P0 Merged File Documentation

Jack Goon

7/30/2019

Description of the Problem

When working on an mRNA isoform analysis using MIAx00 data, I realized that 11 metadata samples did not have matching fastq files. There were also 33 fastq files that didn't have matching metadata samples. While I suspected that these fastq files corresponded to the metadata samples, there was no proof of a link. In addition, there was no hint to *how* these discrepancies were linked (i.e. were the fastq files merged, were any fastq files ignored, etc). Discussing with Alex, Rinaldo, and Karol provided no certainty. Below I will outline the links between the unmatched fastq files and the sample information in the metadata.

Summary

For each of the relavent samples, there were 3 fastq files. The original count data was generated by merging all 3 fastq files. I verified this by:

1. merging all 3 fastq files with the `cat` command in linux
2. aligning the fastq files with STAR
3. generating count data with featurecounts
4. comparing count data to the original data
5. comparing differential expression with the original analysis

Metadata Sample Information

SampleID	Library	CountFile	DPC	Condition	Response
MIA.P0.1.S44.L006.R1.001	MIA_P0_1	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	Saline	high
MIA.P0.10.S53.L006.R1.001	MIA_P0_10	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	Inhibitor	med
MIA.P0.11.S54.L006.R1.001	MIA_P0_11	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	Inhibitor	high
MIA.P0.2.S45.L006.R1.001	MIA_P0_2	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	Saline	med
MIA.P0.3.S46.L006.R1.001	MIA_P0_3	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	Saline	med
MIA.P0.4.S47.L006.R1.001	MIA_P0_4	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	Inhibitor	high
MIA.P0.5.S48.L006.R1.001	MIA_P0_5	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	PolyIC	high
MIA.P0.6.S49.L006.R1.001	MIA_P0_6	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	PolyIC	high
MIA.P0.7.S50.L006.R1.001	MIA_P0_7	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	PolyIC	med
MIA.P0.8.S51.L006.R1.001	MIA_P0_8	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	PolyIC	med
MIA.P0.9.S52.L006.R1.001	MIA_P0_9	Project_Nord_MERGED_HS674_mm9-50_fC0_UN.txt	19.5	Inhibitor	high

Initial fastq data

Each sample had 3 fastq files to start. These files share an identifying number with their sample, between 1-11, which appears immediately after "P0" in the name. NOTE: Outside of these 11 samples, this ID number is NOT unique.

SampleID	fastq files	path
MIA.P0.1.S44.L006.R1.001	MIA-P0-1_S47_L006_R1_001.fastq.gz	share/nordlab/users/jbgoon/RNAseq/Project_ANIZ_L6_H674P_Zdilar/MIA-P0-1_S47
^	MIA_P0_1_CGATGT_L003_R1_001.fastq.gz	share/nordlab/users/jbgoon/RNAseq/Project_Nord_L3_HS674/Sample_MIA_P0_1
^	MIA_P0_1_CGATGT_L007_R1_001.fastq.gz	share/nordlab/users/jbgoon/RNAseq/Project_Nord_L7_HS674/Sample_MIA_P0_1

Where "MIA.P0.1…" is the pattern for all sample 1 fastq files, "MIA.P0.2…" for sample two, and so on.

Final data, with server paths

After merging the appropriate fastq files, each sample has a single fastq file. In addition, I reorganized all MIAx00 fastq files, realigned all of them, and regenerated count data.

