Project Title: Differential Gene Expression between Fetal and Adult Brains

List of Tasks

- Week4_Task2: Alignment
- Week5_Task3: QC
- Week6_Task4: Assemble & Gene Count Feature
- Week7_Task5: Exploratory Analysis
- Week8_Task6: Statistical Analysis
- Week9_Task7: Gene Set Analysis

Software

- ballgown version 2.22.0
- FastQC v0.11.9
- HISAT2 version 2.2.1
- samtools 1.11
- sratoolkit.2.10.9
- Stringtie version 2.1.4
- R version 4.0.4 (2021-02-15) -- “Lost Library Book “, platform: x86_64-apple-darwin17.0
- macOS 11.2.3 (20D91), Terminal/Bash5.1

Downloaded Files

- Human genome hisat2 index: https://genome-idx.s3.amazonaws.com/hisat/grch38_genome.tar.gz
- SRR1554534(R2857), SRR1554535(R2869), SRR1554536(R3452), SR1554538(R3462), SRR1554561(R4166), SRR1554568(R4707)
- Human genome annotations:http://ftp.ensembl.org/pub/release-103/gtf/homo_sapiens/Homo_sapiens.GRCh38.103.gtf.gz

Week4_Task2: Alignment

1. Retrieve RNA-seq reads from SRA database**

```
$ prefetch SRR1554534
$ fasterq-dump SRR1554534
```

2. Align the paired reads to the reference genome using HISAT2 index

```
$ hisat2 -p 8 --dta -x grch38/genome -1 Adults/ R2857_ SRR1554534_1.fastq -2 R2857_
SRR1554534_2.fastq -S R2857.sam
```

Note: the same steps were performed for other SRRs.

Result

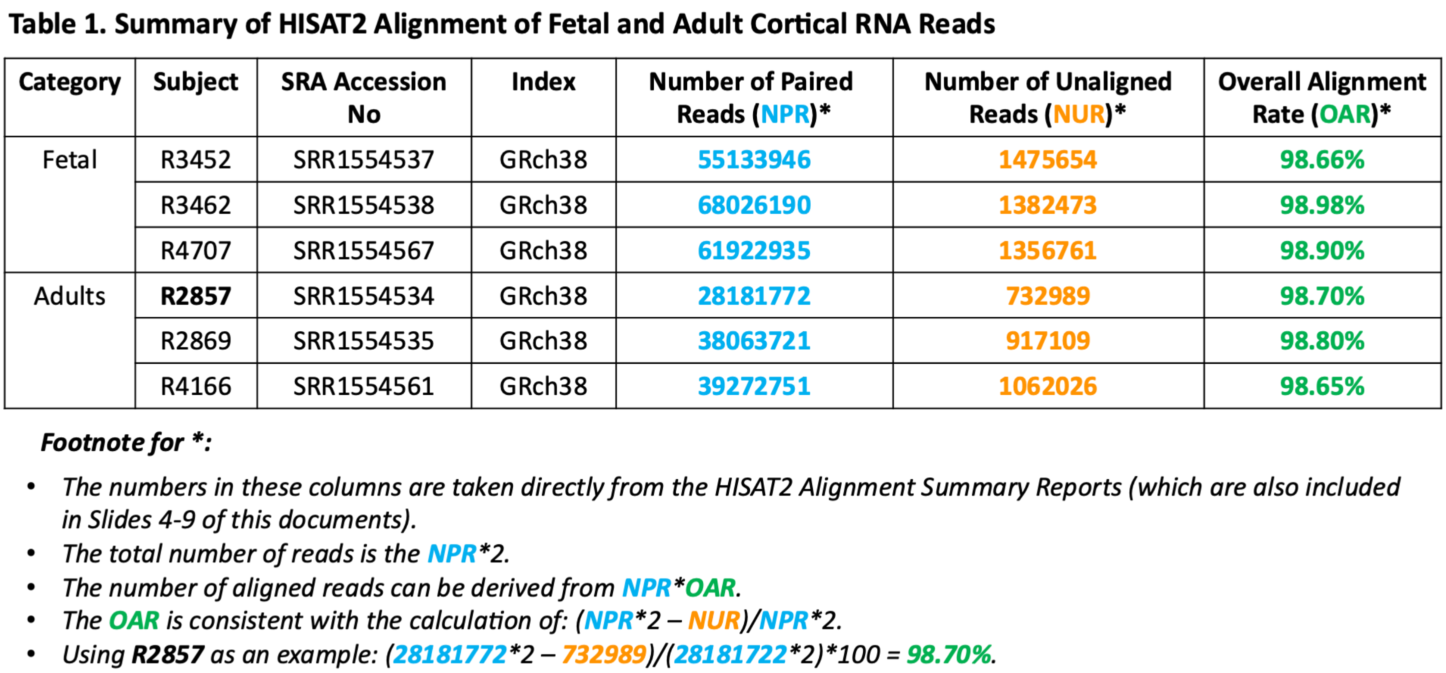


Table 1. Summary of HISAT2 Alignment of Fetal and Adult Cortical RNA Reads

Category	Subject	SRA Accession No	Index	Number of Paired Reads (NPR)*	Number of Unaligned Reads (NUR)*	Overall Alignment Rate (OAR)*
Fetal	R3452	SRR1554537	GRch38	55133946	1475654	98.66%
	R3462	SRR1554538	GRch38	68026190	1382473	98.98%
	R4707	SRR1554567	GRch38	61922935	1356761	98.90%
Adults	**R2857**	SRR1554534	GRch38	28181772	732989	98.70%
	R2869	SRR1554535	GRch38	38063721	917109	98.80%
	R4166	SRR1554561	GRch38	39272751	1062026	98.65%

Footnote for *:

- *The numbers in these columns are taken directly from the HISAT2 Alignment Summary Reports (which are also included in Slides 4-9 of this documents).*
- *The total number of reads is the **NPR***2.*
- *The number of aligned reads can be derived from **NPR******OAR**.*
- *The **OAR** is consistent with the calculation of: (**NPR***2 – **NUR**)/**NPR***2.*
- *Using **R2857** as an example: (**28181772***2 – **732989**)/(**28181722***2)*100 = **98.70%**.*

Week5_Task3: QC

FASTQC mainly does quality checks on reads, but not the alignment. The flagstat function in samtools was used here to obtain the statistics on the alignment which was compared to that from HISAT2.

1. Create a directory (FastQC_Alignment) for the output files from FastQC

```
$ fastqc -o ~/Documents/RNA-seq/FastQC_Alignment -f sam R2857.sam
```

2. FASTQC outputs two files: R2857_fastqc.html and R2857_fastqc.zip

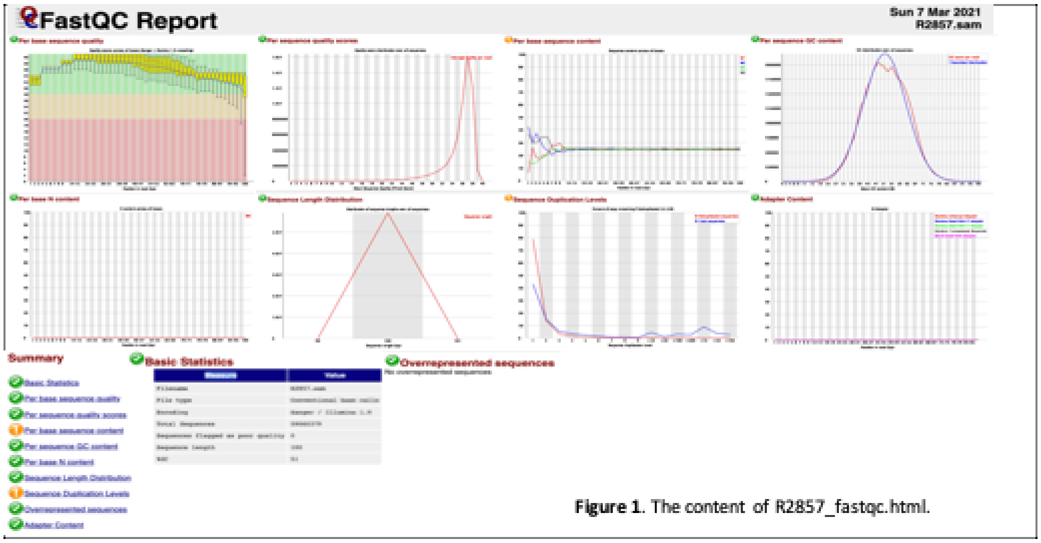
```
# Flagstat takes only .bam file,        samtools view was used to convert .sam to .bam
$ samtools view -S R2857.sam > R2857.bam
```

3. Create a directory (flagstat) for the output files from flagstat

```
$ samtools flagstat R2857.bam > ~/Documents/RNA-seq/flagstat/R2857.bam.flagstat
$ cat flagstat/R2857.bam.flagstat
```

Result

1. The graphic content of the FastQC result is shown in Fig.1.


FastQC Report
Sun 7 Mar 2021
R2857.sam
Per base sequence quality
Per sequence quality scores
Per base sequence content
Per sequence GC content
Per base N content
Sequence Length Distribution
Sequence Duplication Levels
Adapter Content

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per sequence quality scores
- ! Per base sequence content
- ✓ Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ! Sequence Duplication Levels
- ✓ Overrepresented sequences
- ✓ Adapter Content

✓ Basic Statistics

Measure	Value
Filename	R2857.sam
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	[illegible]
Sequences flagged as poor quality	0
Sequence length	100
%GC	51

✓ Overrepresented sequences

No overrepresented sequences

Figure 1. The content of R2857_fastqc.html.

2. FastQC results demonstrate consistent and good qualities of sequence reads from all 6 samples (Table 2). In addition, the following observations can be made about the differences between reads from adult and fetal samples:

- The numbers of reads are in general higher in Fetal than Adult samples (**Table 2**, “Total Sequence”)
- The differences between A versus T or G versus C contents appear greater in Fetal than Adult samples (**Table 2**, “Per base sequence content”)
- c)The GC% appears lower in Fetal than Adult samples (**Table 2**, “per sequence GC content”)
- The degree of enrichment in the library appears lower in Fetal samples which also means that the diversity of the reads is higher in Fetal than Adult samples (**Table 2**, “Sequence Duplication Levels”)

Table 2. Summary of FastQC Results*

Components	**R2857**	**R2869**	**R4166**	**R3452**	**R3462**	**R4707**
Total Sequence	59060379	79163966	81680945	115947927	142865978	129615588
Per base sequence quality: Median Range	33-41	33-40	32-40	29-40	30-40	29-40
Per sequence quality scores: Peak Value	37	36.5	37	37	37	37
Per base sequence content: Region of different A vs T or C vs G content	< 14-15 bp	< 14-15 bp	< 14-15 bp	< 14-15 bp	< 14-15 bp	< 14-15 bp
Per sequence GC content	51%	47%	52%	48%	47%	46%
Per base N content	PASS	PASS	PASS	PASS	PASS	PASS
Sequence Length Distribution	100	100	100	100	100	100
Sequence Duplication Levels: de-duplication percentage	54.52%	51.3%	58.37%	71.12%	69.30%	70.79%
Overrepresented sequences	No	No	No	No	No	No
Adapter Content	PASS	PASS	PASS	PASS	PASS	PASS

**Information in the table are taken from R*_fastqc.html & fastqc_data.txt file. Texts in orange and red correspond to “WARN” and “FAIL” conclusion as determined by FastQC. The texts are not colored if the conclusion is “PASS”. The labels of Adult and Fetal samples are in light blue and yellow backgrounds, respectively*.

