indexSwapRevisited

Devin Dinwiddie

September 12, 2017

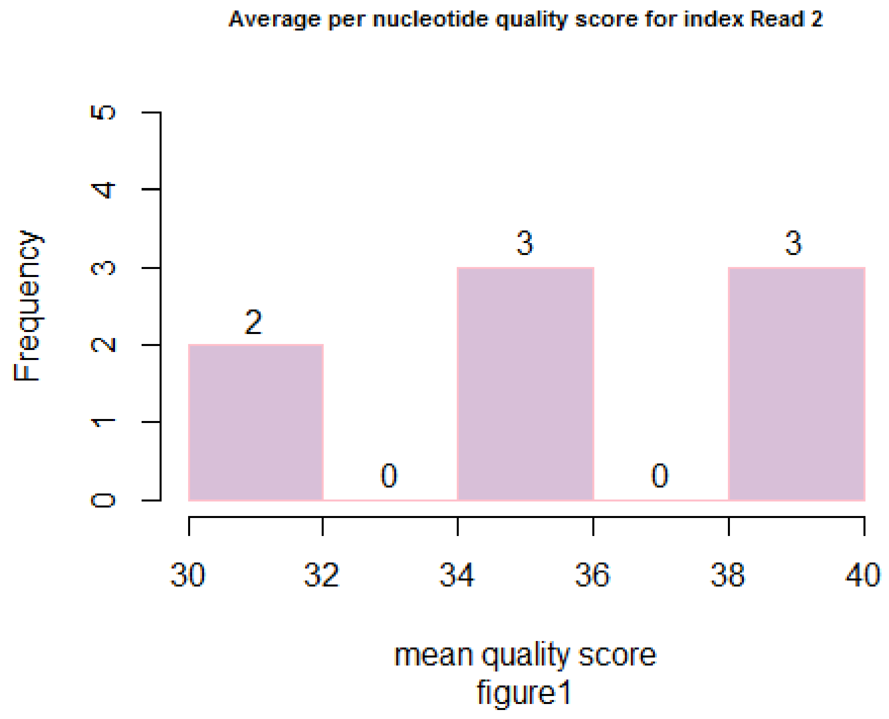
1)

a)

```
#plot  average nucleotide  quality scores in a histogram for index reads ie :
R2, R3

# read in the data
meanBp2=read.table("meanBasePairQual2.tsv")
meanBp3=read.table("meanBasePairQual3.tsv")

#make the plots
#par(mfrow=c(1,2))
hist(meanBp2$V2, labels = TRUE, ylim=c(0, 5), col = "thistle", border =
"pink",main="Average per nucleotide quality score for index Read 2",xlab =
"mean quality score",cex.main=.7,sub="figure1")
```



```r
hist(meanBp3$V2,labels = TRUE, ylim=c(0, 5), col = "thistle", border =
"pink",main="Average per nucleotide quality score for index Read 3",xlab =
"mean quality score",cex.main=.7,sub="figure2")
```