Final fastq info

SampleID	fastq file
MIA.P0.1.S44.L006.R1.001	MIA_P0_allmerged_1.fastq
MIA.P0.10.S53.L006.R1.001	MIA_P0_allmerged_10.fastq
MIA.P0.11.S54.L006.R1.001	MIA_P0_allmerged_11.fastq
MIA.P0.2.S45.L006.R1.001	MIA_P0_allmerged_2.fastq
MIA.P0.3.S46.L006.R1.001	MIA_P0_allmerged_3.fastq
MIA.P0.4.S47.L006.R1.001	MIA_P0_allmerged_4.fastq
MIA.P0.5.S48.L006.R1.001	MIA_P0_allmerged_5.fastq
MIA.P0.6.S49.L006.R1.001	MIA_P0_allmerged_6.fastq
MIA.P0.7.S50.L006.R1.001	MIA_P0_allmerged_7.fastq
MIA.P0.8.S51.L006.R1.001	MIA_P0_allmerged_8.fastq
MIA.P0.9.S52.L006.R1.001	MIA_P0_allmerged_9.fastq

These files can be found in /share/nordlab/users/jbgoon/fastq_data, which organizes all MIAx00 fastq files by timepoint and condition. The newly created "allmerged" fastq files are in their appropriate directories within /share/nordlab/users/jbgoon/fastq_data/P0

Final alignment/count info

Alignments for these fastq files (bam format) can be found in an identically structured directory /share/nordlab/users/jbgoon/alignment_data

Count data for all MIAx00 can be found in /share/nordlab/users/jbgoon/featurecounts/allcounts.txt

1. Merging fastq files

To merge the 3 fastq.gz files, I first moved all of the files to a temporary directory /share/nordlab/users/jbgoon/fastq_data/P0/Merging_Archive/temp. Then, I unpacked the files so that they would be mergeable.

```
#!/bin/bash

#SBATCH --job-name=unpack.sh
#SBATCH --time=02:00:00
#SBATCH --mem=32000

gunzip *.gz
```

Next, I merged the fastq files in two batches. The first batch merged the 2 Project_Nord files, and the second batch merged the new Project_Nord file to the Project_ANIZ file.

Batch 1:

```
#!/bin/bash

#SBATCH --job-name=cat.sh
#SBATCH --time=02:00:00
#SBATCH --mem=8000

cat MIA_P0_10_CCGTCC* > MIA_P0_CCGTCC_merged.fastq
cat MIA_P0_11_BarcodeMissing_GTCCGC* > MIA_P0_11_BarcodeMissing_GTCCGC_merged.fastq
cat MIA_P0_1_CGATGT* > MIA_P0_1_CGATGT_merged.fastq
cat MIA_P0_2_TGACCA* > MIA_P0_2_TGACCA_merged.fastq
cat MIA_P0_3_ACAGTG* > MIA_P0_3_ACAGTG_merged.fastq
cat MIA_P0_4_GCCAAT* > MIA_P0_4_GCCAAT_merged.fastq
cat MIA_P0_5_CAGATC* > MIA_P0_5_CAGATC_merged.fastq
cat MIA_P0_6_CTTGTA* > MIA_P0_6_CTTGTA_merged.fastq
cat MIA_P0_7_AGTCAA* > MIA_P0_7_AGTCAA_merged.fastq
cat MIA_P0_8_AGTTCC* > MIA_P0_8_AGTTCC_merged.fastq
cat MIA_P0_9_ATGTCA* > MIA_P0_9_ATGTCA_merged.fastq
```