3. The mapping rate and quality is very high in all 6 sample and consistent between Fetal and Adults groups. The statistics from HISAT2 and flagstat are in agreement with each other (Table 3).

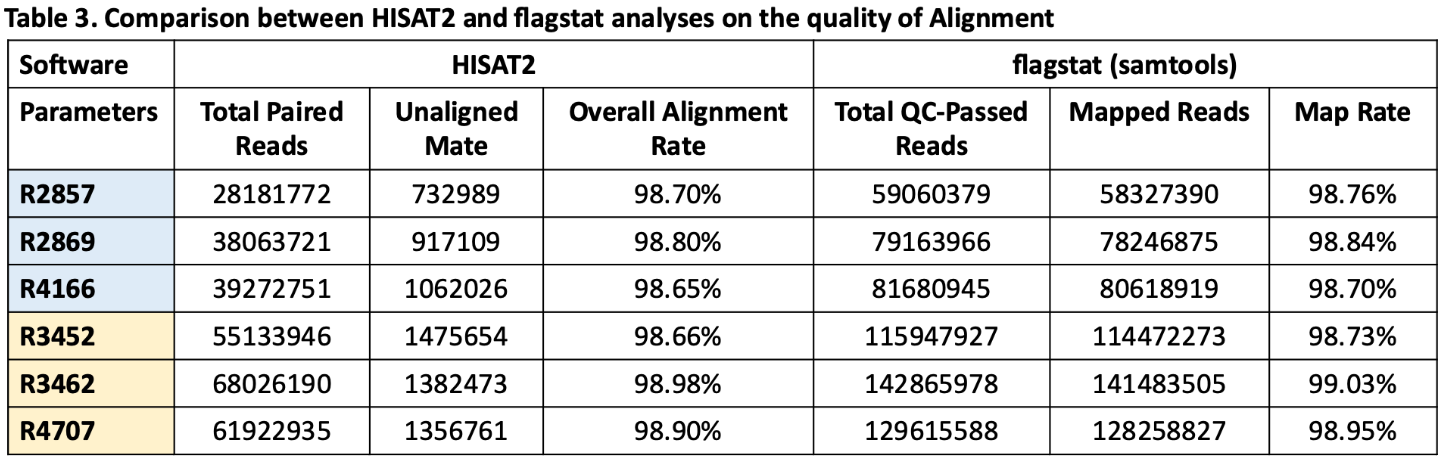


Table 3. Comparison between HISAT2 and flagstat analyses on the quality of Alignment

Software	HISAT2			flagstat (samtools)		
Parameters	Total Paired Reads	Unaligned Mate	Overall Alignment Rate	Total QC-Passed Reads	Mapped Reads	Map Rate
R2857	28181772	732989	98.70%	59060379	58327390	98.76%
R2869	38063721	917109	98.80%	79163966	78246875	98.84%
R4166	39272751	1062026	98.65%	81680945	80618919	98.70%
R3452	55133946	1475654	98.66%	115947927	114472273	98.73%
R3462	68026190	1382473	98.98%	142865978	141483505	99.03%
R4707	61922935	1356761	98.90%	129615588	128258827	98.95%

Week6_Task4: Assemble & Gene Count Feature

1. Sort and convert .sam to .bam

```
$ samtools sort -@ 8 -o R2857_SRR1554534.bam R2857.sam
```

2. Assemble transcripts using the reference annotation file for each sample:

```
$ stringtie -p 8 -G Homo_sapiens.GRCh38.103.gtf -o R2857_SRR1554534.gtf R2857_SRR1554534.bam
```

3. Generate a file containing the list of 6 assembled .gtf files named "mergelist.txt". Merge assembled transcripts from each sample:

```
$ stringtie --merge -p 8 -G Homo_sapiens.GRCh38.103.gtf -o stringtie_merged.gtf mergelist.txt
```

(The following steps are to obtain gene count feature using ballgown downstream of Stringtie)

4. Generate ballgown count table:

```
$ stringtie -e -B -p 8 -G stringtie_merged.gtf -o ballgown/R2857/R2857_SRR1554534.gtf
R2857_SRR1554534.bam
```

5. Upload phenotype data in R

```
> pheno_data = read.csv(“phenotype.csv”)
```

6. Create a ballgown object from which the gene count feature can be obtained

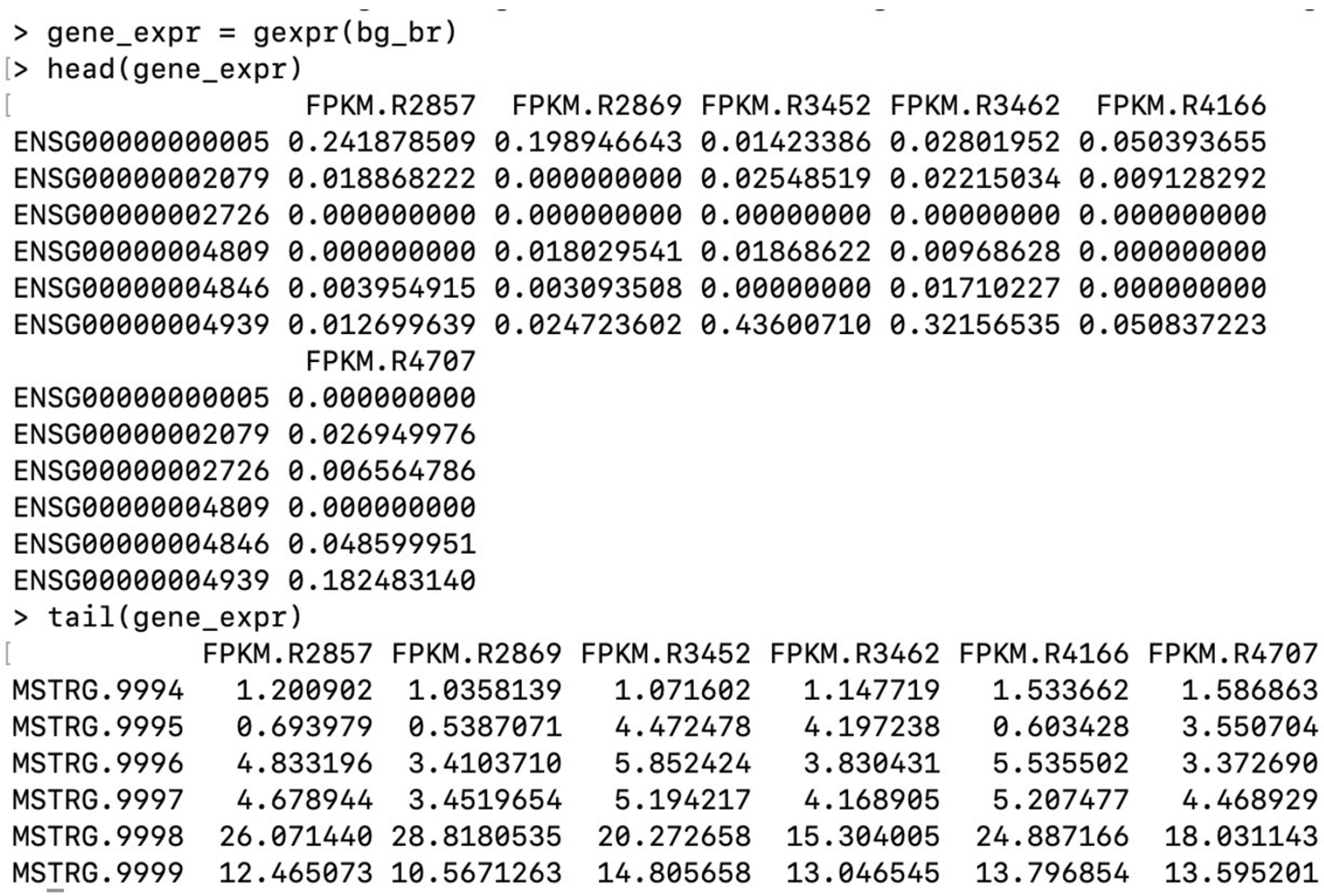
```
> bg_br = ballgown(dataDir="ballgown", samplePattern="R", pData=pheno_data)
> save(bg_br, file=‘bg_br.rda’)
> gene_expr = gexpr(bg_br)
```

7. Obtain the number of expressed genes for each sample

```
$ cat gene_expr |cut -f2|grep -v "0"|wc -l
    3513
$ cat gene_expr |cut -f3|grep -v "0"|wc -l
    3834
$ cat gene_expr |cut -f4|grep -v "0"|wc -l
    4133
$ cat gene_expr |cut -f5|grep -v "0"|wc -l
    4115
$ cat gene_expr |cut -f6|grep -v "0"|wc -l
    3720
$ cat gene_expr |cut -f7|grep -v "0"|wc -l
    4198
```

Result

1. Gene Expression Table was obtained (Fig.2)



```
> gene_expr = gexpr(bg_br)
> head(gene_expr)
                 FPKM.R2857  FPKM.R2869 FPKM.R3452 FPKM.R3462  FPKM.R4166
ENSG00000000005 0.241878509 0.198946643 0.01423386 0.02801952 0.050393655
ENSG00000002079 0.018868222 0.000000000 0.02548519 0.02215034 0.009128292
ENSG00000002726 0.000000000 0.000000000 0.00000000 0.00000000 0.000000000
ENSG00000004809 0.000000000 0.018029541 0.01868622 0.00968628 0.000000000
ENSG00000004846 0.003954915 0.003093508 0.00000000 0.01710227 0.000000000
ENSG00000004939 0.012699639 0.024723602 0.43600710 0.32156535 0.050837223
                 FPKM.R4707
ENSG00000000005 0.000000000
ENSG00000002079 0.026949976
ENSG00000002726 0.006564786
ENSG00000004809 0.000000000
ENSG00000004846 0.048599951
ENSG00000004939 0.182483140
> tail(gene_expr)
           FPKM.R2857 FPKM.R2869 FPKM.R3452 FPKM.R3462 FPKM.R4166 FPKM.R4707
MSTRG.9994   1.200902  1.0358139   1.071602   1.147719   1.533662   1.586863
MSTRG.9995   0.693979  0.5387071   4.472478   4.197238   0.603428   3.550704
MSTRG.9996   4.833196  3.4103710   5.852424   3.830431   5.535502   3.372690
MSTRG.9997   4.678944  3.4519654   5.194217   4.168905   5.207477   4.468929
MSTRG.9998  26.071440 28.8180535  20.272658  15.304005  24.887166  18.031143
MSTRG.9999  12.465073 10.5671263  14.805658  13.046545  13.796854  13.595201
```

Figure 2. Screenshot image of the head and tail part of the tab-delimited table containing gene IDs and expression level in FPKM for all samples..

