Erica Chio, Assigned coursework #4

Due date: March 12th (two weeks), 11pm EST

Submission: PDF document by email to daniel.depledge@nyulangone.org

1. Datasets: HCMV-infected normal human dermal fibroblasts treated with either

`/gpfs/data/courses/bminga3004/Practicum5/Assignment/`

- a non-silencing control (LT34/LT35/LT36)

  LT34_R1.fastq.gz LT35_R2.fastq.gz

  LT34_R2.fastq.gz LT36_R1.fastq.gz

  LT35_R1.fastq.gz LT36_R2.fastq.gz
- an EIF3D-silencing siRNA (LT46/LT47/LT48)

  LT46_R1.fastq.gz LT47_R2.fastq.gz

  LT46_R2.fastq.gz LT48_R1.fastq.gz

  LT47_R1.fastq.gz LT48_R2.fastq.gz
- Note: each condition as three biological replicates each
- Illumina paired-end dataset generated using NEB Ultra II stranded protocol [dUTP]

2. Align data against human genome (Tophat2) and transcriptome (kallisto)

bash specifications for all scripts:

```
#!/bin/bash
#SBATCH --job-name=<filename> # Job name
#SBATCH --mail-type=END,FAIL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Erica.Chio@nyulangone.org # Where to send mail
#SBATCH --ntasks=16 # Run on a single CPU
#SBATCH --mem=32gb # Job memory request
#SBATCH --time=24:00:00 # Time limit hrs:min:sec
#SBATCH --output=<filename>_%j.log # Standard output and error log
#SBATCH -p cpu_short # Specifies location to submit job
```

- Human genome: **you will need to point to the bowtie2 indexed version of the human genome that you downloaded previously (see week three lecture slide 50)**

```
/gpfs/scratch/ebc308/Homo_sapiens/UCSC/hg38/Sequence/Bowtie2Index/genome
```

```
module purge

module load trimgalore/0.5.0
module load python/cpu/2.7.15-ES
module load bowtie2/2.3.5.1
module load tophat/2.1.1
module load samtools/1.3
module load subread/1.6.3

do
echo $i

trim_galore --paired --length 30 -o /gpfs/scratch/ebc308/AIS/coursework4data/ /
gpfs/data/courses/bminga3004/Practicum5/Assignment/"$i"_R1.fastq.gz /gpfs/data/
courses/bminga3004/Practicum5/Assignment/"$i"_R2.fastq.gz &

done < /gpfs/scratch/ebc308/AIS/coursework4data/datasets.txt
```

```
module purge

module load trimgalore/0.5.0
module load python/cpu/2.7.15-ES
module load bowtie2/2.3.5.1
module load tophat/2.1.1
module load samtools/1.3
module load subread/1.6.3

echo ${SLURM_ARRAY_TASK_ID}

tophat -o /gpfs/scratch/ebc308/AIS/coursework4data/LT${SLURM_ARRAY_TASK_ID}
_tophat/ -G /gpfs/scratch/ebc308/AIS/coursework4data/genes.gtf -p 8 --library-
type fr-firststrand /gpfs/scratch/ebc308/Homo_sapiens/UCSC/hg38/Sequence/
Bowtie2Index/genome /gpfs/scratch/ebc308/AIS/coursework4data/LT$
{SLURM_ARRAY_TASK_ID}_R1_val_1.fq.gz /gpfs/scratch/ebc308/AIS/coursework4data/
```

```
LT${SLURM_ARRAY_TASK_ID}_R2_val_2.fq.gz
```

```
sbatch --array=34,35,36,46,47,48 tophatArray.sh
```

- Human transcriptome: `/gpfs/data/courses/bminga3004/Practicum5/Homo_sapiens.GRCh38.cdna.all.fa`

```
module load kallisto/0.44.0

### Indexing
kallisto index -i /gpfs/scratch/ebc308/AIS/coursework4data/HomoSapiens /gpfs/scratch/ebc308/AIS/coursework4data/Homo_sapiens.GRCh38.cdna.all.fa

while read i;

do
echo $i

kallisto quant -i /gpfs/scratch/ebc308/AIS/coursework4data/HomoSapiens -o /gpfs/scratch/ebc308/AIS/coursework4data/"$i"_kallisto -b 100 --bias /gpfs/data/courses/bminga3004/Practicum5/Assignment/"$i"_R1.fastq.gz /gpfs/data/courses/bminga3004/Practicum5/Assignment/"$i"_R2.fastq.gz

done < /gpfs/scratch/ebc308/AIS/coursework4data/datasets.txt
```

3. Generate gene counts from genome alignment using featurecounts or HTSEQ

- GTF file: `/gpfs/data/courses/bminga3004/Practicum5/genes.gtf`

```
featureCounts -s 2 -p -B -a /gpfs/scratch/ebc308/AIS/coursework4data/genes.gtf -o /gpfs/scratch/ebc308/AIS/coursework4data/final_counts.txt /gpfs/scratch/ebc308/AIS/coursework4data/LT34_tophat/accepted_hits.bam /gpfs/scratch/ebc308/
```

```
AIS/coursework4data/LT35_tophat/accepted_hits.bam /gpfs/scratch/ebc308/AIS/
coursework4data/LT36_tophat/accepted_hits.bam /gpfs/scratch/ebc308/AIS/
coursework4data/LT46_tophat/accepted_hits.bam /gpfs/scratch/ebc308/AIS/
coursework4data/LT47_tophat/accepted_hits.bam /gpfs/scratch/ebc308/AIS/
coursework4data/LT48_tophat/accepted_hits.bam
```

