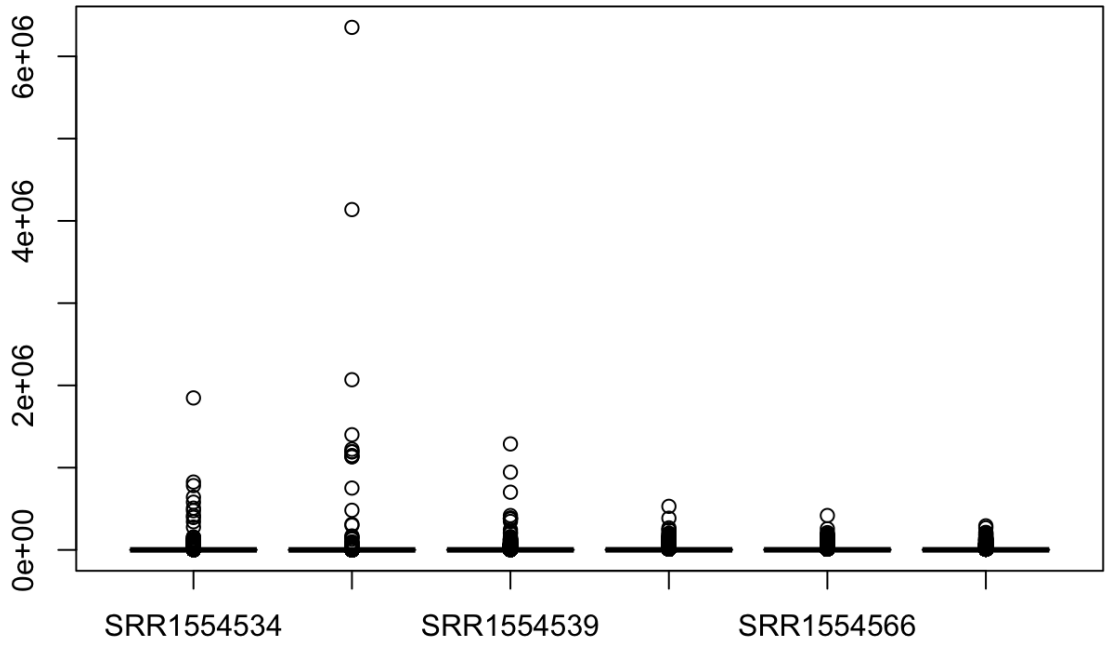
Exploratory analysis

```
library(GenomicRanges)
library(SummarizedExperiment)
library(edgeR)
# remove low expression data
feature_table = feature_table[rowMeans(feature_table) > 10, ]
```

```
# create a SummarizedExperiment data
col_data = phenotype_table
row_data = relist(GRanges(), vector("list", length=nrow(feature_table)))
se = SummarizedExperiment(assays = list(counts = feature_table), rowRanges = row_data, colData = col
_data)
se
```