2. The number of genes expressed in Fetal brain is signigicantly higher than that in Adult brain (Fig. 3)

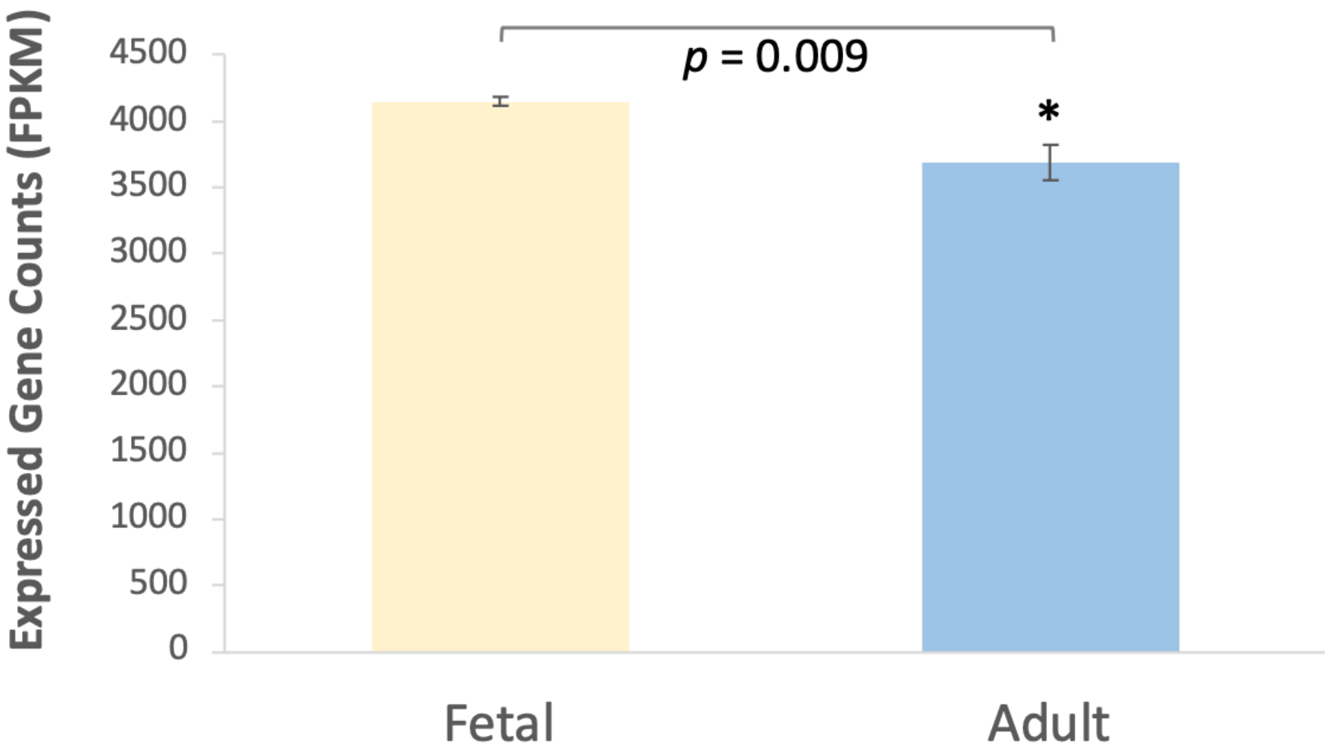
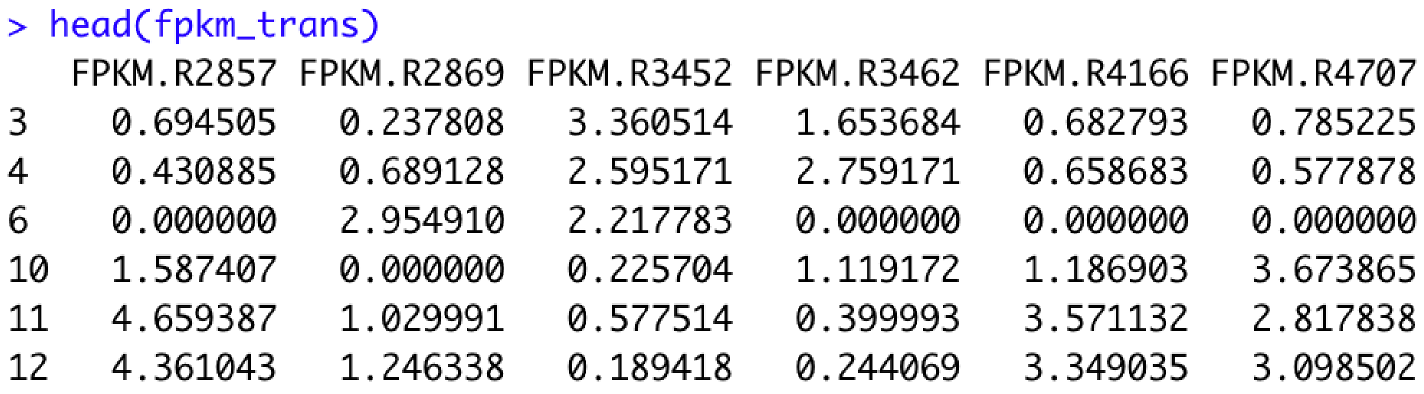


Figure 3. Comparison of expressed genes counts between Fetal and Adult groups. *p* value was from T.Test. Error bars: Mean ± SD.

Week7_Task5: Exploratory Analysis

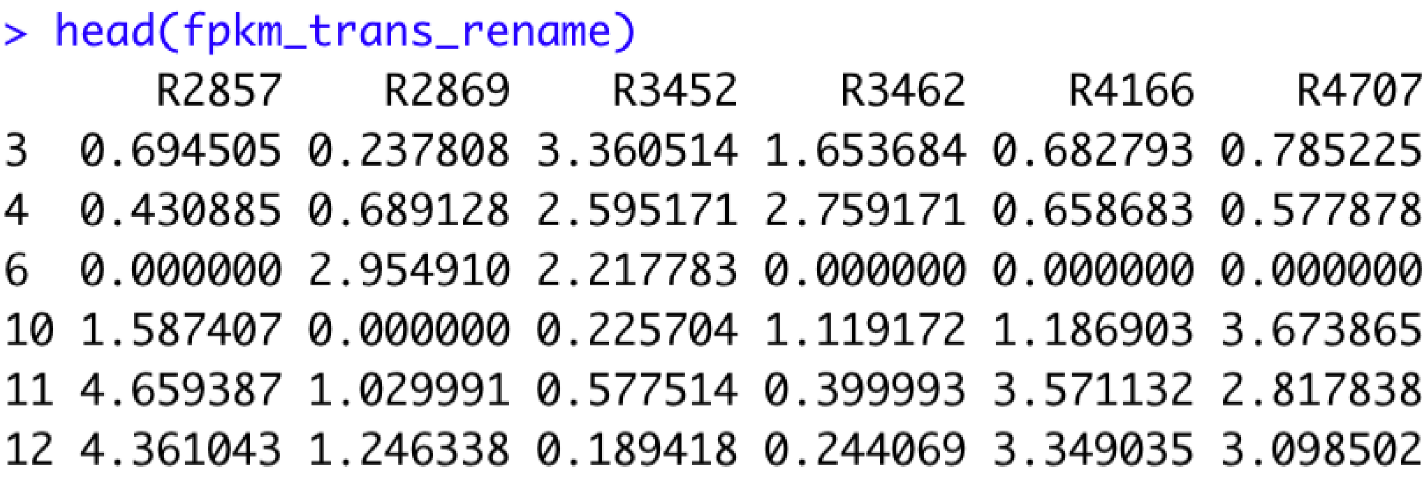
1. Visualize trnascript expression for all samples

```
> library(genefilter)
> library(tidyverse)
> library(Biobase)
> library(devtools)
> library(cowplot)

> bg_br_filt = subset(bg_br, "rowVars(texpr(bg_br)) > 1", genomesubsete=TRUE))
> fpkm_trans = texpr(bg_br_filt) # This is matrix file
> fpkm_gene = gexpr(bg_br_filt) # This is to be used for PCA analysis later
> head(fpkm_trans)
```



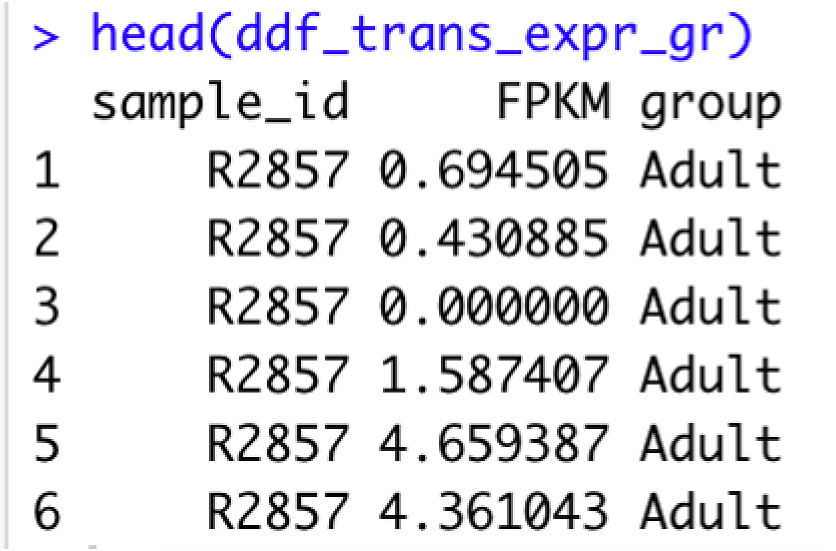
```
> head(fpkm_trans)
   FPKM.R2857 FPKM.R2869 FPKM.R3452 FPKM.R3462 FPKM.R4166 FPKM.R4707
3    0.694505   0.237808   3.360514   1.653684   0.682793   0.785225
4    0.430885   0.689128   2.595171   2.759171   0.658683   0.577878
6    0.000000   2.954910   2.217783   0.000000   0.000000   0.000000
10   1.587407   0.000000   0.225704   1.119172   1.186903   3.673865
11   4.659387   1.029991   0.577514   0.399993   3.571132   2.817838
12   4.361043   1.246338   0.189418   0.244069   3.349035   3.098502
```

```
> fpkm_trans_rename <- as.data.frame(fpkm_trans) %>%
rename(R2857=FPKM.R2857,R2869=FPKM.R2869, R3452=FPKM.R3452, R3462=FPKM.R3462,
R4166=FPKM.R4166, R4707=FPKM.R4707)
> head(fpkm_trans_rename) # This is data.frame file
```



```
> head(fpkm_trans_rename)
      R2857    R2869    R3452    R3462    R4166    R4707
3  0.694505 0.237808 3.360514 1.653684 0.682793 0.785225
4  0.430885 0.689128 2.595171 2.759171 0.658683 0.577878
6  0.000000 2.954910 2.217783 0.000000 0.000000 0.000000
10 1.587407 0.000000 0.225704 1.119172 1.186903 3.673865
11 4.659387 1.029991 0.577514 0.399993 3.571132 2.817838
12 4.361043 1.246338 0.189418 0.244069 3.349035 3.098502
```

```
> ddf_trans_expr <- fpkm_trans_rename %>% gather(sample_id, FPKM) --> tify data.frame
for ggplot

# The next few steps are to add group phenotype information to this table)
> ddf_trans_expr$gr = ddf_trans_expr$sample_id # duplicate the sample_id column
> gr <- c(R2857 ="Adult", R2869="Adult", R3452="Fetal", R3462="Fetal", R4166="Adult",
R4707="Fetal") # provide the information for replacement
> ddf_trans_expr$group <- as.character(gr[ddf_trans_expr$group]) # replace the
characters with group information
> ddf_trans_expr_gr <- subset(ddf_trans_expr, select = -c(gr)) # remove the unwanted
"gr" column
> head(ddf_trans_expr_gr)
```



```
> head(ddf_trans_expr_gr)
  sample_id     FPKM group
1     R2857 0.694505 Adult
2     R2857 0.430885 Adult
3     R2857 0.000000 Adult
4     R2857 1.587407 Adult
5     R2857 4.659387 Adult
6     R2857 4.361043 Adult
```

```
> p1 <- ddf_trans_expr_gr %>% ggplot(aes(x=sample_id, y=FPKM)) + geom_boxplot()
> p2 <- dddf_trans_expr_gr %>% ggplot(aes(x=sample, y=log2(FPKM+1))) + geom_boxplot()
> plot_grid(p1, p2, labels = c('A', 'B'))
```