4. Generate transcript counts from transcriptome alignment + turn into gene counts

```
### LOAD REQUIRED LIBRARIES
library(biomaRt)
library(tximport)
library(rhdf5)

### SET WORKING DIRECTORY ### You will need to edit this and direct it your
downloaded kallisto folder
setwd("/Users/ericachio/Documents/sackler/applied informatics sequencing/
coursework4/Kallisto")

### IMPORT ENSEMBl ANNOTATIONS FOR HUMAN GENOME & GENERATE TWO COLUMN FILE
LINKING TRANSCRIPbiocMT AND GENE IDS
mart <- biomaRt::useMart(biomart = "ensembl", dataset =
"hsapiens_gene_ensembl")
t2g <- biomaRt::getBM(attributes = c("ensembl_transcript_id",
"transcript_version", "ensembl_gene_id", "external_gene_name", "description",
"transcript_biotype", "refseq_mrna", "refseq_ncrna"), mart = mart)
t2g$target_id <- paste(t2g$ensembl_transcript_id, t2g$transcript_version,
sep=".") # append version number to the transcript ID
t2g[,c("ensembl_transcript_id","transcript_version")] <- list(NULL) # delete
the ensembl transcript ID and transcript version columns
t2g <- dplyr::rename( t2g, gene_symbol = external_gene_name, full_name =
description, biotype = transcript_biotype )
t2g<-t2g[,c(ncol(t2g),1:(ncol(t2g)-1))]

### GENERATE ADDITIONAL OBJECT CONTAINING ONLY PROTEIN CODING GENES
gb <- getBM(attributes=c("ensembl_gene_id","gene_biotype"), mart=mart)
gb_coding<-subset(gb, gb$gene_biotype=="protein_coding")
```

```
genes<-gb_coding$ensembl_gene_id

### USE TXIMPORT TO SUMMARIZE TRANSCRIPT COUNTS INTO GENE COUNTS
## For single sample (can be abundance.h5 or abundance.tsv file)
#files <- file.path("Ctrl1", "abundance.h5")
#names(files)<-"test_sample1"
#tx.kallisto <- tximport(files, type = "kallisto", tx2gene = t2g,
countsFromAbundance ="no")

## For multiple samples, each named as a folder in the kallisto directory (can
be abundance.h5 or abundance.tsv file)
accessions <- list.dirs(full.names=FALSE)[-1]
kallisto.dir<-paste0(accessions)
kallisto.files<-file.path(kallisto.dir,"abundance.h5") #can also be
abundance.tsv
names(kallisto.files)<- accessions
tx.kallisto <- tximport(kallisto.files, type = "kallisto", tx2gene = t2g,
countsFromAbundance ="no")

### GENERATE TWO COLUMN OUTPUT FORMAT, ROUND VALUES (DESEQ2 DOES NOT LIKE
FRACTIONS), AND WRITE TO OUTPUT FILE
counts<-as.data.frame(tx.kallisto$counts[row.names(tx.kallisto$counts) %in%
genes, ])
len <- as.data.frame(tx.kallisto$len[row.names(tx.kallisto$len) %in% genes, ])
ids<-rownames(counts)

### WRITE OUTPUT TABLE
write.table(round(counts),paste("output",".txt",sep=""), row.names=ids,
quote=F, col.names=T, sep="\t")
```

5. Generate scatter plots of gene counts (genome) vs. gene counts (transcriptome) and calculate correlation

```
library(dplyr)
library(ggplot2)
library(ggpubr)
library(scales)
```

```
temp_counts <- read.table("counts/final_counts.txt", stringsAsFactors = FALSE)
# temp_counts
counts <- temp_counts %>%
  select(V1, V7,V8,V9,V10,V11,V12)

names <- data_frame("Geneid", "LT34", "LT35", "LT36", "LT46", "LT47", "LT48")
# counts <- setNames(counts, )
counts <- setNames(counts, names)
counts <- counts[-1,]
counts[,1] <- sub("*\\.[0-9]", "", counts[,1])
rownames(counts) <- counts$Geneid
counts[,1] <- NULL
counts
```

```
temp_output <- read.table("kallisto/output.txt", stringsAsFactors = FALSE)
df = merge(counts, temp_output, by=0)
df
```

code for every dataset

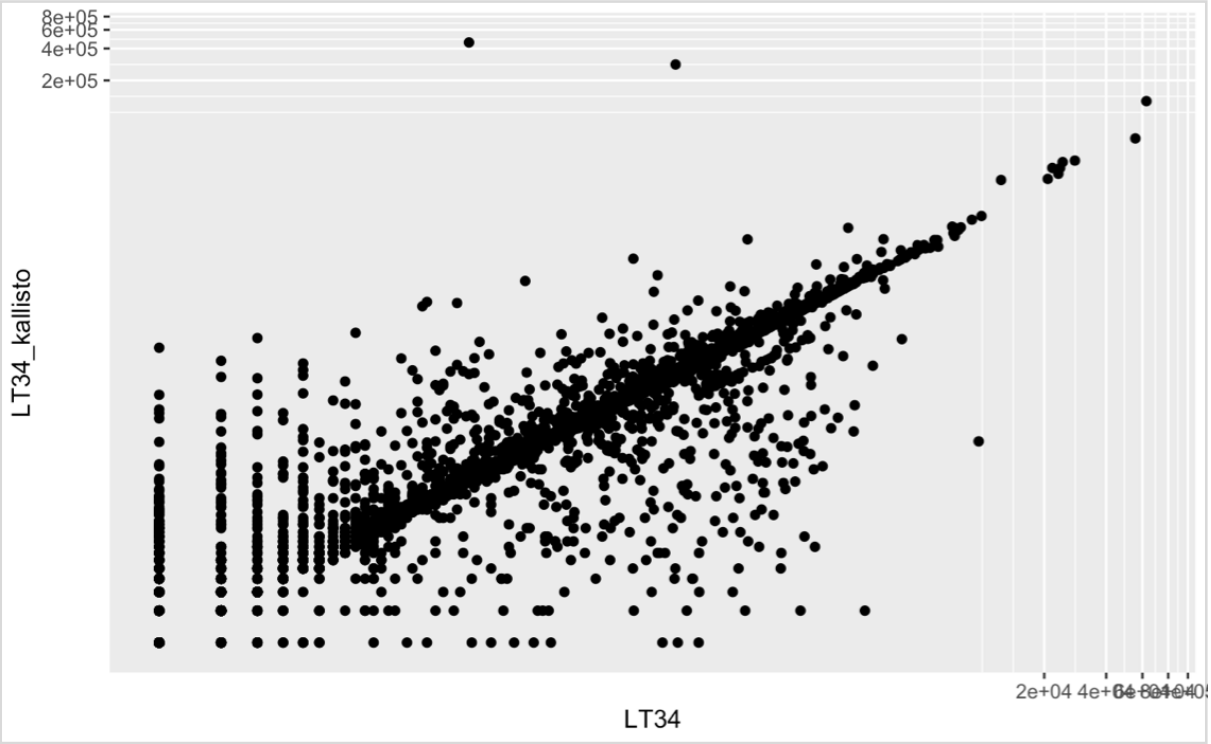
```
LT34 <- df %>%
  select(Row.names, LT34,LT34_kallisto)
LT34 <- filter(LT34, LT34 != 0 & LT34_kallisto != 0)
LT34 <- transform(LT34, LT34 = as.numeric(LT34))
```

```
p <- ggplot(LT34, aes(x=LT34, y=LT34_kallisto)) + geom_point()
p + scale_x_log10(breaks=pretty_breaks()) +
scale_y_log10(breaks=pretty_breaks())
```