Batch 2:

```
cat MIA_P0_10_CCGTCC_merged.fastq MIA-P0-10_S56_L006_R1_001.fastq > MIA_P0_allmerged_10.fastq
cat MIA_P0_11_BarcodeMissing_GTCCGC_merged.fastq MIA-P0-11_S57_L006_R1_001.fastq > MIA_P0_allmerged_11.fastq
cat MIA_P0_1_CGATGT_merged.fastq MIA-P0-1_S47_L006_R1_001.fastq > MIA_P0_allmerged_1.fastq
cat MIA_P0_2_TGACCA_merged.fastq MIA-P0-2_S48_L006_R1_001.fastq > MIA_P0_allmerged_2.fastq
cat MIA_P0_3_ACAGTG_merged.fastq MIA-P0-3_S49_L006_R1_001.fastq > MIA_P0_allmerged_3.fastq
cat MIA_P0_4_GCCAAT_merged.fastq MIA-P0-4_S50_L006_R1_001.fastq > MIA_P0_allmerged_4.fastq
cat MIA_P0_5_CAGATC_merged.fastq MIA-P0-5_S51_L006_R1_001.fastq > MIA_P0_allmerged_5.fastq
cat MIA_P0_6_CTTGTA_merged.fastq MIA-P0-6_S52_L006_R1_001.fastq > MIA_P0_allmerged_6.fastq
cat MIA_P0_7_AGTCAA_merged.fastq MIA-P0-7_S53_L006_R1_001.fastq > MIA_P0_allmerged_7.fastq
cat MIA_P0_8_AGTTCC_merged.fastq MIA-P0-8_S54_L006_R1_001.fastq > MIA_P0_allmerged_8.fastq
cat MIA_P0_9_ATGTCA_merged.fastq MIA-P0-9_S55_L006_R1_001.fastq > MIA_P0_allmerged_9.fastq
```

Then, I moved these fastq files to their places in /share/nordlab/users/jbgoon/fastq_data manually,

2. Aligning with STAR

First, I generated a STAR index with mm9 fasta/gtf files from Illumina's iGenome website.

```
#!/bin/bash
/share/nordlab/users/jbgoon/alignment/STAR-2.7.1a/bin/Linux_x86_64/STAR --runThreadN 8 --runMode genomeGenerate --genomeDir /share/nordlab/users/jbgoon/Reference/mm9/STAR_index --genomeFastaFiles /share/nordlab/users/jbgoon/Reference/mm9/Sequence/WholeGenomeFasta/genome.fa --sjdbGTFfile /share/nordlab/users/jbgoon/Reference/mm9/Annotation/Genes/genes.gtf --sjdbOverhang 49
```

I aligned all of the fastq files in /share/nordlab/users/jbgoon/fastq_data. I also did this in batches to make computation quicker and easier to fix if a problem arose. I will just provide the script for aligning the P0 files of interest.

```
#!/bin/bash
#SBATCH --mem=32000 # NOTE: 32000 MB was necessary.

array=('MIA_P0_allmerged_1' 'MIA_P0_allmerged_2' 'MIA_P0_allmerged_3' 'MIA_P0_allmerged_4' 'MIA_P0_allmerged_5' 'MIA_P0_allmerged_6' 'MIA_P0_allmerged_7' 'MIA_P0_allmerged_8' 'MIA_P0_allmerged_9' 'MIA_P0_allmerged_10' 'MIA_P0_allmerged_11')
for item in ${array[*]};
do
mkdir /share/nordlab/users/jbgoon/alignment_data/P0_some_merged/temp/$item
/share/nordlab/users/jbgoon/alignment/STAR-2.7.1a/bin/Linux_x86_64/STAR --runThreadN 16 --genomeDir /share/nordlab/users/jbgoon/Reference/mm9/STAR_index --readFilesIn /share/nordlab/users/jbgoon/fastq_data/P0_some_merged/temp/$item.fastq --outFileNamePrefix /share/nordlab/users/jbgoon/alignment_data/P0_some_merged/temp/$item/$item_ --outSAMtype BAM SortedByCoordinate
done
```

I moved the files the files to their appropriate directories manually. Inexplicably, the output files did not contain the designated prefix $item_, so I had to rename the files with the following (quite clever) script. As requested by Karol, the bam file names contain timepoint, condition, and sample ID. NOTE: This script renames all P0 files, not just the ones of interest. I used this basic script to name all of my alignment data.

```
for i in /share/nordlab/users/jbgoon/alignment_data/P0/*/*/*.bam;
do
FILENAME=$(basename ${i})
FILEPATH=$(dirname ${i})
SAMPLENAME=$(basename ${FILEPATH})
SAMPLEPATH=$(dirname ${FILEPATH})
CONDNAME=$(basename ${SAMPLEPATH})
CONDPATH=$(dirname ${SAMPLEPATH})
TIMENAME=$(basename ${CONDPATH})
TIMEPATH=$(dirname ${CONDPATH})
mv ${i} ${FILEPATH}/${TIMENAME}_${CONDNAME}_${SAMPLENAME}_Aligned.sortedByCoord.out.bam
done
```