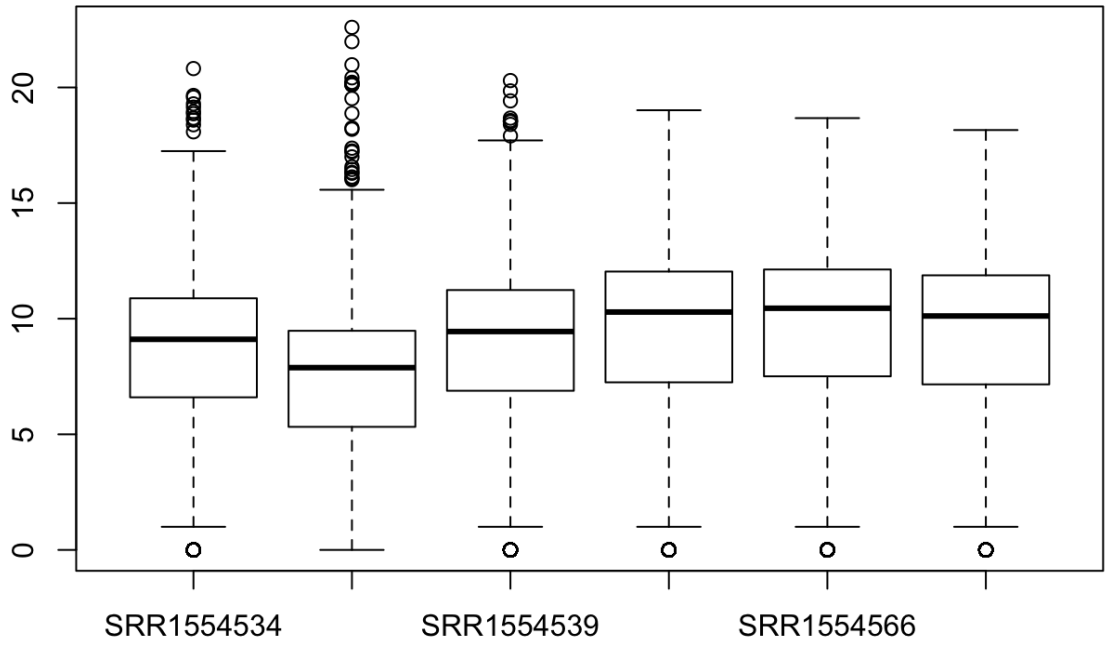
```
## class: RangedSummarizedExperiment
## dim: 18403 6
## metadata(0):
## assays(1): counts
## rownames(18403): A1BG A2M ... NA..1946 NA..1948
## rowData names(0):
## colnames(6): SRR1554534 SRR1554536 ... SRR1554566 SRR1554568
## colData names(10): SAMPLE Age ... Average.Quality.per.read X.GC
```

```