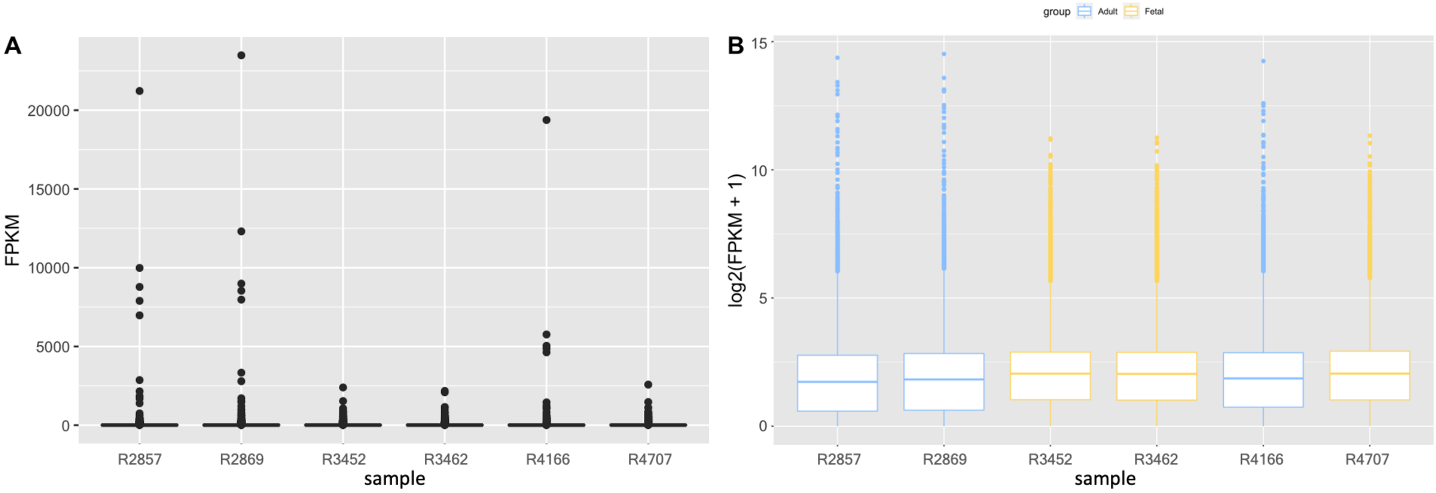
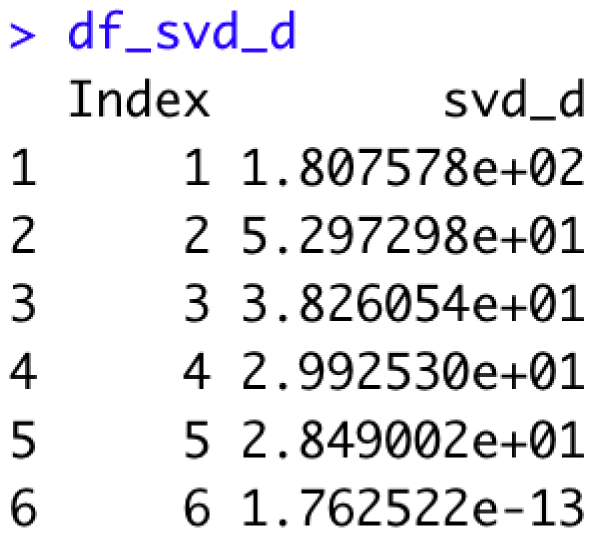


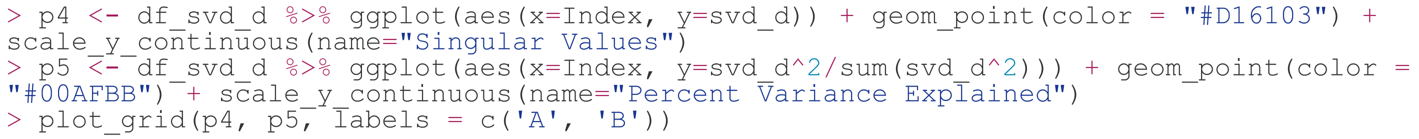
Figure 4. FPKM distributions for gene expression in all samples with un-transformed (A) and log2 transformed (B) values. Adult and Fetal brains are colored in blue and light orange (B), respectively.

2. Perform PCA

```
> edata = as.data.frame(fpkm_gene)
> pdata = pheno_data
> edata_filt = edata[rowMeans(edata) > 1, ]
> edata_filt_log = log2(edata_filt + 1)
> edata_filt_log_centered = edata_filt_log - rowMeans(edata_filt_log)
> svd1 = svd(edata_filt_log_centered)
> svd1 %>% names %>% head
[1] ”d” “u” “v”
> svd1$d
[1] 1.807578e+02 5.297298e+01 3.826054e+01 2.992530e+01 2.849002e+01 1.762522e-13
> df_svd_d = data.frame(Index=c(“1”, “2”, “3”, “4”, “5”, “6”), svd_d= c(1.807578e+02,
5.297298e+01, 3.826054e+01, 2.992530e+01, 2.849002e+01, 1.762522e-13))
> df_svd_d
```



```
> df_svd_d
  Index        svd_d
1     1 1.807578e+02
2     2 5.297298e+01
3     3 3.826054e+01
4     4 2.992530e+01
5     5 2.849002e+01
6     6 1.762522e-13
```



```
> p4 <- df_svd_d %>% ggplot(aes(x=Index, y=svd_d)) + geom_point(color = "#D16103") +
scale_y_continuous(name="Singular Values")
> p5 <- df_svd_d %>% ggplot(aes(x=Index, y=svd_d^2/sum(svd_d^2))) + geom_point(color =
"#00AFBB") + scale_y_continuous(name="Percent Variance Explained")
> plot_grid(p4, p5, labels = c('A', 'B'))
```

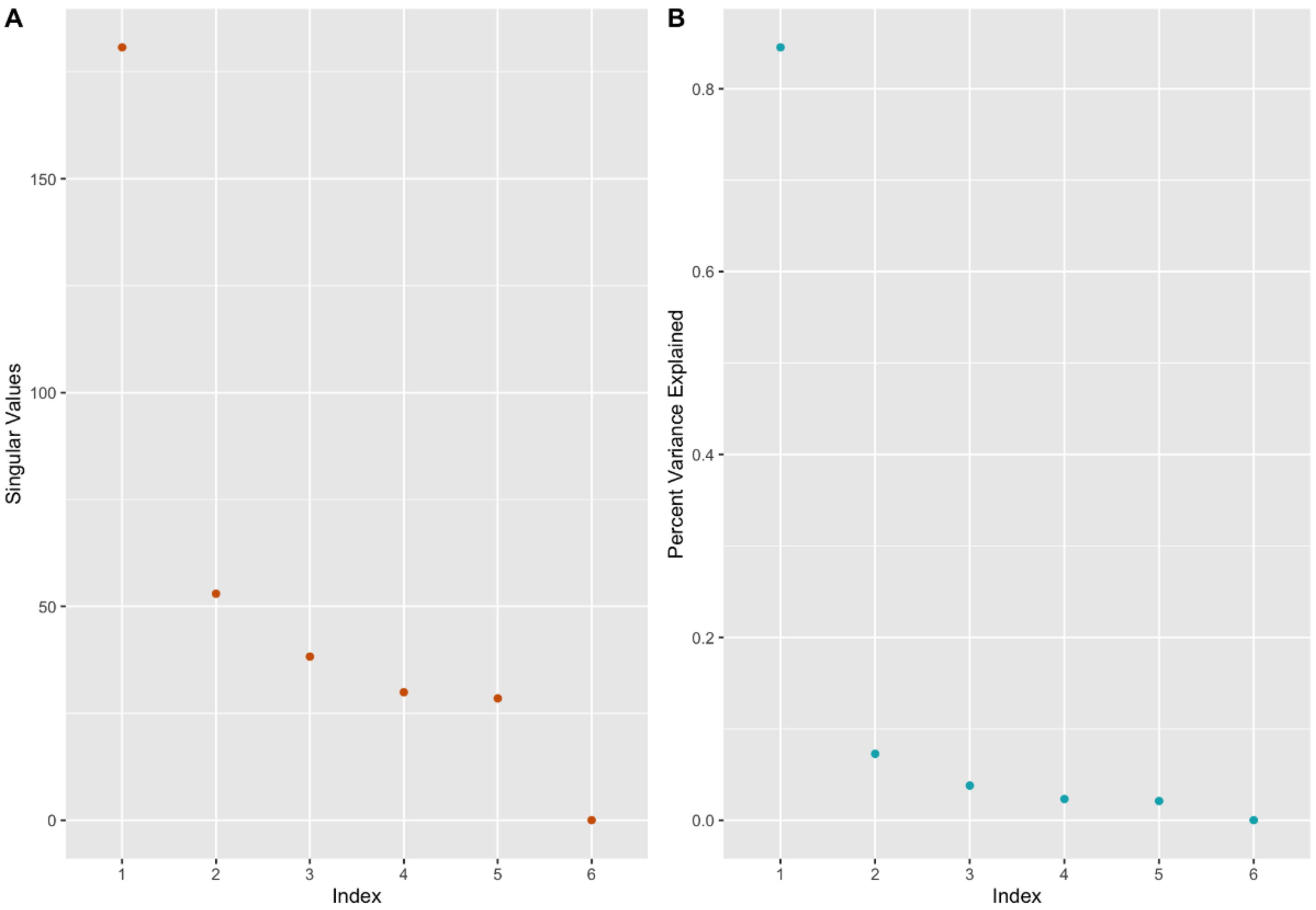
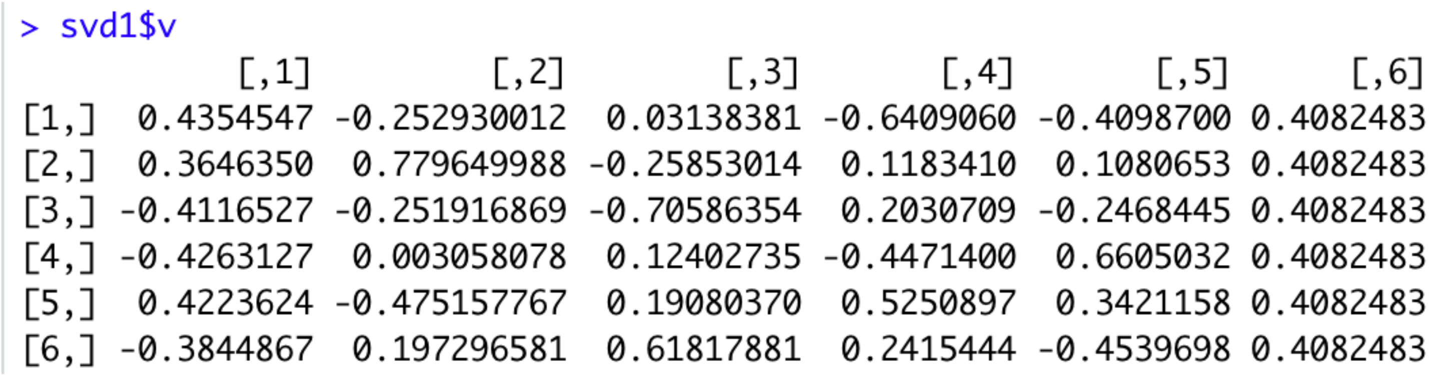