3. Generating Count Data

Once all data was aligned, generating count data was quite straightforward with the following script:

```
#!/bin/bash
/share/nordlab/users/jbgoon/featurecounts/subread-1.6.4-Linux-x86_64/bin/featureCounts -a /share/nordlab/users/jbgoon/Reference/mm9/Annotation/Genes/genes.gtf -T 16 -t exon -g gene_id -o /share/nordlab/users/jbgoon/featurecounts/allcounts.txt /share/nordlab/users/jbgoon/alignment_data/*/*/*/*.bam
```

4. Comparing Count Data

The following data table compares the old count data to the new count data for each sample, across a small set of genes. This suggests that the new merging/alignment process works identically to whatever was done in the past. I will further test this assertion in step 5.

Gene	Sample1	Sample2	Sample3	Sample4	Sample5	Sample6	Sample7	Sample8	Sample9	Sample10	Sample11
0610005C13Rik	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
0610007P14Rik	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
0610009B22Rik	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
0610009L18Rik	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
0610009O20Rik	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
0610010B08Rik	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE

5. Comparing Differential Expression Data

The following chunks outline a DE analysis almost identical to the one used in the original analysis. If the fastq merging process is correct, the resulting new DE datashould match the old DE data.

```r
# Load packages
library(sva)
library(edgeR)
library(GenomicFeatures)
library(parallel)
library(ggplot2)
library(dplyr)
library(data.table)
library(lessR)
```

```r
# generate mm9 list of exon sizes
txdb <- makeTxDbFromGFF("/Users/jackbgoon/Desktop/NordLab/Reference/genes.gtf",format="gtf")
exons.list.per.gene <- exonsBy(txdb,by="gene")
exonic.gene.sizes <- parallel::mclapply(exons.list.per.gene,function(x){sum(width(reduce(x)))}) #use parallel to do it faster

# import count data
allcounts <- read.table("/Users/jackbgoon/Desktop/NordLab/DEanalysis/allcounts.txt", header = T, sep = "\t", as.is=T, stringsAsFactors=F)
```