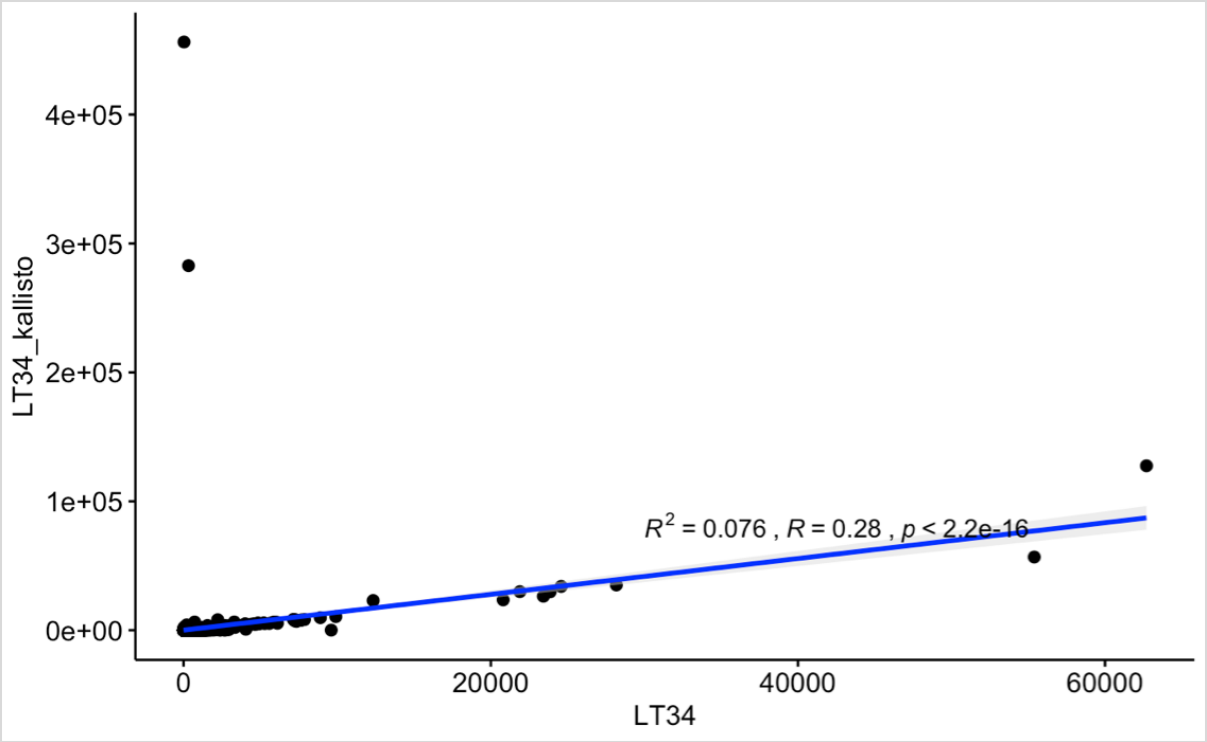
```
ggscatter(LT34, x = "LT34", y = "LT34_kallisto", add = "reg.line", conf.int =
TRUE, add.params = list(color = "blue", fill = "lightgray")) +
stat_cor(aes(label = paste(..rr.label.., ..r.label.., sep = "~`,`~")), label.x
= 30000, label.y = 80000)
```

LT34

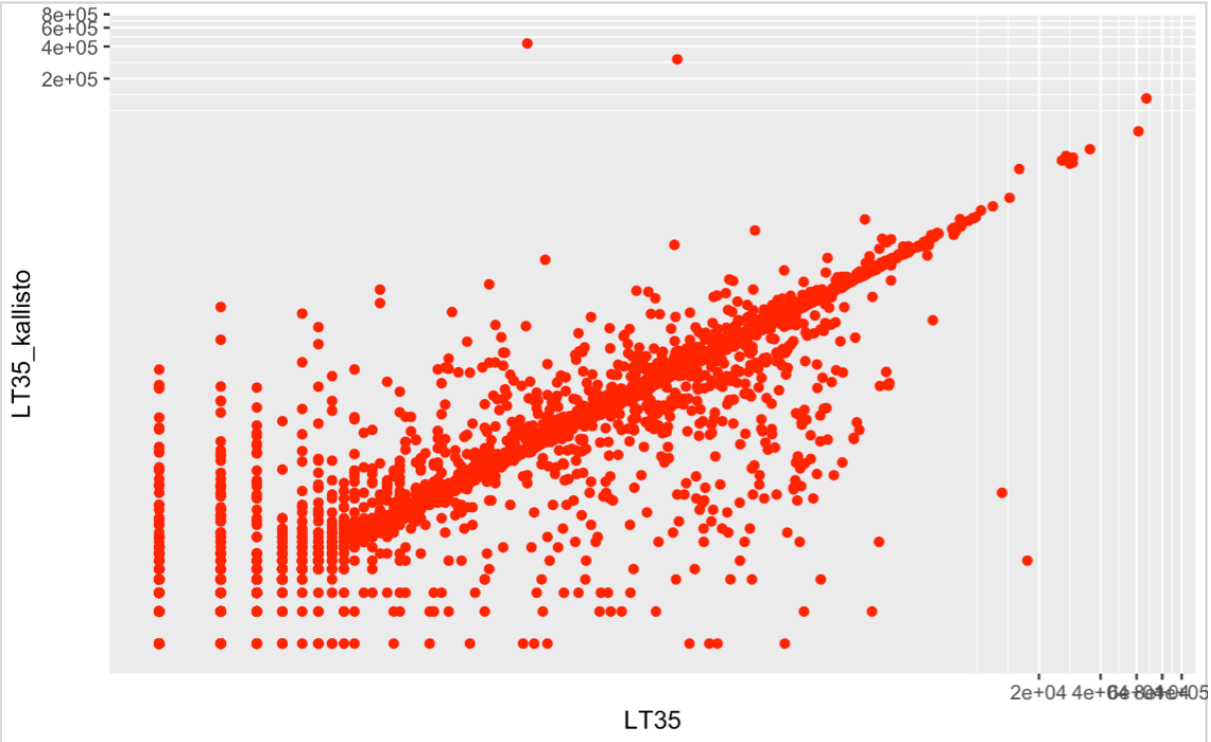
I used regular ggplot to plot features count vs kallisto counts. I applied scale_x_log to see what the points looked like. It seems to indicate a correlation between the genome and transcriptome counts.