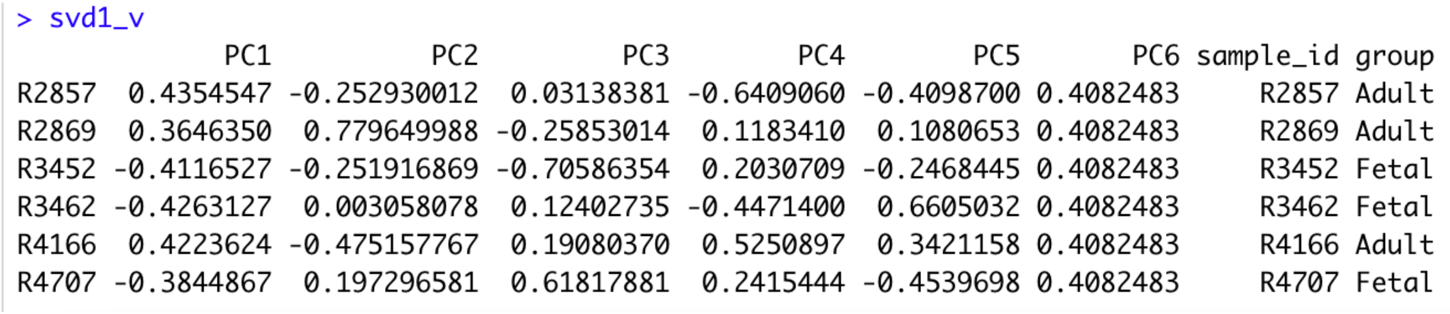
Figure 5. Scatter plot of PCA (using Single Value Decomposition method) explaining variances in gene expression.

```
> svd1$v
```



```
> svd1$v
           [,1]         [,2]        [,3]       [,4]       [,5]      [,6]
[1,]  0.4354547 -0.252930012  0.03138381 -0.6409060 -0.4098700 0.4082483
[2,]  0.3646350  0.779649988 -0.25853014  0.1183410  0.1080653 0.4082483
[3,] -0.4116527 -0.251916869 -0.70586354  0.2030709 -0.2468445 0.4082483
[4,] -0.4263127  0.003058078  0.12402735 -0.4471400  0.6605032 0.4082483
[5,]  0.4223624 -0.475157767  0.19080370  0.5250897  0.3421158 0.4082483
[6,] -0.3844867  0.197296581  0.61817881  0.2415444 -0.4539698 0.4082483
```

```
# Tidy svd1$v for ggplot
> colnames(svd1$v) = c("PC1", "PC2", "PC3", "PC4", "PC5", "PC6")
> rownames(svd1$v) = c("R2857", "R2869", "R3452", "R3462", "R4166", "R4707")
> svd1_v = data.frame(svd1$v)
> svd1_v$sample_id = rownames(svd1_v)
> svd1_v$group = c("Adult", "Adult", "Fetal", "Fetal", "Adult", "Fetal")
> svd1_v
```



```
> svd1_v
             PC1          PC2         PC3        PC4        PC5       PC6 sample_id group
R2857  0.4354547 -0.252930012  0.03138381 -0.6409060 -0.4098700 0.4082483     R2857 Adult
R2869  0.3646350  0.779649988 -0.25853014  0.1183410  0.1080653 0.4082483     R2869 Adult
R3452 -0.4116527 -0.251916869 -0.70586354  0.2030709 -0.2468445 0.4082483     R3452 Fetal
R3462 -0.4263127  0.003058078  0.12402735 -0.4471400  0.6605032 0.4082483     R3462 Fetal
R4166  0.4223624 -0.475157767  0.19080370  0.5250897  0.3421158 0.4082483     R4166 Adult
R4707 -0.3844867  0.197296581  0.61817881  0.2415444 -0.4539698 0.4082483     R4707 Fetal
```

```
> p6_bw <- svd1_v %>% ggplot(aes(x=sample_id, y=PC1)) + geom_point(aes(color=group)) +
scale_color_manual(values = c("#99CCFF", "#E69F00")) + theme_bw()
> p7_bw <- svd1_v %>% ggplot(aes(x=sample_id, y=PC2)) + geom_point(aes(color=group)) +
scale_color_manual(values = c("#99CCFF", "#E69F00")) + theme_bw()
> p8_bw <- svd1_v %>% ggplot(aes(x=PC1, y=PC2)) + geom_point(aes(color=group)) +
scale_color_manual(values = c("#99CCFF", "#E69F00")) + theme_bw()
> plot_grid(p6_bw, p7_bw, p8_bw, labels=c("A", "B", "C"), ncol=3, nrow=3)
```

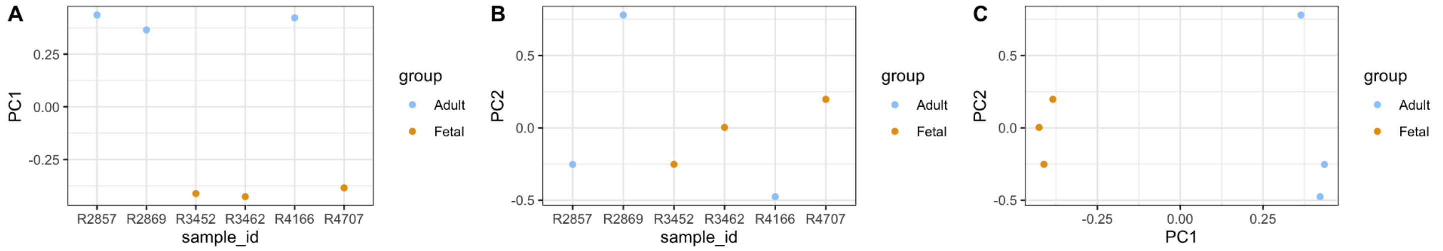


Figure 6. Sample distributions for the 1st (A) and 2nd (B) principal components (PC1 and PC2, respectively as well as the associations between PC1 and PC2 for each sample (C).

Result

1. Log2 transformation of transcript expression data outputs more meaningful visualization in boxplot (Fig.4)

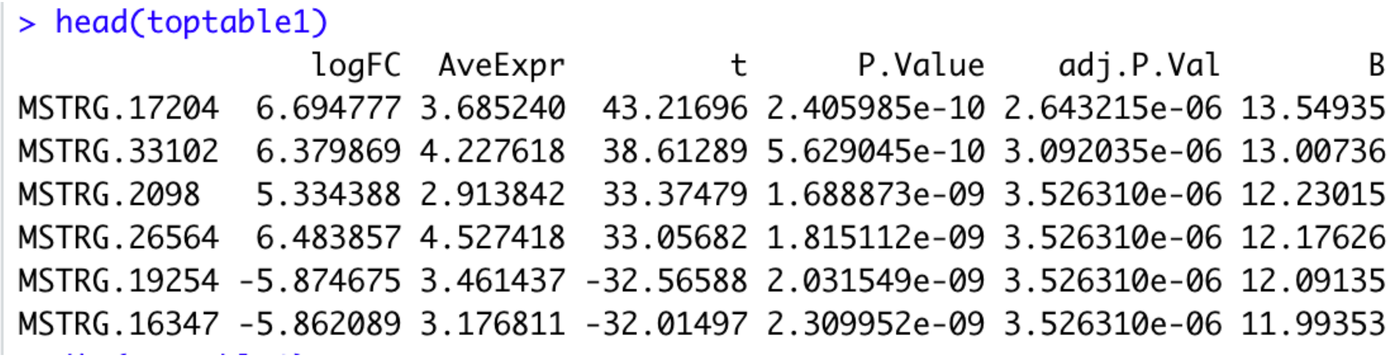
2. PCA demonstrates that close to 90% of variances in gene expression is accounted for by PC1 and PC2 (Fig.5)

3. Gene expression is highly correlated to the group type (i.e. Fetal vs. Adult) in PC1 but not in PC2 (Fig.6)

Week8_Task6: Statistical Analysis

1. Generate differential gene expression table with adjusted *p* values

```
> library(broom)
> library(genefilter)
> library(tidyverse)
> library(limma)
> library(Biobase)

> edata = read.table(“gene_expr.txt”, sep=“\t”)
# generated in Wk6_Task4
> pdata = read.table("phenotype_txt", sept="\t")
> edata <- edata %>% rename(R2857=FPKM.R2857,R2869=FPKM.R2869, R3452=FPKM.R3452,
R3462=FPKM.R3462, R4166=FPKM.R4166, R4707=FPKM.R4707)
> write.table(edata, "edata.txt", sep = "\t", append = FALSE, row.names = TRUE,
col.names = TRUE)

> edata_ffilt = edata[rowMeans(edata) > 100, ]
> edata_log = log2(edata+1)
> edata_ffilt_log = log2(edata_ffilt + 1)
> mod = model.matrix(~pdata$group)
> fit1=lmFit(edata_log, mod)
> fit2=lmFit(edata_ffilt_log, mod)
> ebayes_fit1 = eBayes(fit1)
> ebayes_fit2 = eBayes(fit2)
> toptable1 = topTable(ebayes_fit1, number=dim(edata_log)[1], sort.by="p", coef=2)
> toptable2 = topTable(ebayes_fit2, number=dim(edata_ffilt_log)[1], sort.by="p",
coef=2)
> write.table(toptable1, "diff_expr.txt", sep = "\t", append = FALSE, row.names =
TRUE, col.names = TRUE)
> write.table(toptable2, "diff_expr_filt.txt", sep = "\t", append = FALSE, row.names =
TRUE, col.names = TRUE)

# Two tables have been generated without and with filtering gene counts, which will be
used for volcano plot.
> dim(toptable1)
[1] 10986     6
> dim(toptable2)
[1] 177   6
> toptable1_q0.05 <- subset(toptable1, toptable1$adj.P.Val < 0.05)
> dim(toptable1_q0.05)
[1] 7674    6
> toptable2_q0.05 <- subset(toptable2, toptable2$adj.P.Val < 0.05)
> dim(toptable2_q0.05)
[1] 143 6
> head(toptable1)
```



```
> head(toptable1)
                 logFC  AveExpr         t      P.Value    adj.P.Val        B
MSTRG.17204   6.694777 3.685240  43.21696 2.405985e-10 2.643215e-06 13.54935
MSTRG.33102   6.379869 4.227618  38.61289 5.629045e-10 3.092035e-06 13.00736
MSTRG.2098    5.334388 2.913842  33.37479 1.688873e-09 3.526310e-06 12.23015
MSTRG.26564   6.483857 4.527418  33.05682 1.815112e-09 3.526310e-06 12.17626
MSTRG.19254  -5.874675 3.461437 -32.56588 2.031549e-09 3.526310e-06 12.09135
MSTRG.16347  -5.862089 3.176811 -32.01497 2.309952e-09 3.526310e-06 11.99353
```