# make a boxplot of the expression levels for each sample
dge <- DGEList(counts = assay(se, "counts"), group = phenotype_table$age.group )
dge$samples <- merge(dge$samples, as.data.frame(colData(se)), by = 0)
boxplot(dge$counts)
```



Most of the data push to the bottom in the boxplot, so I perform log2 transformation on the data.

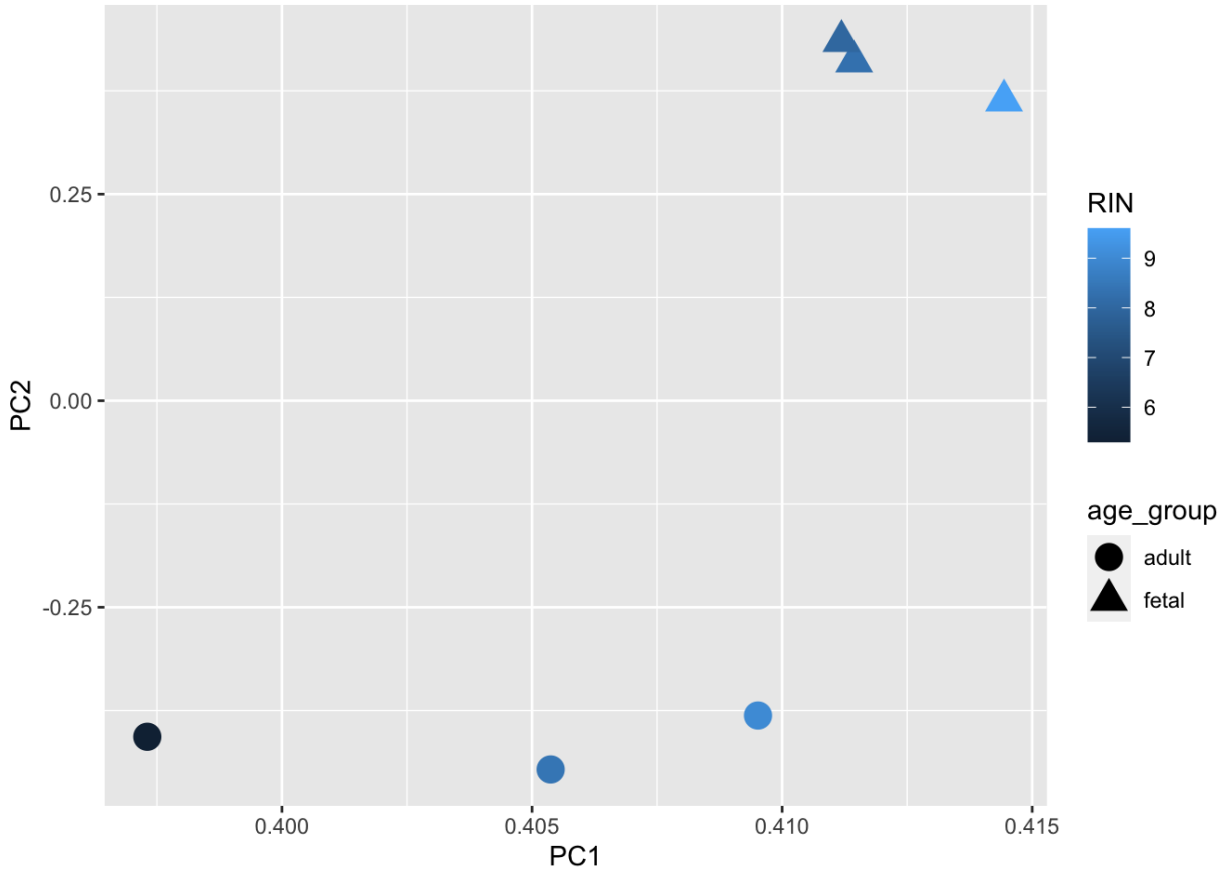
```
log2_dge_count = log2(dge$counts + 1)
boxplot(log2_dge_count)
```