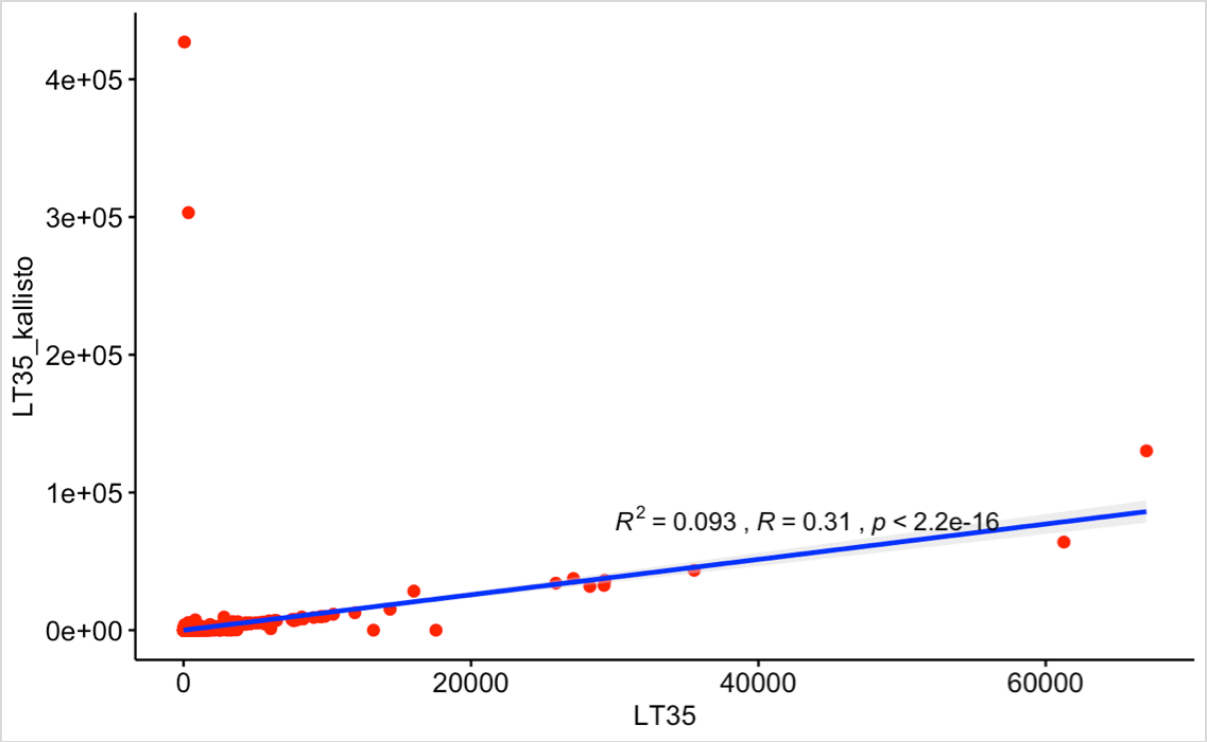


To get the R, R^2, value and p-value to see correlation, I used ggscatter.