2. Generate Volcano plots for differential gene expression between Fetal and Adult brains using ggplot

```
> toptable1$gene_id <- rownames(toptable1)
> toptable1 %>% ggplot(aes(x=logFC, y=-log10(P.Value), color=adj.P.Val < 0.05)) +
geom_point(size=0.8) # Fig.8a_A
> toptable2 %>% ggplot(aes(x=logFC, y=-log10(P.Value), color=adj.P.Val < 0.05)) +
geom_point(size=0.8) # Fig.8a_B

# To try out the cool things in ggplot, a different visualization of the Volcano plot
colored in the possible combinatorial conditions determined by two parameters: -
log(adj.P.Val) < 0.05 and |logFC| > 1.0 was also generated using the un-filted
dataset.
> toptable$clr1 <- paste(toptable1$logFC > 1.0, toptable1$adj.P.Val < 0.05)
> toptable$clr2 <- paste(toptable1$logFC < -1.0, toptable1$adj.P.Val < 0.05)
> > toptable1 %>% ggplot(aes(x=logFC, y=-log10(P.Value), color=clr1)) +geom_point()
> toptable1 %>% ggplot(aes(x=logFC, y=-log10(P.Value), color=clr2)) +geom_point()
# Two plots were superimposed on each other to generate Fig.8b.
```

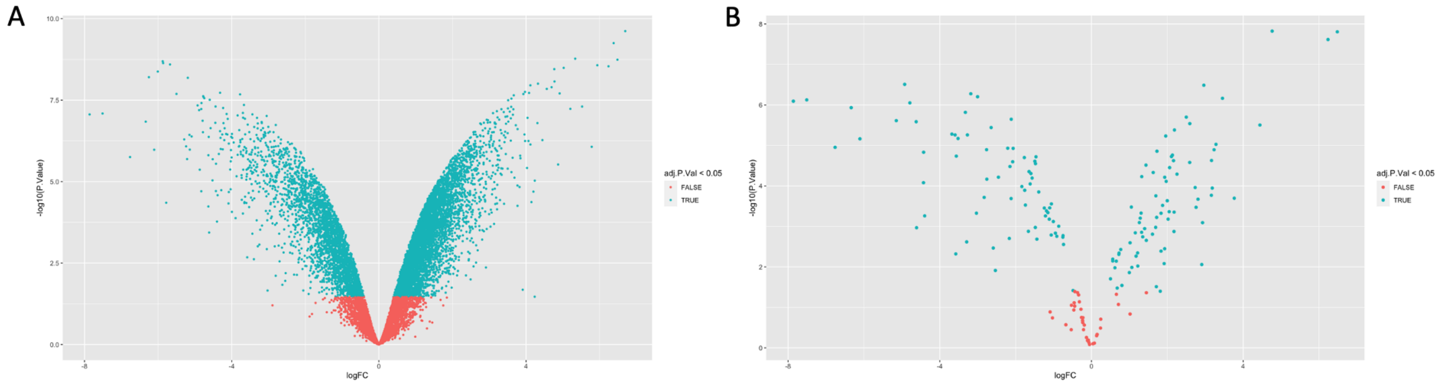


Figure 8a. The Volcano Plot for differential gene expression between Fetal and Adult brains using unfiltered (A) and filtered (B) dataset. Genes with adjusted p value < 0.05 are labeled with a different color.

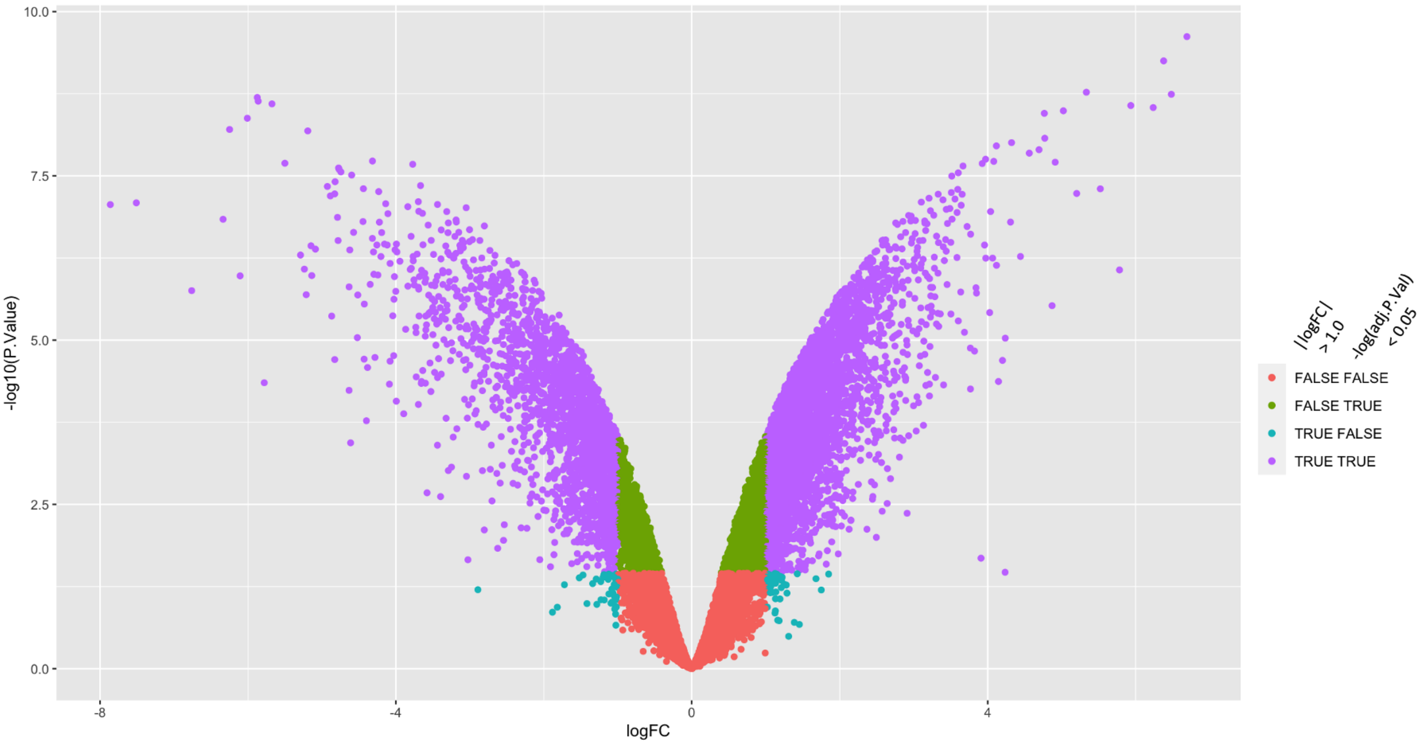
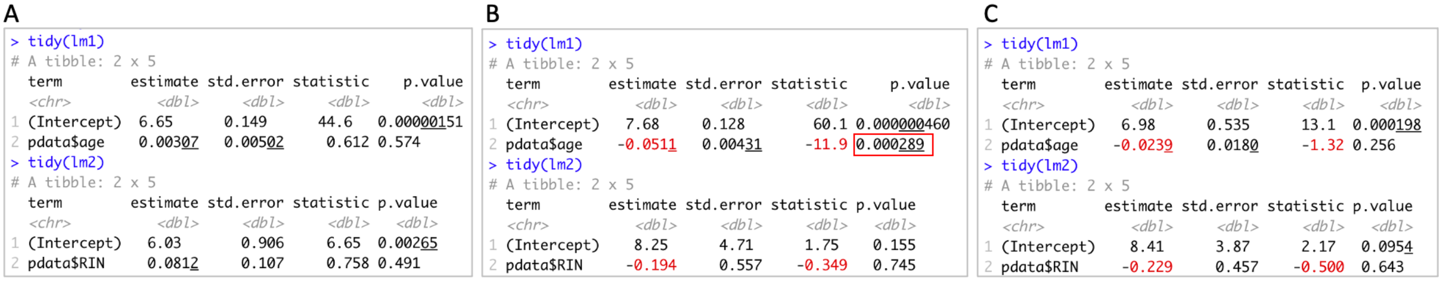


Figure 8b. The Volcano Plot for differential gene expression between Fetal and Adult brains using unfiltered dataset. Genes with adjusted *p* value < 0.05 and/or |logFC| > 1.0 are labeled with different colors.

3. Linear regression analysis of gene expression in relation to either age or RIN

```
> edata_ffilt = edata[rowMeans(edata) > 100, ]
> edata_ffilt_AsMatrixLog = log2(as.matrix(edata_ffilt) + 1)
> lm1 = lm(edata_filt_AsMatrixLog[1, ] ~ pdata$age)
> lm2 = lm(edata_filt_AsMatrixLog[1, ] ~ pdata$RIN)
> tidy(lm1)
> tidy(lm2)
# The same process was repeated for the 2nd and 3rd genes. The output of the
statistics for the three genes is shown in Fig.9.
```



A

```
> tidy(lm1)
# A tibble: 2 x 5
  term        estimate std.error statistic    p.value
  <chr>          <dbl>     <dbl>     <dbl>      <dbl>
1 (Intercept)  6.65      0.149      44.6   0.00000151
2 pdata$age    0.00307   0.00502     0.612 0.574
> tidy(lm2)
# A tibble: 2 x 5
  term        estimate std.error statistic p.value
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)   6.03       0.906     6.65  0.00265
2 pdata$RIN     0.0812     0.107     0.758 0.491
```

B

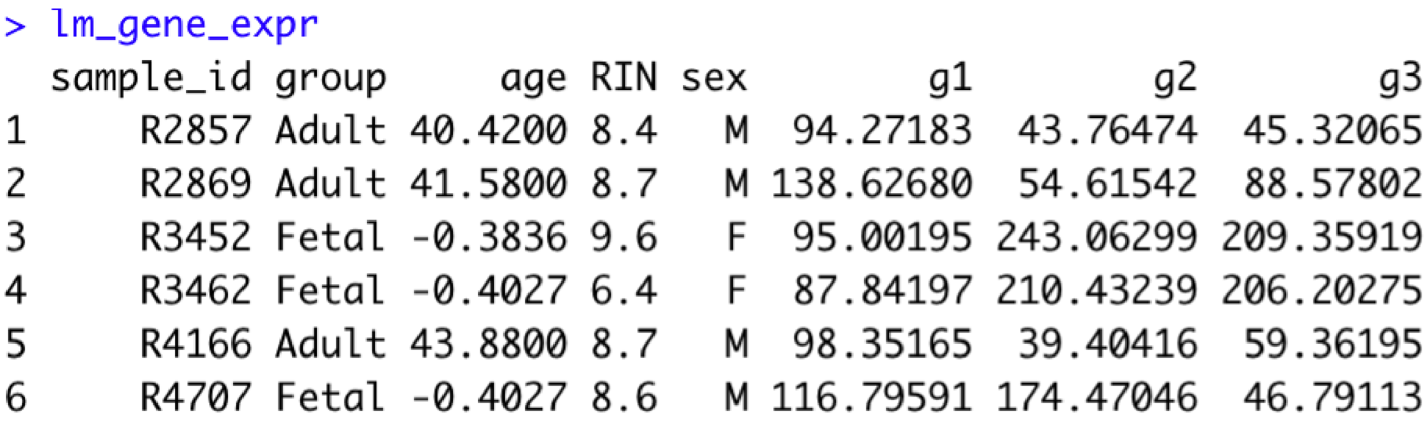
```
> tidy(lm1)
# A tibble: 2 x 5
  term        estimate std.error statistic     p.value
  <chr>          <dbl>     <dbl>     <dbl>       <dbl>
1 (Intercept)   7.68     0.128        60.1 0.000000460
2 pdata$age    -0.0511   0.00431     -11.9 0.000289
> tidy(lm2)
# A tibble: 2 x 5
  term        estimate std.error statistic p.value
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)    8.25      4.71      1.75    0.155
2 pdata$RIN     -0.194     0.557    -0.349   0.745
```