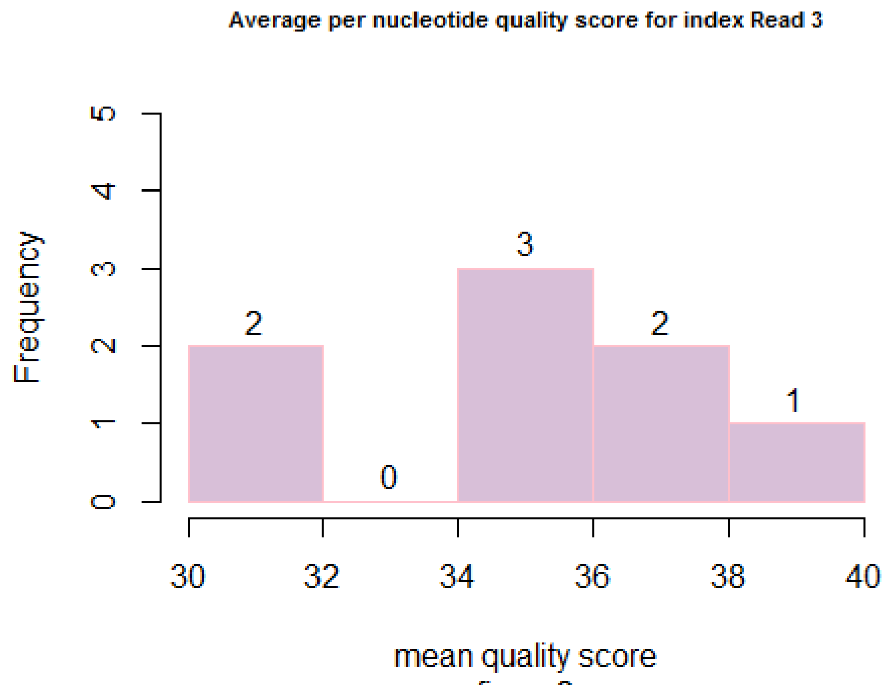
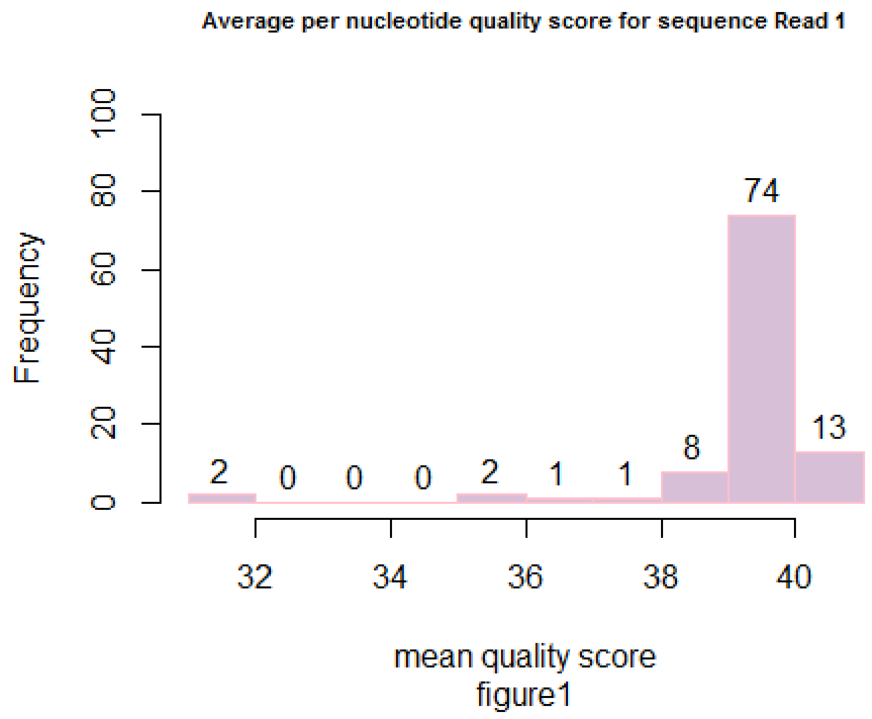