Now the boxplot looked much better. It seems many outliers with extremly high expression in adult data but not in fetal data.

Next I performed a principal component analysis.

```
library(ggfortify)

# perform PCA

count_pca = prcomp(log2_dge_count, center=TRUE, scale=TRUE)

dat = data.frame(X=count_pca$rotation[,1], Y=count_pca$rotation[,2], age_group=phenotype_table$age.group, RIN=phenotype_table$RIN)


# scatterplot using PC1 and PC2, colored by RIN, shaped by age.group

ggplot(dat, aes(x=X, y=Y, shape=age_group, color=RIN)) + geom_point(size=5) + xlab("PC1") + ylab("PC2")
```



Adult gene expression and fetal gene expression data were easily differentiate by PC1 and PC2. If we only use RIN, we cannot distinguish adult and fetal group.

Statistical analysis

Next I will perform a statistical analysis to detect genes that are differentially expressed. The null and alternative hypothesis is:
H0: the mean gene expression level of each gene is equal between adult and fetal samples.
H1: the mean gene expression level of each gene is not equal between adult and fetal samples.
To test the hypothesis, I used limma package to make linear models for the assessment of differential expression.

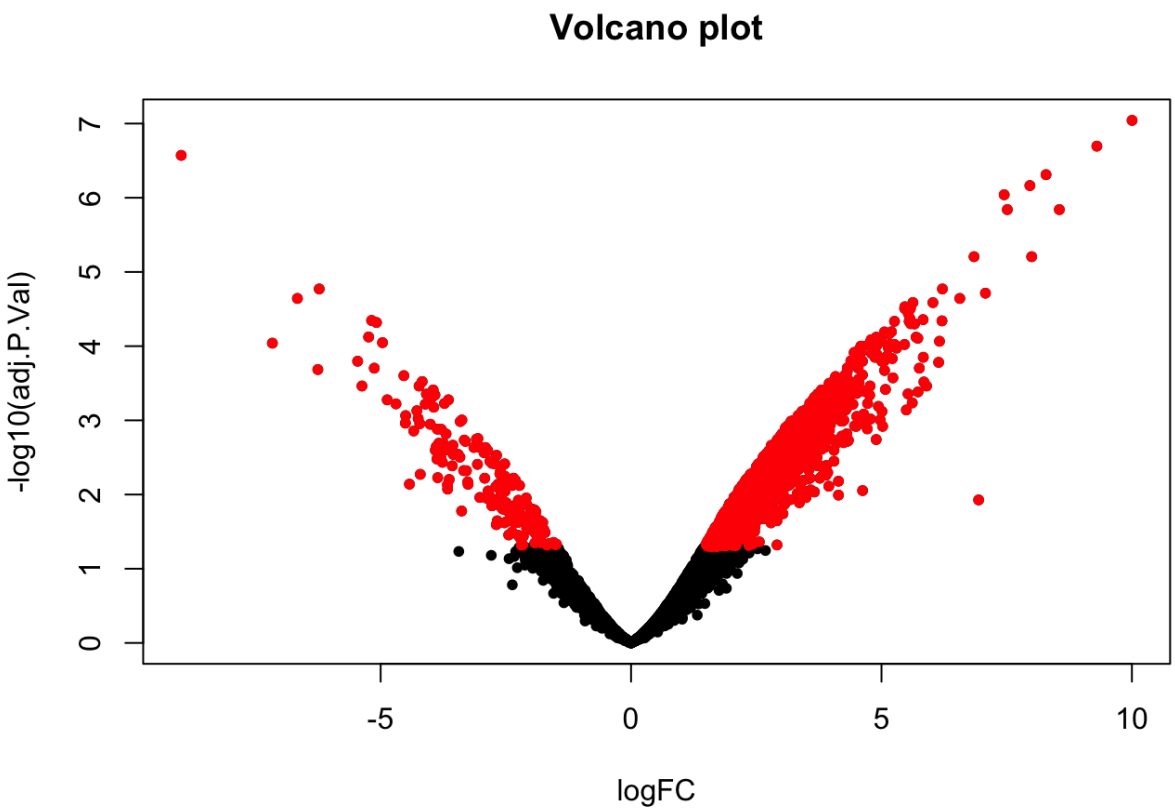
```
library(limma)
library(edge)


# make log2 transformation and remove low expression
edata = assay(se)
edata = log2(as.matrix(edata) + 1)
edata = edata[rowMeans(edata) > 10, ]
```

```
# fit the model by age.group and write results to a tab-delimited file with gene name, log2 fold-change, p-value and adjusted p-value
mod = model.matrix(~ se$age.group)
fit_limma = lmFit(edata,mod)
ebayes_limma = eBayes(fit_limma)
limma_toptable = topTable(ebayes_limma,number=dim(edata)[1])
limma_table_output = limma_toptable[,c(1,4,5)]
write.table(limma_table_output, file="dif_exp_genes.txt", sep='\t', row.names=TRUE, col.names=TRUE)
head(limma_table_output)
```

```
##          logFC      P.Value    adj.P.Val
## SOX11 10.003633 1.137319e-11 9.054194e-08
## MEX3A  9.300562 5.064686e-11 2.015998e-07
## MBP   -8.981780 1.011567e-10 2.684361e-07
## SOX4   8.285515 2.455308e-10 4.886676e-07
## SLA    7.963284 4.306779e-10 6.857253e-07
## GPC2   7.449973 6.866159e-10 9.110249e-07
```

Then I made a volano plot, which is a plot of the fold change for age in each linear model versus log10 p-value.

```
# make the volcano plot, mark those gene with p-value less than 0.05 as red
with(limma_toptable, plot(logFC, -log10(adj.P.Val), pch=20, main="Volcano plot"))
with(subset(limma_toptable, adj.P.Val < 0.05), points(logFC, -log10(adj.P.Val), pch=20, col="red"))
```