Renaming data columns (housekeeping):

```r
# reducing column names to timepoint.condition.samplename
columns <- colnames(allcounts)
new_colnames <- columns[7:112]
new_colnames <- sub("_Aligned.sortedByCoord.out.bam", "", new_colnames)
new_colnames <- sub("X.share.nordlab.users.jbgoon.alignment_data.", "", new_colnames)

#get rid of second half of name, which is repetitive
new_colnames <- sub("_R1_001.", "/", new_colnames)
new_colnames<- sub(".P0_PolyIC", "/", new_colnames)
new_colnames <- sub(".P0_Saline", "/", new_colnames)
new_colnames <- sub(".P0_Inhibitor", "/", new_colnames)
new_colnames <- read.table(text = new_colnames, sep ="/", as.is = TRUE)$V1
new_colnames <- gsub("_", ".", new_colnames)
new_colnames <- gsub("-", ".", new_colnames)

# get rid of descriptive prefix, to leave just the sample name that matches sampleinfo
new_colnames <- gsub("Saline.", "/", new_colnames)
new_colnames <- gsub("PolyIC.", "/", new_colnames)
new_colnames <- gsub("Inhibitor.", "/", new_colnames)
new_colnames <- gsub("Unknown.", "/", new_colnames)
new_colnames <- read.table(text=new_colnames, sep = "/", as.is = TRUE)$V2

# replace old column names with new column names
colnames(allcounts)[7:112] <- new_colnames

# remove unwanted samples/columns
allcounts <- allcounts[,-c(2,3,4,5,6,62,63,111,112)] # descriptors and sample info not counted
allcounts <- allcounts[,-c(16,21,22,17,18,19,20,70,71,69,72,73,74,99,75,76,78,80,77,79,53,54,55,56,36,37,38,39)]
# outliers
expData <- allcounts

# load metadata (which accounts for outliers), adjust names to match countnames
metadata <- read.table("/Users/jackbgoon/Desktop/NordLab/Filing:Metadata/Old Metadata/metadata.csv", sep=",", header=T)

# housekeeping name changes
metanames <- metadata[,"SampleID"]
metanames <- sub(".R1.001", "", metanames)
metanames <- sub("MIA.e12.5.", "", metanames)
metanames <- sub("MIA.e17.5.", "", metanames)
metanames <- sub(".fastq.gz", "", metanames)

# re-labelled P0 samples that were merged and renamed, to match count data
metanames <- sub("MIA.P0.1.S44.L006","MIA.P0.allmerged.1",metanames)
metanames <- sub("MIA.P0.2.S45.L006","MIA.P0.allmerged.2",metanames)
metanames <- sub("MIA.P0.3.S46.L006","MIA.P0.allmerged.3",metanames)
metanames <- sub("MIA.P0.4.S47.L006","MIA.P0.allmerged.4",metanames)
metanames <- sub("MIA.P0.5.S48.L006","MIA.P0.allmerged.5",metanames)
metanames <- sub("MIA.P0.6.S49.L006","MIA.P0.allmerged.6",metanames)
metanames <- sub("MIA.P0.7.S50.L006","MIA.P0.allmerged.7",metanames)
metanames <- sub("MIA.P0.8.S51.L006","MIA.P0.allmerged.8",metanames)
metanames <- sub("MIA.P0.9.S52.L006","MIA.P0.allmerged.9",metanames)
metanames <- sub("MIA.P0.10.S53.L006","MIA.P0.allmerged.10",metanames)
metanames <- sub("MIA.P0.11.S54.L006","MIA.P0.allmerged.11",metanames)
metadata[,"SampleID"] <- metanames

#order both dataframes
sortedmetadata <- metadata[order(metadata$"SampleID"),]
sortedexpData=expData[,order(colnames(expData))]
geneid <- grep("Geneid", names(sortedexpData))
sortedexpData=sortedexpData[,c(geneid, (1:ncol(sortedexpData))[-geneid])]

expData <- sortedexpData #expData is the primary count dataframe
metadata <- sortedmetadata #metadata is the sample information

#check that allcount sample names == metadata names
print(paste("all count sample names match metadata sample names:", all(colnames(expData)[2:ncol(expData)] == metadata[,"SampleID"])))
```

```
## [1] "all count sample names match metadata sample names: TRUE"
```

Organizing data based on sample information:

```r
#create vectors/factors for samples in expData. Column names go like: dpc_group_sex.by.rna_response_lane_metadata[,"Library"]
# dpc: 12.5 - 19.5
# group: 1 saline, 2 polyic
# sex.by.rna: 1 male, 2 female
# response: 1 med response, 2 high response

group <- ifelse(metadata[,"Condition"]=="Saline",1,2)
group <- ifelse(metadata[,"Condition"]=="Inhibitor",3,group) #no inhibitor data in metadata
group <- factor(group)

sex <- ifelse(metadata[,"Sex"]=="M",1,2)
sex <- ifelse(metadata[,"Sex"]=="unknown",3,sex)
sex <- factor(sex)

sex.by.rna <- ifelse(metadata[,"sex_by_rna"]=="M","1","2")
sex.by.rna <- factor(sex.by.rna)

response <- ifelse(metadata[,"Response"]=="med",1,2)
response <- factor(response)

lane <- factor(metadata[,"Lane"])
dpc <- metadata[,"DPC"]
dpc <- ifelse(dpc=="P0", 19.5, dpc)
dpc <- ifelse(dpc=="e14.5", 14.5, dpc)
dpc <- ifelse(dpc=="e12.5", 12.5, dpc)
dpc <- ifelse(dpc=="e17.5", 17.5, dpc)
dpc <- as.factor(dpc)
expData_original <- expData
row.names(expData) <- expData[,1]
expData <- expData[,-1]
colnames(expData) <- paste(dpc, group, sex.by.rna, response, lane, metadata[,"Library"], sep="_")
```