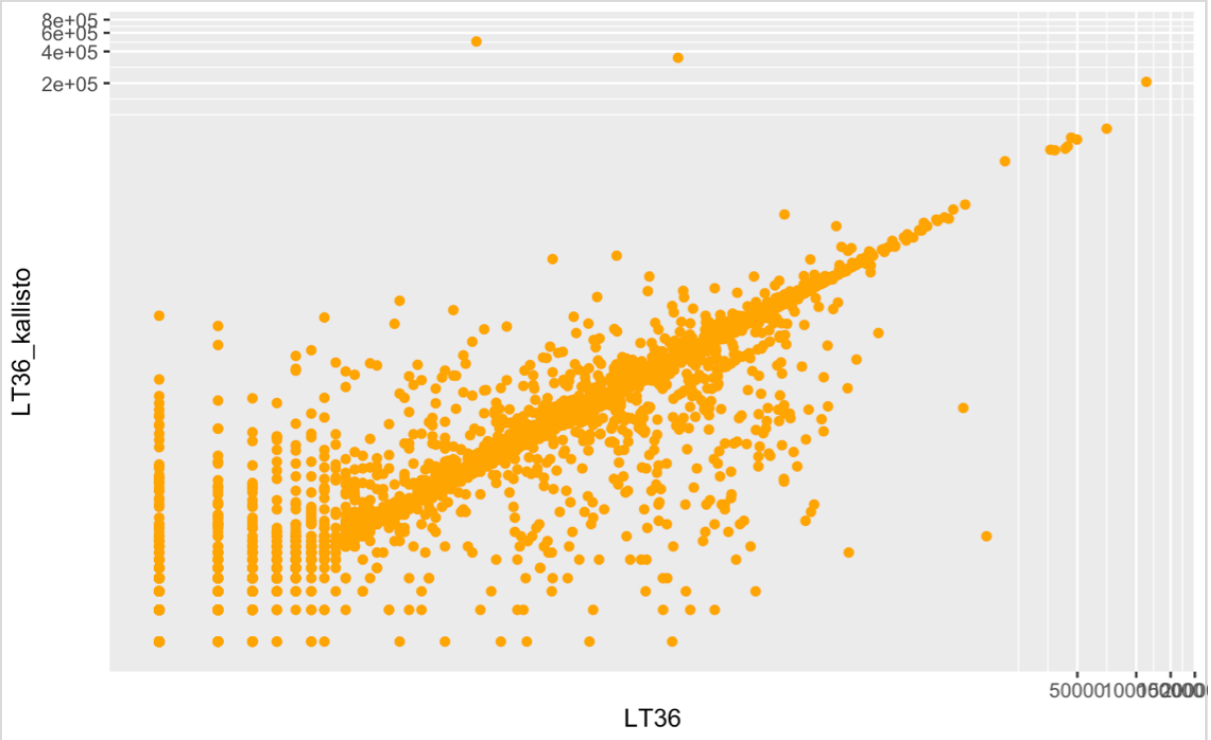

4e+05
3e+05
2e+05
1e+05
0e+00
LT34_kallisto
R^2 = 0.076 , R = 0.28 , p < 2.2e-16
0
20000
40000
60000
LT34