Those genes show in red dots were considered as differentially expressed. Next I summarized those genes:

```
print(paste0("Genes differentially expressed: ", sum(limma_table_output$adj.P.Val < 0.05)))
```

```
## [1] "Genes differentially expressed: 3864"
```

```
print(paste0("Genes differentially expressed and down-regulated from fetal to adult: ", sum(limma_table_output$adj.P.Val < 0.05 & limma_table_output$logFC > 1)))
```

```
## [1] "Genes differentially expressed and down-regulated from fetal to adult: 3693"
```

```
print(paste0("Genes differentially expressed and up-regulated from fetal to adult: ", sum(limma_table_output$adj.P.Val < 0.05 & limma_table_output$logFC < -1)))
```

```
## [1] "Genes differentially expressed and up-regulated from fetal to adult: 171"
```

There were total 3,864 genes differentially expressed betweem adult and fetal samples. Among those genes, 3,693 were down-regulated and 171 were up-regulated from fetal to adult.

Gene set analysis

To do a further analysis, I would like to know:
1. If those differentially expressed genes between fetal and adult are associated with changes in H3K4me3 in their promoters.
2. Whether promoters for the list of differentially expressed genes are marked by H3K4me3 in liver.

First, fetal brain, adult prefrontal cortex and adult liver datasets from roadmap epigenomics project are downloaded using AnnotationHub and TxDb.Hsapiens.UCSC.hg19.knownGene.

```
library(AnnotationHub)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)


ah <- AnnotationHub()
ah <- subset(ah, species == "Homo sapiens")
ah_fetal <- query(ah, c("EpigenomeRoadMap", "H3K4me3", "E081"))
ah_adult <- query(ah, c("EpigenomeRoadMap", "H3K4me3", "E073"))
ah_liver <- query(ah, c("EpigenomeRoadMap", "H3K4me3", "E066"))


# download narrowPeak datasets
fetal_gr <- ah_fetal[[2]]
adult_gr <- ah_adult[[2]]
liver_gr <- ah_liver[[2]]
```

Notice I used gene name in the previous study, but roadmap uses entrez gene id, so I converted the gene name to the corresponding gene id.

```
library(mygene)
dif_exp_genes = row.names(limma_toptable[limma_toptable$adj.P.Val < 0.05,])
dif_exp_gene_ids = queryMany(dif_exp_genes, scopes = "symbol", fields = "entrezgene", species = "hum
an" )
```

Extract differentially expressed genes corresponding promoters.

```
txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
txdb_genes <- genes(txdb)
dif_exp_promoters <- promoters(txdb_genes[dif_exp_gene_ids$entrezgene %in% txdb_genes$gene_id])
```

Find the overlap between differentially expressed gene promoters and narrowPeak datasets.

```
adult_perc_peak = length(subsetByOverlaps(adult_gr, dif_exp_promoters, ignore.strand=TRUE)) / length(
adult_gr)
fetal_perc_peak = length(subsetByOverlaps(fetal_gr, dif_exp_promoters, ignore.strand=TRUE)) / length(f
etal_gr)
liver_perc_peak = length(subsetByOverlaps(liver_gr, dif_exp_promoters, ignore.strand=TRUE)) / length(li
ver_gr)
print(paste0("Percentage of differentially expressed gene in adult narrowpeaks: ", round(adult_perc_peak,
3)))
```

```
## [1] "Percentage of differentially expressed gene in adult narrowpeaks: 0.243"
```

```
print(paste0("Percentage of differentially expressed gene in fetal narrowpeaks: ", round(fetal_perc_peak, 3)))
```

```
## [1] "Percentage of differentially expressed gene in fetal narrowpeaks: 0.364"
```

```
print(paste0("Percentage of differentially expressed gene in adult liver narrowpeaks: ", round(liver_perc_peak, 3)))
```

```
## [1] "Percentage of differentially expressed gene in adult liver narrowpeaks: 0.201"
```

Notice that 36.5% of promoters which associate differentially expressed genes overlaps with fetal brain narrowpeak data, 24.3% overlaps with adult brain narrowpeak data, and 20.1% overlaps with adult liver narrowpeak data.