figure2

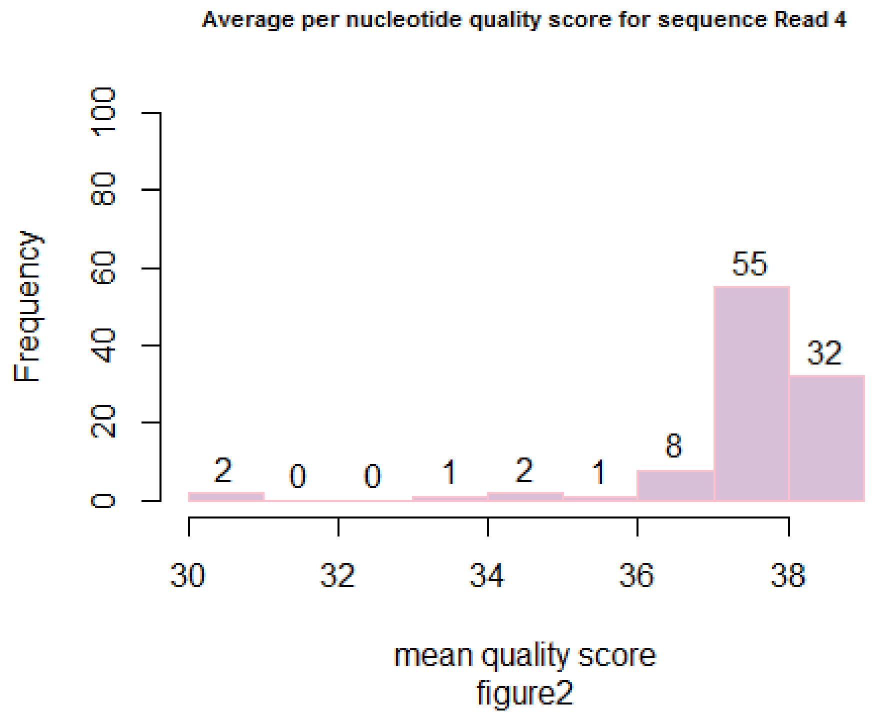
```r
#plot  average nucleotide  quality scores in a histogram for sequence reads
ie : R1, R4

# read in the data
meanBp1=read.table("meanBasePairQual1.tsv")
meanBp4=read.table("meanBasePairQual4.tsv")

#make the plots
#par(mfrow=c(1,2))
hist(meanBp1$V2, labels = TRUE, ylim=c(0, 100), col = "thistle", border =
"pink",main="Average per nucleotide quality score for sequence Read 1",xlab =
"mean quality score",cex.main=.7,sub="figure1")
```