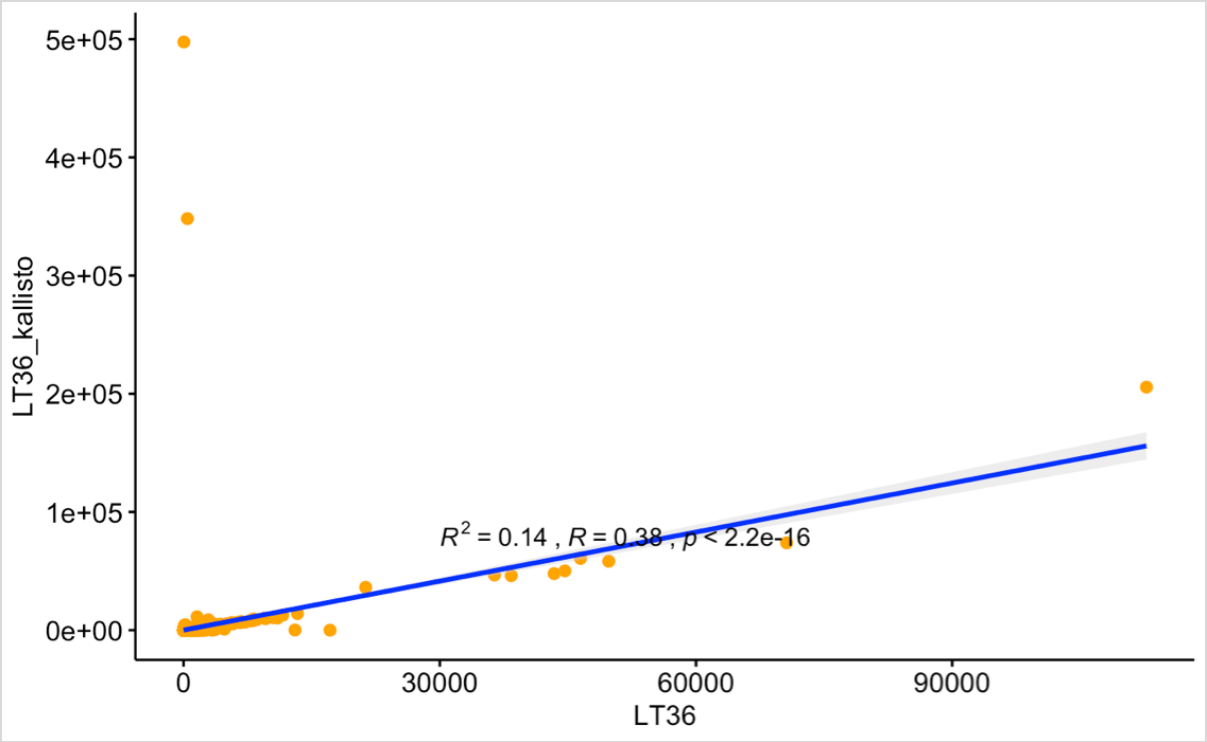
LT35


8e+05
6e+05
4e+05
2e+05
LT35_kallisto
LT35

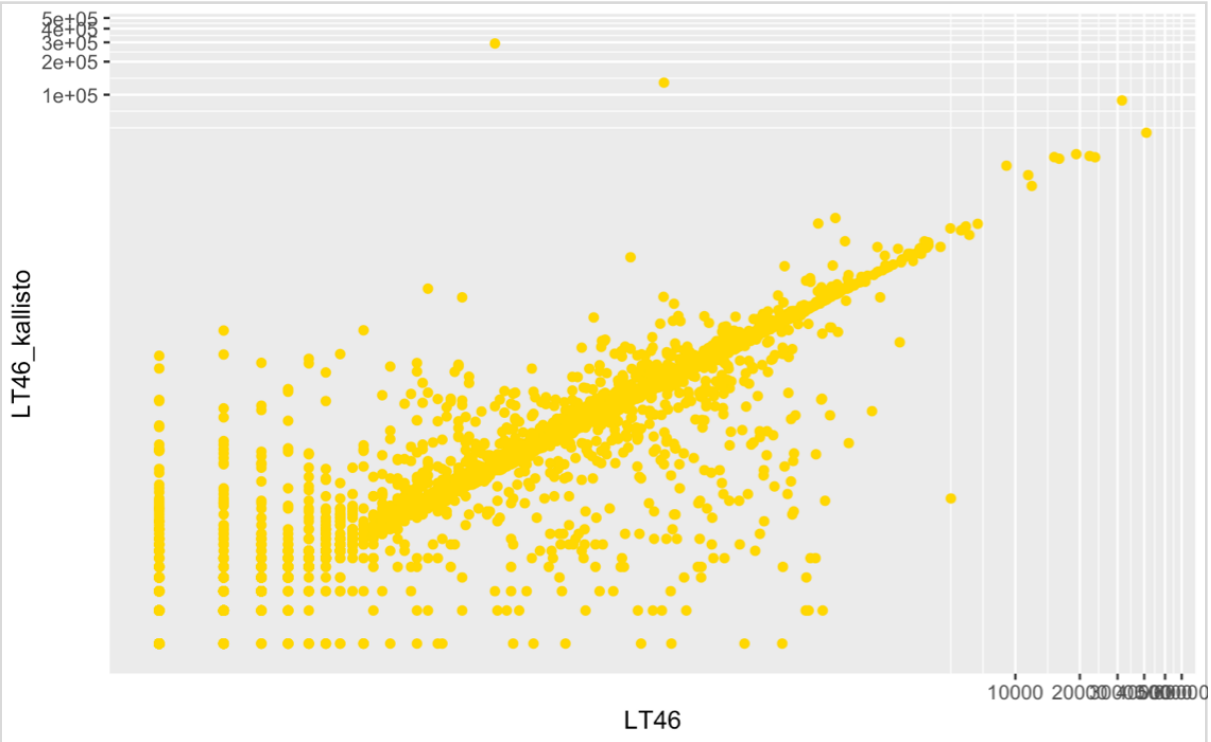

4e+05
3e+05
2e+05
1e+05
0e+00
LT35_kallisto
0
20000
40000
60000
LT35
R^2 = 0.093 , R = 0.31 , p < 2.2e-16