To do a further analysis, I calculate the odds ratio for adult brain, fetal brain and adult liver by creating a 2*2 overlap matrix.

```
odds_ratio = function(prom_counts, peak_counts, print=TRUE){
overlapMat <- matrix(0,, ncol = 2, nrow = 2)
colnames(overlapMat) <- c("in.peaks", "out.peaks")
rownames(overlapMat) <- c("in.promoters", "out.promoter")

prom <- reduce(prom_counts, ignore.strand = TRUE)
peaks <- reduce(peak_counts)
both <- intersect(prom, peaks)
only.prom <- setdiff(prom, both)
only.peaks <- setdiff(peaks, both)

overlapMat[1,1] <- sum(width(both))
overlapMat[1,2] <- sum(width(only.prom))
overlapMat[2,1] <- sum(width(only.peaks))
overlapMat[2,2] <- 1.5*10^9 - sum(overlapMat)

oddsRatio <- overlapMat[1,1] * overlapMat[2,2] / (overlapMat[2,1] * overlapMat[1,2])
return(oddsRatio)
}
```

```
print(paste0("Odds ratio in adult brain: ", round(odds_ratio(dif_exp_promoters, adult_gr), 2)))
```

```
## [1] "Odds ratio in adult brain: 16.06"
```

```
print(paste0("Odds ratio in fetal brain: ", round(odds_ratio(dif_exp_promoters, fetal_gr), 2)))
```

```
## [1] "Odds ratio in fetal brain: 18.37"
```

```
print(paste0("Odds ratio in adult liver: ", round(odds_ratio(dif_exp_promoters, liver_gr), 2)))
```

```
## [1] "Odds ratio in adult liver: 12.44"
```

The odds ratio of fetal brain sample is 18.43, indicate the overlap between fetal peaks and the promoters which associate with differentially expressed genes is about 18 fold more enriched than we would expect. The odds ratio of adult brain is 16.08, which a little less than fetal sample. The odds ratio of adult liver sample is 12.45, which less than fetal and adult brain sample, indicates less promoters for the list of fetal and adult brain differentially expressed genes are marked by H3K4me3 in liver sample.

SessionInfo

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS Catalina 10.15.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] mygene_1.22.0
##  [2] BSgenome.Hsapiens.UCSC.hg19_1.4.0
##  [3] BSgenome_1.54.0
##  [4] Biostrings_2.54.0
##  [5] XVector_0.26.0
##  [6] rtracklayer_1.46.0
##  [7] TxDb.Hsapiens.UCSC.hg19.knownGene_3.2.2
##  [8] GenomicFeatures_1.38.2
```

```
##  [9] AnnotationHub_2.18.0
## [10] BiocFileCache_1.10.2
## [11] dbplyr_1.4.3
## [12] edge_2.18.0
## [13] ggfortify_0.4.10
## [14] edgeR_3.28.1
## [15] limma_3.42.2
## [16] SummarizedExperiment_1.16.1
## [17] DelayedArray_0.12.3
## [18] BiocParallel_1.20.1
## [19] matrixStats_0.56.0
## [20] GenomicRanges_1.38.0
## [21] GenomeInfoDb_1.22.1
## [22] annotate_1.64.0
## [23] XML_3.99-0.3
## [24] org.Hs.eg.db_3.10.0
## [25] AnnotationDbi_1.48.0
## [26] IRanges_2.20.2
## [27] S4Vectors_0.24.4
## [28] Biobase_2.46.0
## [29] BiocGenerics_0.32.0
## [30] forcats_0.5.0
## [31] stringr_1.4.0
## [32] dplyr_0.8.5
## [33] purrr_0.3.4
## [34] readr_1.3.1
## [35] tidyr_1.0.2
## [36] tibble_3.0.1
## [37] ggplot2_3.3.0
## [38] tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##   [1] readxl_1.3.1                backports_1.1.6
##   [3] Hmisc_4.4-0                  plyr_1.8.6
```