Filtering data in preparation for EdgeR:

```r
# Generate RPKM dataframes, use to filter genes of low expression

# Gene lengths calculated with lapply
gene.lengths <- as.numeric(lapply(1:nrow(expData), function(x) FUN= as.numeric(exonic.gene.sizes[rownames(expData)[x]])))

#edgeR settings
rpkmData <- rpkm(expData, gene.length=gene.lengths, log=T, prior.count=.25)

#Removes batch effect associated with lane of sequencing. "design: optional design matrix relating to treatment conditions to be preserved". I think this is may be incorrect since the exp.data contains raw counts.Run this on rpkm data instead.
rpkmBatch <- removeBatchEffect(expData, batch=lane, design=cbind(dpc, group, sex.by.rna, response))

# reformat rpkm data, convert back to linear scale
df <- melt(rpkmData)
df$rpkm_linear <- 2^df$value

# filter for low expression
threshold <- 2 # min rpkm value that must be expressed in 2 samples
keep <- as.data.frame(melt(rowSums(rpkmData > threshold) >= 2))
keep$gene_name <- rownames(keep)
keep <- filter(keep, value == "TRUE")$gene_name

datExprTest <- as.data.frame(rpkmData)
datExprTest$gene_name <- rownames(datExprTest)
datExprTest <- filter(datExprTest, gene_name %in% keep)
rownames(datExprTest) <- datExprTest$gene_name
datExprTest$gene_name <- NULL

# use Rn45s as rRNA normalization factor
rRNA <- expData["Rn45s",]/colSums(expData)

# separate datapoints by condition, filtering out lane 12 and samples with high rRNA
control.datapoints <- intersect(which(group=="1"),  which(lane != "12"))
control.datapoints <- intersect(control.datapoints, which(rRNA < 0.01))

polyic.datapoints <- intersect(which(group=="2"),  which(lane != "12"))
polyic.datapoints <- intersect(polyic.datapoints,  which(rRNA < 0.01))
```