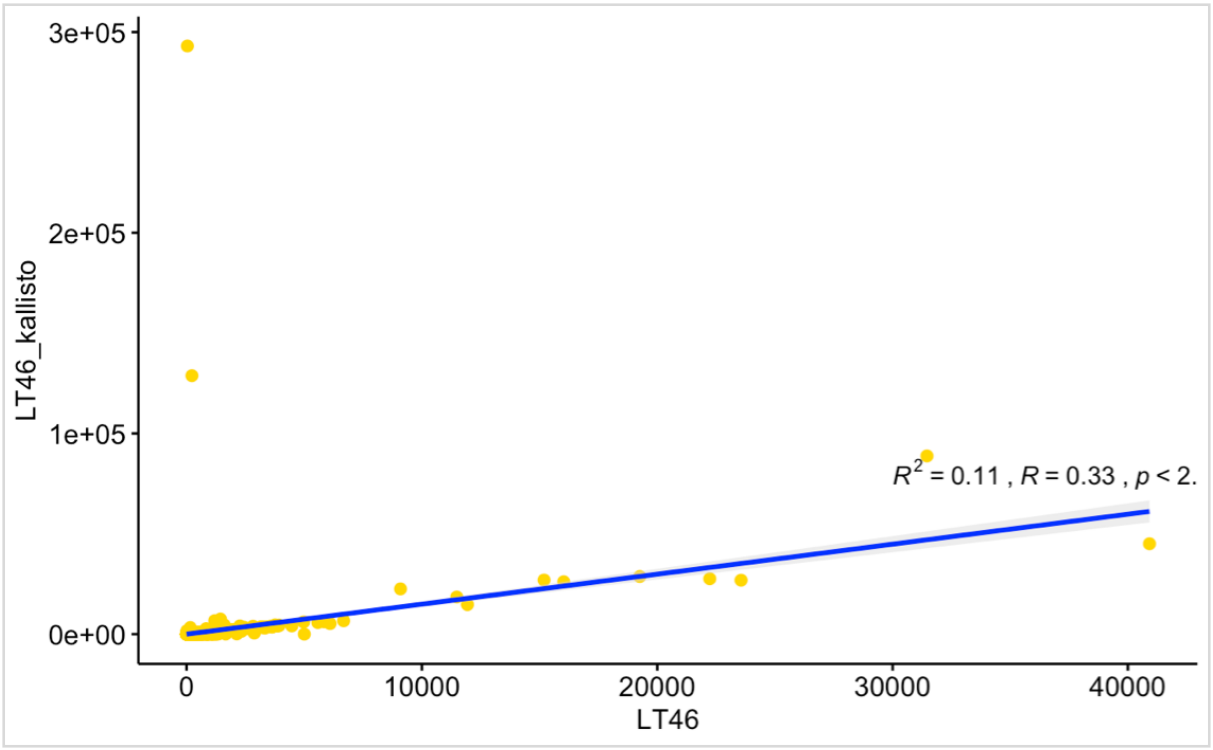
LT36


8e+05
6e+05
4e+05
2e+05
LT36_kallisto
LT36

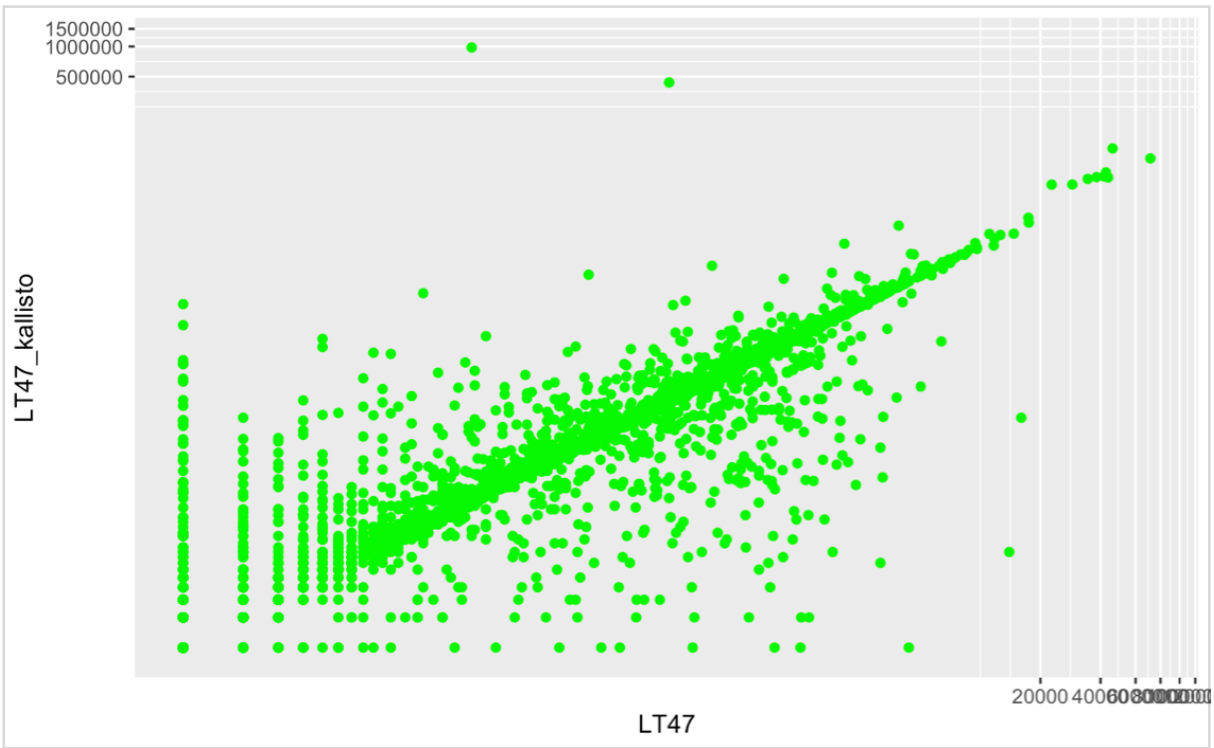

5e+05
4e+05
3e+05
2e+05
1e+05
0e+00
LT36_kallisto
R^2 = 0.14 , R = 0.38 , p < 2.2e-16
0
30000
60000
90000
LT36