C

```
> tidy(lm1)
# A tibble: 2 x 5
  term        estimate std.error statistic  p.value
  <chr>          <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept)   6.98      0.535      13.1  0.000198
2 pdata$age    -0.0239    0.0180     -1.32 0.256
> tidy(lm2)
# A tibble: 2 x 5
  term        estimate std.error statistic p.value
  <chr>          <dbl>     <dbl>     <dbl>   <dbl>
1 (Intercept)    8.41      3.87      2.17   0.0954
2 pdata$RIN     -0.229     0.457    -0.500  0.643
```

Figure 9. The statistic output of linear regression analysis for an individual gene (the first three genes as (A), (B), and (C), respectively) in relation to either age or RIN values. Note that the only significant p value is for the 2nd gene in relation to age (B, indicated by a red rectangle).

```
# Visulization of the gene expression pattern was also generated (Fig. 10)
> lm_gene_expr = t(edata_ffilt[1:3, ])
> lm_gene_expr$sample_id = rownames(lm_gene_expr)
> lm_gene_expr = merge(pdata, lm_gene_expr, by="sample_id")
> lm_gene_expr = rename(lm_gene_expr, g1=MSTRG.10464, g2=MSTRG.10768, g3=MSTRG.11028)
> lm_gene_expr
```



```
> lm_gene_expr
  sample_id group      age RIN sex        g1        g2        g3
1     R2857 Adult 40.4200 8.4    M  94.27183  43.76474  45.32065
2     R2869 Adult 41.5800 8.7    M 138.62680  54.61542  88.57802
3     R3452 Fetal -0.3836 9.6    F  95.00195 243.06299 209.35919
4     R3462 Fetal -0.4027 6.4    F  87.84197 210.43239 206.20275
5     R4166 Adult 43.8800 8.7    M  98.35165  39.40416  59.36195
6     R4707 Fetal -0.4027 8.6    M 116.79591 174.47046  46.79113
```

```
> a1_log <- lm_gene_expr %>% ggplot(aes(x=age, y=log2(g1+1), color=group)) +
geom_point() + geom_smooth(method=lm, se=FALSE,color="#FFCC99")
> a2_log <- lm_gene_expr %>% ggplot(aes(x=age, y=log2(g2+1), color=group)) +
geom_point() + geom_smooth(method=lm, se=FALSE,color="#FFCC99")
> a3_log <- lm_gene_expr %>% ggplot(aes(x=age, y=log2(g3+1), color=group)) +
geom_point() + geom_smooth(method=lm, se=FALSE,color="#FFCC99")
> r3_log <- lm_gene_expr %>% ggplot(aes(x=RIN, y=log2(g3+1), color=group)) +
geom_point() + geom_smooth(method=lm, se=FALSE,color="#FFCC99")
> r2_log <- lm_gene_expr %>% ggplot(aes(x=RIN, y=log2(g2+1), color=group)) +
geom_point() + geom_smooth(method=lm, se=FALSE,color="#FFCC99")
> r1_log <- lm_gene_expr %>% ggplot(aes(x=RIN, y=log2(g1+1), color=group)) +
geom_point() + geom_smooth(method=lm, se=FALSE,color="#FFCC99")
> plot_grid(a1_log,a2_log,a3_log,r1_log,r2_log,r3_log, nrow=2, ncol=3) #Fig. 10
```

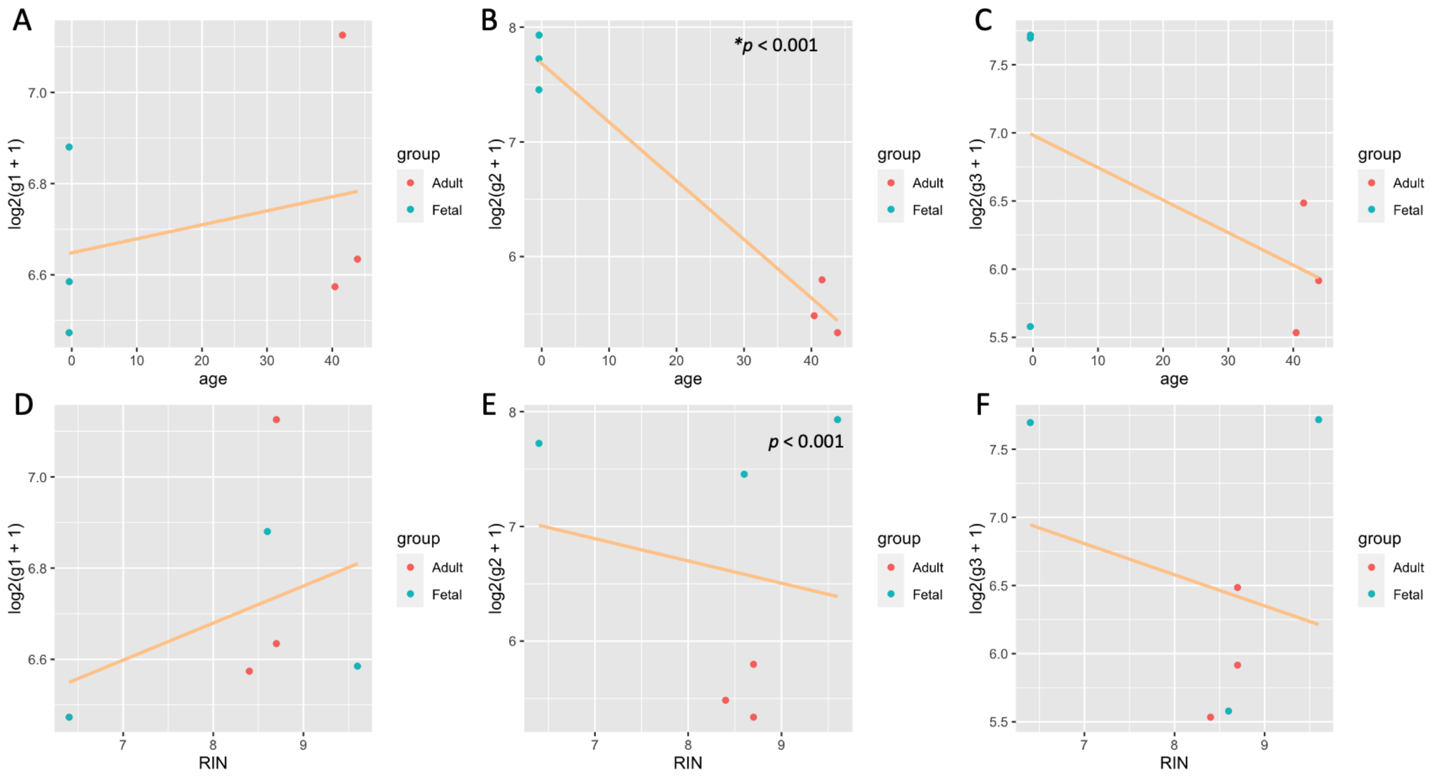


Figure 10. Visualization of the liner regression model of gene expression patterns in relation to the factors of age (A-C) and RIN (D-F) for Fetal and Adult samples. The only significant p value was indicated for g2 & age in (B).

Result

1. The number for differentially expressed genes between Fetal and Adult brains is 7674 and 143, using un-filtered and filtered dataset, respectively. The significance level is determined by the adjusted *p* value < 0.05.

2. The Volcano Plot shows the degree of downregulation is larger than that of upregulation (Fig.8a, further left-end dots), while there is a higer degree of significance for upregulated genes (Fig.8a, higher right-side dots).

3. With the nature of age data (not continuous but clustered into 2 groups), the linear regression model is not expected to be suitable for assessing the pattern of gene expression in relation to age (Fig.9), even though the p value for g2 is signicant (Fig.10B). With the limited number of samples, there appear to be no significant association between gene expression level and RIN (Fig. 9).

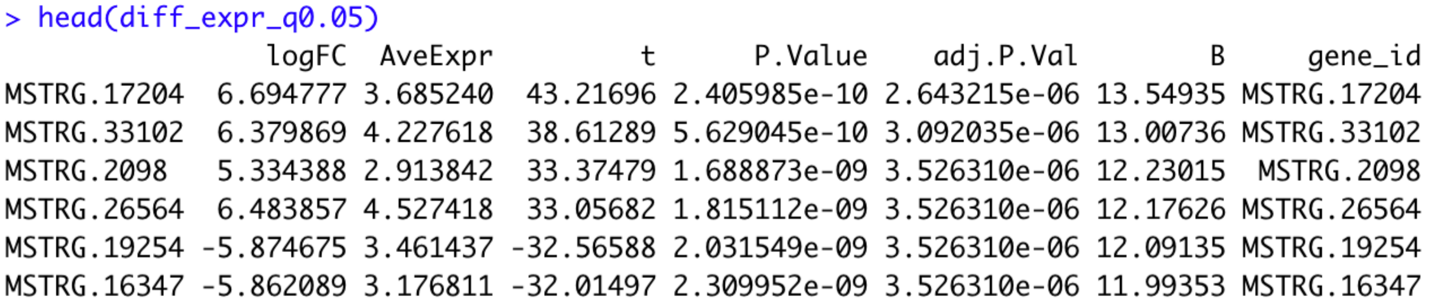
Week9_Task7: Gene Set Analysis

1. The subset of differentially expressed genes (i.e. *adj.P.Val* < 0.05) of un-filtered dataset was used for the Gene Set Analysis

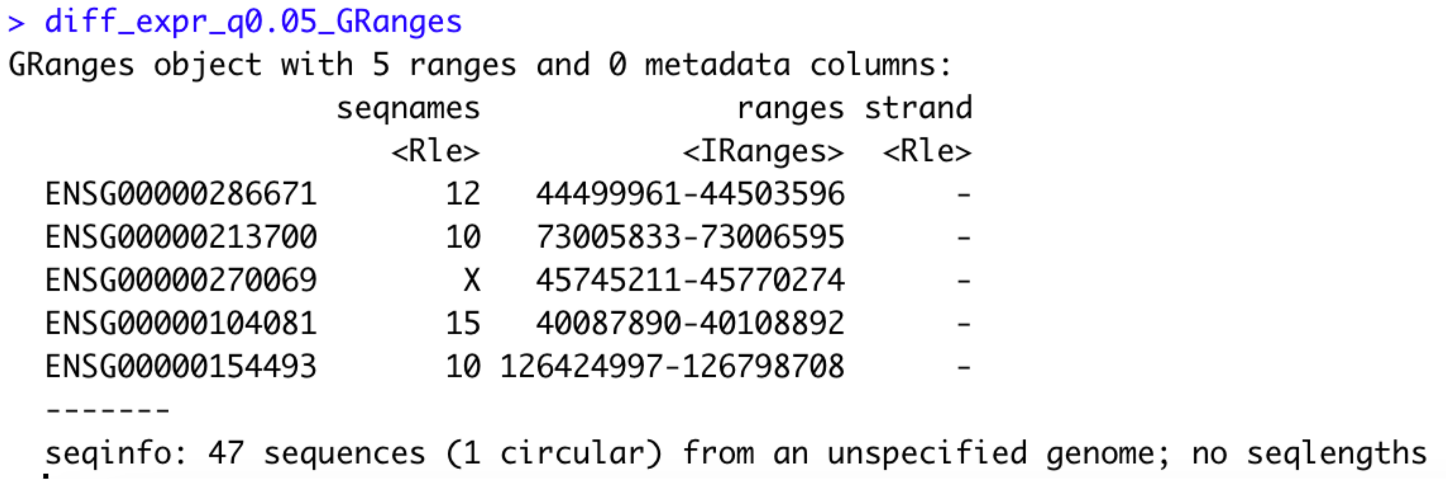
```
> library(GenomicFeatures)

> diff_expr = read.table("diff_expr.txt", sep="\t")
> diff_expr_q0.05 = subset(diff_expr, diff_expr$adj.P.Val < 0.05)
> head*diff_expr_q0.05)
```