Now, EdgeR:

```r
# Following code taken straight from past DE analysis, as suggested by Karol.

use.cols <- c(control.datapoints, polyic.datapoints)

test.dpc <- dpc[use.cols]
test.sex.by.rna <- sex.by.rna[use.cols]
test.response <- response[use.cols]
test.rRNA <- as.numeric(rRNA)[use.cols]
test.group <- group[use.cols]
test.lane <- as.numeric(lane)[use.cols]
test.data <- expData[,use.cols]

design <- model.matrix(~test.sex.by.rna+test.dpc+test.lane+as.numeric(test.group))
y <- DGEList(counts=test.data, group=group[use.cols])
keep <- rowSums(cpm(y)>.1) >=2 #keeps only genes expressed in above min.cpm in at least 2 libraries in each group
y <- y[keep, , keep.lib.sizes=FALSE]
y <- estimateGLMCommonDisp(y,design)
y <- estimateGLMTrendedDisp(y,design)
y <- estimateGLMTagwiseDisp(y,design)
fit <- glmFit(y,design)
lrt <- glmLRT(fit) # Genewise Negative Binomial Generalized Linear Models.

glm.output <- topTags(lrt, n=Inf)
glm.output.full <- glm.output$table

glm.output.full$gene_name <- rownames(glm.output.full)
rownames <- NULL

# Function that essentially copies the code from above and analyzes just one timepoint.
single_timepoint_glm_function<- function(x){

  control.datapoints <- intersect(which(group=="1"),  which(lane != "12"))
  control.datapoints <- intersect(control.datapoints, which(rRNA < 0.01))
  control.datapoints <- intersect(control.datapoints, which(dpc == x))

  polyic.datapoints <- intersect(which(group=="2"),  which(lane != "12"))
  polyic.datapoints <- intersect(polyic.datapoints,  which(rRNA < 0.01))
  polyic.datapoints <- intersect(polyic.datapoints, which(dpc == x))


  use.cols <- c(control.datapoints, polyic.datapoints)

  test.dpc <- dpc[use.cols]
  test.sex.by.rna <- sex.by.rna[use.cols]
  test.response <- response[use.cols]
  test.rRNA <- as.numeric(rRNA)[use.cols]
  test.group <- group[use.cols]
  test.lane <- as.numeric(lane)[use.cols]
  test.data <- expData[,use.cols]

  min.cpm <- 0.1

  design <- model.matrix(~test.sex.by.rna+test.lane+as.numeric(test.group))
  y <- DGEList(counts=test.data, group=group[use.cols])
  keep <- rowSums(cpm(y)>.1) >=2 #keeps only genes expressed above .1 cpm in at least 2 libraries in each group
  y <- y[keep, , keep.lib.sizes=FALSE]
  y <- estimateGLMCommonDisp(y,design)
  y <- estimateGLMTrendedDisp(y,design)
  y <- estimateGLMTagwiseDisp(y,design)
  fit <- glmFit(y,design)
  lrt <- glmLRT(fit) # Genewise Negative Binomial Generalized Linear Models.

  glm.output <- topTags(lrt, n=Inf)
  #write.table(glm.output$table, "DE_Full_PolyIC.txt", sep="\t", col.names=T, row.names=T, quote=F)
  glm.output.full <- glm.output$table

  glm.output.full$gene_name <- rownames(glm.output.full)
  rownames(glm.output.full) <- NULL
  glm.output.full <- glm.output.full[,c(6,1:5)]

  glm.output.full

  #a <- volcano_plot_text(glm.output.full, k)

  #list(a, glm.output.full)
}
```

Plotting the new EdgeR data with the old analysis, to see if values stayed consistent after re-merging the fastq files and replicating the DE pipeline.

```r
# import old data
old.glm.output.full <- read.csv("/Users/jackbgoon/Desktop/NordLab/DEanalysis/old.19.5.glm.output.full")
old.glm.output.full$X <- NULL

# reformat new data
new.glm.output.full <- single_timepoint_glm_function("19.5")
rownames(new.glm.output.full) <- NULL

# merge data
colnames(old.glm.output.full) <- c("gene_name", paste0(colnames(old.glm.output.full)[2:6], "_old"))
colnames(new.glm.output.full) <- c("gene_name", paste0(colnames(new.glm.output.full)[2:6], "_new"))

b <- merge(old.glm.output.full, new.glm.output.full, by="gene_name", all=T)

# run plotting function
test <- ifelse(b$PValue_old <= 0.05 & b$PValue_new <= 0.05, "Both", "Non_significant")
test <- ifelse(b$PValue_old <= 0.05 & b$PValue_new > 0.05, "old_only", test)
test <- ifelse(b$PValue_old > 0.05 & b$PValue_new <= 0.05, "new_only", test)

b$sig_by_method <- test

s <- ggplot(b, aes(x=logFC_old, y=logFC_new, color=sig_by_method))+
  geom_point(alpha=0.6)+
  theme_bw()+
  scale_color_manual(values=c("Both"="red", "old_only"="green", "new_only"="blue", "Non_significant"="black"))+
  labs(title=paste("Old vs new methods: logFC consistency in P0 timepoint"))

s
```


Conclusion

As you can see, the re-merging and re-analysis of the P0 data produced almost identical results, proving the fastq merging process successful. New merged files can confidently be used for future analyses. In addition, newly aligned bam files and the new count file can be used for future anlayses as well.