```
##   [5] splines_3.6.3              gmp_0.5-13.6
##   [7] sva_3.34.0                 digest_0.6.25
##   [9] htmltools_0.4.0            fansi_0.4.1
##  [11] checkmate_2.0.0            magrittr_1.5
##  [13] memoise_1.1.0              cluster_2.1.0
##  [15] modelr_0.1.7               askpass_1.1
##  [17] prettyunits_1.1.1          jpeg_0.1-8.1
##  [19] colorspace_1.4-1           blob_1.2.1
##  [21] rvest_0.3.5                rappdirs_0.3.1
##  [23] haven_2.2.0                xfun_0.13
##  [25] crayon_1.3.4               RCurl_1.98-1.2
##  [27] jsonlite_1.6.1             snm_1.34.0
##  [29] genefilter_1.68.0          lme4_1.1-23
##  [31] survival_3.1-12            glue_1.4.0
##  [33] gtable_0.3.0               zlibbioc_1.32.0
##  [35] jackstraw_1.3              scales_1.1.0
##  [37] lfa_1.16.0                 DBI_1.1.0
##  [39] Rcpp_1.0.4.6               htmlTable_1.13.3
##  [41] xtable_1.8-4               progress_1.2.2
##  [43] foreign_0.8-76             bit_1.1-15.2
##  [45] rsvd_1.0.3                 sqldf_0.4-11
##  [47] Formula_1.2-3              htmlwidgets_1.5.1
##  [49] httr_1.4.1                 RColorBrewer_1.1-2
##  [51] acepack_1.4.1              ellipsis_0.3.0
##  [53] ClusterR_1.2.1             pkgconfig_2.0.3
##  [55] farver_2.0.3               nnet_7.3-14
##  [57] locfit_1.5-9.4             tidyselect_1.0.0
##  [59] labeling_0.3               rlang_0.4.6
##  [61] reshape2_1.4.4             later_1.0.0
##  [63] munsell_0.5.0              BiocVersion_3.10.1
##  [65] cellranger_1.1.0           tools_3.6.3
##  [67] cli_2.0.2                  gsubfn_0.7
##  [69] generics_0.0.2             RSQLite_2.2.0
##  [71] broom_0.5.6                evaluate_0.14
```

```
##  [73] fastmap_1.0.1               yaml_2.2.1
##  [75] knitr_1.28                  bit64_0.9-7
##  [77] fs_1.4.1                    nlme_3.1-147
##  [79] mime_0.9                    xml2_1.3.2
##  [81] biomaRt_2.42.1              compiler_3.6.3
##  [83] rstudioapi_0.11             png_0.1-7
##  [85] curl_4.3                    interactiveDisplayBase_1.24.0
##  [87] reprex_0.3.0                statmod_1.4.34
##  [89] stringi_1.4.6               lattice_0.20-41
##  [91] Matrix_1.2-18               nloptr_1.2.2.1
##  [93] vctrs_0.2.4                 pillar_1.4.3
##  [95] lifecycle_0.2.0             BiocManager_1.30.10
##  [97] data.table_1.12.8           bitops_1.0-6
##  [99] irlba_2.3.3                 corpcor_1.6.9
## [101] httpuv_1.5.2                qvalue_2.18.0
## [103] latticeExtra_0.6-29         R6_2.4.1
## [105] promises_1.1.0              gridExtra_2.3
## [107] boot_1.3-25                 MASS_7.3-51.6
## [109] gtools_3.8.2                assertthat_0.2.1
## [111] chron_2.3-55                proto_1.0.0
## [113] openssl_1.4.1               withr_2.2.0
## [115] GenomicAlignments_1.22.1    Rsamtools_2.2.3
## [117] GenomeInfoDbData_1.2.2      mgcv_1.8-31
## [119] hms_0.5.3                   rpart_4.1-15
## [121] grid_3.6.3                  minqa_1.2.4
## [123] rmarkdown_2.1               base64enc_0.1-3
## [125] shiny_1.4.0.2               lubridate_1.7.8
```