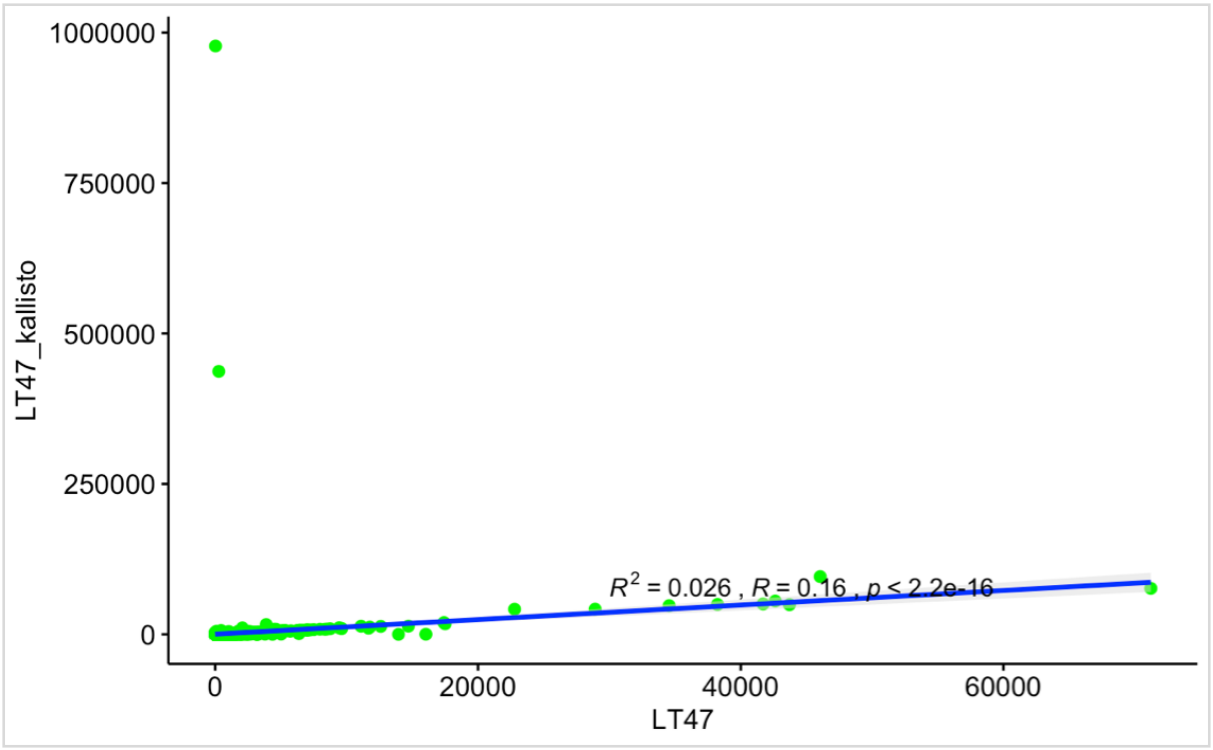
LT46


5e+05
4e+05
3e+05
2e+05
1e+05
LT46_kallisto
10000
LT46

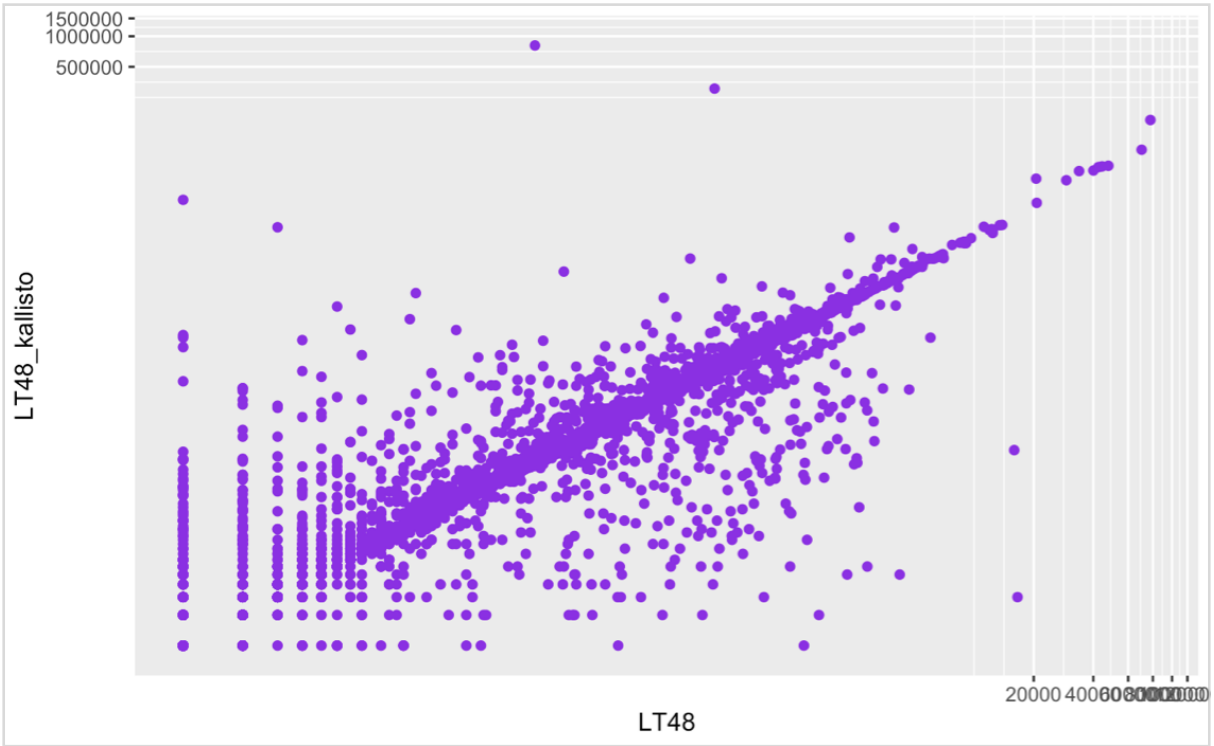

3e+05
2e+05
1e+05
0e+00
LT46_kallisto
R^2 = 0.11 , R = 0.33 , p < 2.
0
10000
20000
30000
40000
LT46

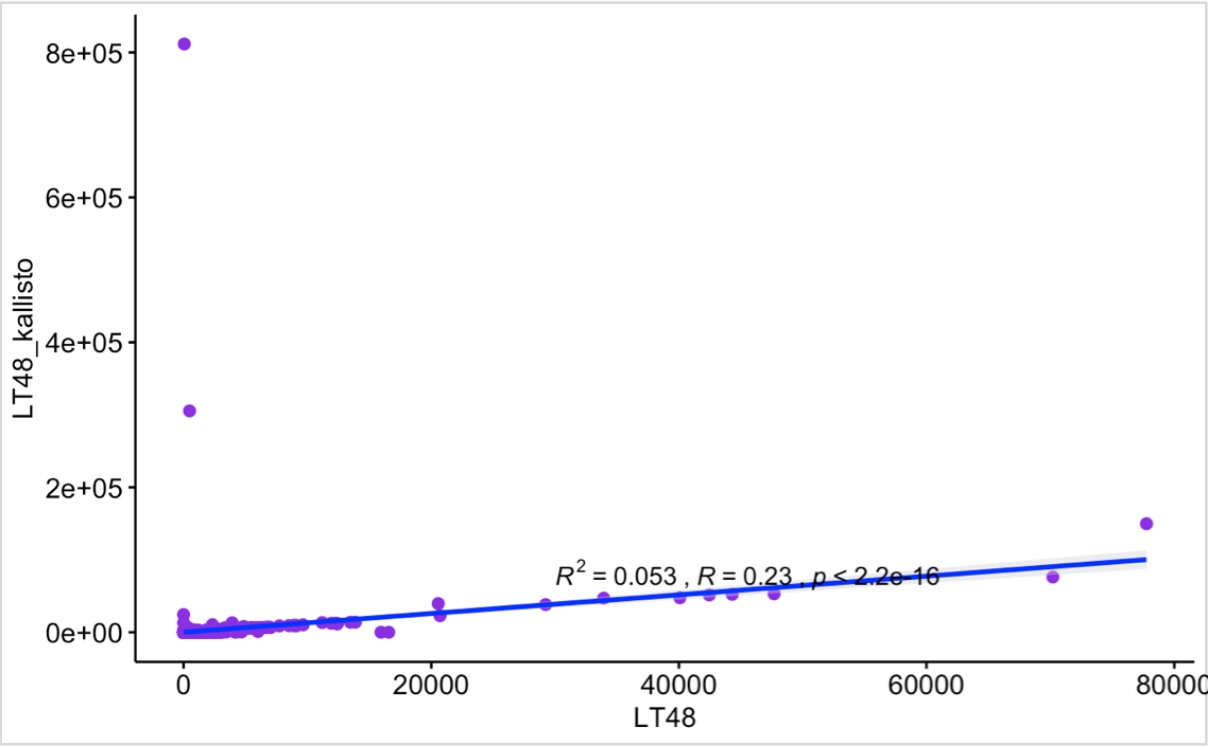
LT47


1500000
1000000
500000
LT47_kallisto
20000 40000
LT47


1000000
750000
500000
250000
0
LT47_kallisto
R^2 = 0.026 , R = 0.16 , p < 2.2e-16
0
20000
40000
60000
LT47

LT48


1500000
1000000
500000
LT48_kallisto
20000
LT48



Are the results similar? If not, can you experiment with featurecounts/htseqto improve this?

These results seem pretty similar to me. All the p-values are less than 0.5, which indicate there is a correaltion. R² values seem to be on the smaller side, but I think that since most of the data points are clustered at one point, it is harder for it to represent the whole model.

Provide full coding, scatter plots, and R² values, as well as a short write-up on the final question

#appliedsequencinginformatics