2. Convert the differential gene expression data.frame to GRanges object and obtain promoter ranges

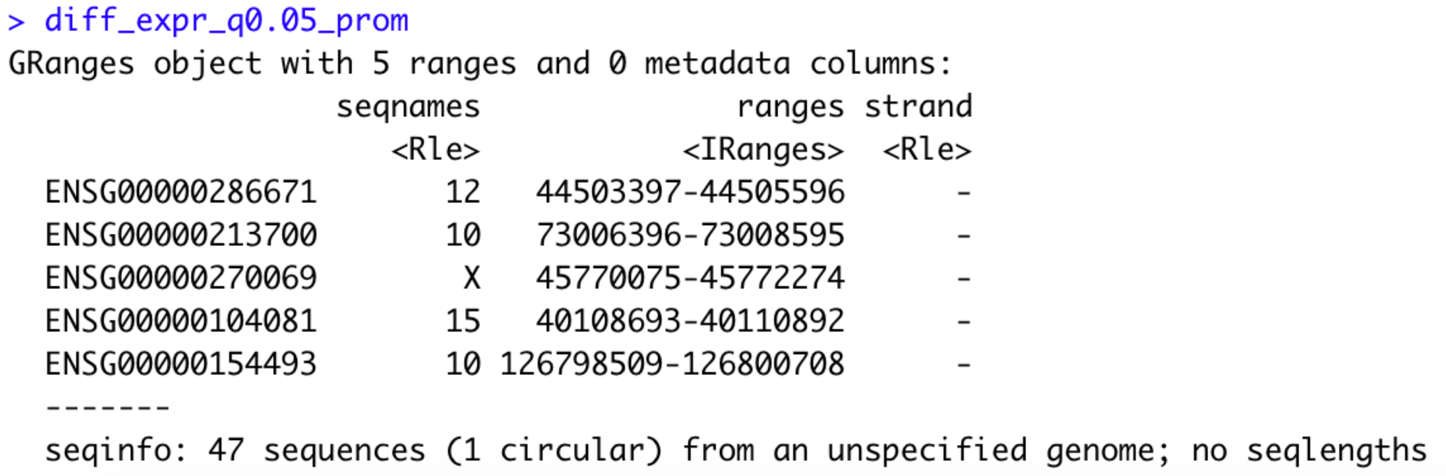
```
> txdb <- makeTxDbFromGFF(file="Homo_sapiens.GRCh38.
103.gtf", format="gtf")
# I decided to create txdb from the same .gtf file I used in transcript assemble
instead of downloading so that the information used will be consistent.
> exoRanges <- exonsBy(txdb, "gene") %>% range() %>% unlist()
> diff_expr_q0.05_GRanges <- exoRanges[na.omit(match(diff_expr_q0.05$gene_id,
names(exoRanges)))]
> diff_expr_q0.05_GRanges
> diff_expr_q0.05_prom <- promoters(diff_expr_q0.05_GRanges)
```



```
> head(diff_expr_q0.05)
                logFC  AveExpr         t      P.Value    adj.P.Val        B     gene_id
MSTRG.17204  6.694777 3.685240  43.21696 2.405985e-10 2.643215e-06 13.54935 MSTRG.17204
MSTRG.33102  6.379869 4.227618  38.61289 5.629045e-10 3.092035e-06 13.00736 MSTRG.33102
MSTRG.2098   5.334388 2.913842  33.37479 1.688873e-09 3.526310e-06 12.23015  MSTRG.2098
MSTRG.26564  6.483857 4.527418  33.05682 1.815112e-09 3.526310e-06 12.17626 MSTRG.26564
MSTRG.19254 -5.874675 3.461437 -32.56588 2.031549e-09 3.526310e-06 12.09135 MSTRG.19254
MSTRG.16347 -5.862089 3.176811 -32.01497 2.309952e-09 3.526310e-06 11.99353 MSTRG.16347
```



```
> diff_expr_q0.05_GRanges
GRanges object with 5 ranges and 0 metadata columns:
                  seqnames              ranges strand
                     <Rle>           <IRanges>  <Rle>
  ENSG00000286671       12   44499961-44503596      -
  ENSG00000213700       10   73005833-73006595      -
  ENSG00000270069        X   45745211-45770274      -
  ENSG00000104081       15   40087890-40108892      -
  ENSG00000154493       10 126424997-126798708      -
  -------
  seqinfo: 47 sequences (1 circular) from an unspecified genome; no seqlengths
```



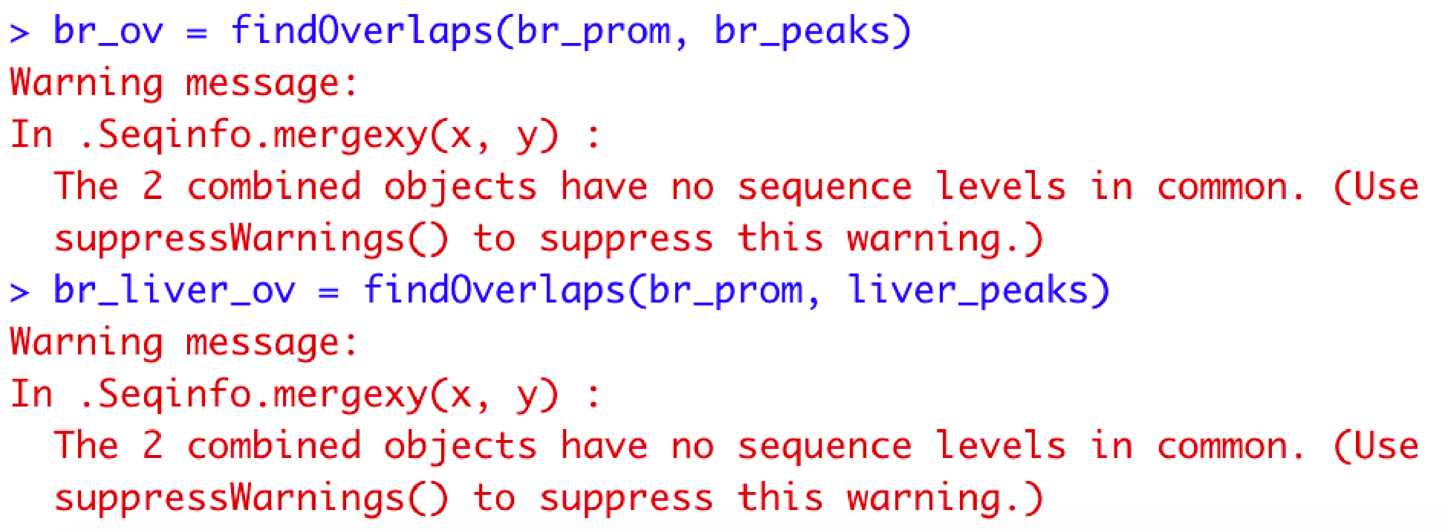
```
> diff_expr_q0.05_prom
GRanges object with 5 ranges and 0 metadata columns:
                  seqnames              ranges strand
                     <Rle>           <IRanges>  <Rle>
  ENSG00000286671       12   44503397-44505596      -
  ENSG00000213700       10   73006396-73008595      -
  ENSG00000270069        X   45770075-45772274      -
  ENSG00000104081       15   40108693-40110892      -
  ENSG00000154493       10 126798509-126800708      -
  -------
  seqinfo: 47 sequences (1 circular) from an unspecified genome; no seqlengths
```

3. Retrieve brain- and liver-specific H3K4me3 peaks using AnnotationHub

```
> library(AnnotationHub)
> library(rtracklayer)

> ahub = AnnotationHub()
> ahub = subset(ahub, species == "Homo sapiens")
> qhs1 = query(ahub, c("H3K4me3", "brain"))
> qhs2 = query(ahub, c("H3K4me3", "liver"))
> br_peaks = qhs1[["AH30413"]]
> liver_peaks = qhs2[["AH30367"]]
> br_prom = diff_expr_q0.05_prom
```

4. Find overlaps between promoters and H3K4me3 peaks

```
> br_ov = findOverlaps(br_prom, br_peaks)
> br_liver_ov = findOverlaps(br_prom, liver_peaks)
# There are no overlaps between br_prom and either brain- or liver-H3K4me3 peaks
```



```
> br_ov = findOverlaps(br_prom, br_peaks)
Warning message:
In .Seqinfo.mergexy(x, y) :
  The 2 combined objects have no sequence levels in common. (Use
  suppressWarnings() to suppress this warning.)
> br_liver_ov = findOverlaps(br_prom, liver_peaks)
Warning message:
In .Seqinfo.mergexy(x, y) :
  The 2 combined objects have no sequence levels in common. (Use
  suppressWarnings() to suppress this warning.)
```

5. A decision was made to test whether brain-H3K4me3 peaks are distinct from liver-H3K4me3 peaks as predicted using br_peaks and liver_peaks with the same downstream analysis for obtaining O.R.

```
> br_liver_peaks_ov = findOverlaps (br_peaks, liver_peaks)
> bs1 <- sum(width(intersect(liver_peaks, br_peaks, ignore.strand= TRUE)))
> bs2 <- sum(width(setdiff(liver_peaks, br_peaks, ignore.strand= TRUE)))
> bs3 <- sum(width(setdiff(br_peaks, liver_peaks, ignore.strand= TRUE)))
> bs4 = 3*10^8 - bs1 - bs2 - bs3
# Assuming the maximal length of all promoter sequences is 10% of the entire human
genome size
> oddsRatio = bs1*bs4/(bs2*bs3)
[1] 10.29928
```

Result & Discussion

1. The promoters of differentially expressed genes between Fetal and Adult brains are not associated with H3K4me3 modification from my dataset.

- This conclusion is unexpected and different from the results obtained by other students seen in peer reviews, though it can be explained by my dataset which is comprised almost exclusively of dataset-specific genes which were not annotated by the reference genome (gene_id prefix: MSTRG.)
- However, the question remains as to why annotated reference genes were not included in the differential gene expression table I obtained. The last clue I have which is to be tested is that I used stringtie for gene assembly which is not commonly used by my peers.

2. The Odds Ratio for the association between brain_peaks and liver_peaks is 10.3, suggesting that H3K4me3 modification in brain and liver promoters are strongly associated, which is not what was predicted. Further research is needed to provide possible explanations which is beyond the scope of this Task.