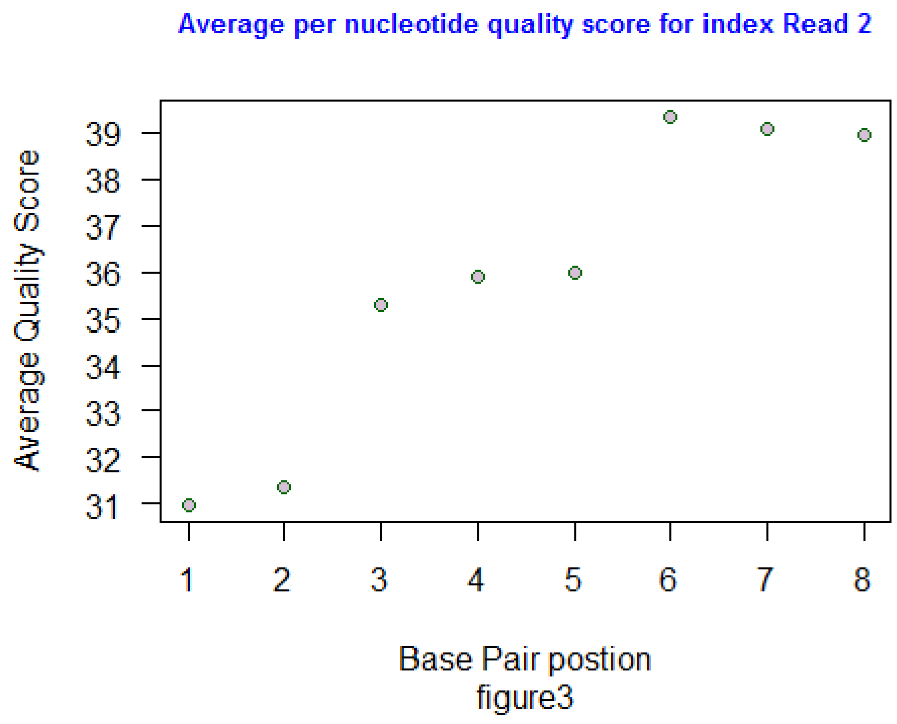
```
hist(meanBp4$V2,labels = TRUE, ylim=c(0, 100), col = "thistle", border =
"pink",main="Average per nucleotide quality score for sequence Read 4",xlab =
"mean quality score",cex.main=.7,sub="figure2")
```



```r
# the histograms are ok but lets visualize as bp position on the x and
quality score on the y axis
# ie make a dot plot of the data

#the Base pair files start @ pos 0 and go to 7 thats confusing for non
programers so lets change that so BP pos 1 ==1 not 0
names=seq(1:8) # create a seq of all numbers from 1 -8
meanBp3$V1=names # replace the column holding postions  with above names ie
1-8
meanBp2$V1=names

plot(meanBp2$V1,meanBp2$V2, yaxt='n',xaxt='n',col = "darkgreen",main="Average
per nucleotide quality score for index Read 2",xlab = "Base Pair
postion",ylab="Average Quality
Score",cex.main=.8,col.main="blue",sub="figure3",cex=1,pch=21,bg="thistle")
axis(1,at=seq(1:8),labels = c(1,2,3,4,5,6,7,8))
axis(2,at=seq_along(1:40),labels = seq(1,40,by=1),las=2)
```



```r
plot(meanBp3$V1,meanBp3$V2, yaxt='n',xaxt='n',col = "darkgreen",main="Average
per nucleotide quality score for index Read 3",xlab = "Base Pair
postion",ylab="Average Quality
Score",cex.main=.8,col.main="blue",sub="figure4",cex=1,pch=21,bg="thistle")
axis(1,at=seq(1:8),labels = c(1,2,3,4,5,6,7,8))
axis(2,at=seq_along(1:40),labels = seq(1,40,by=1),las=2)
```

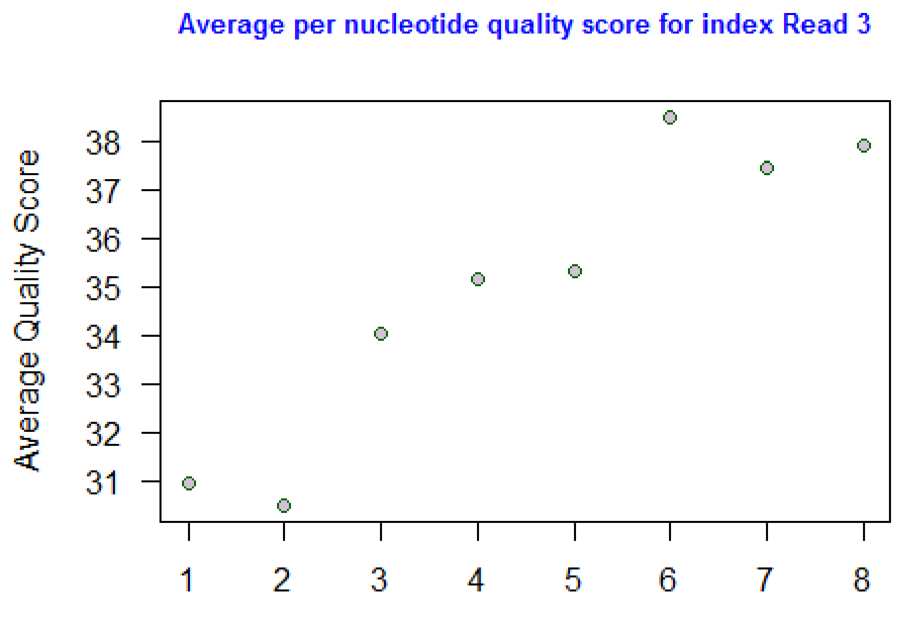
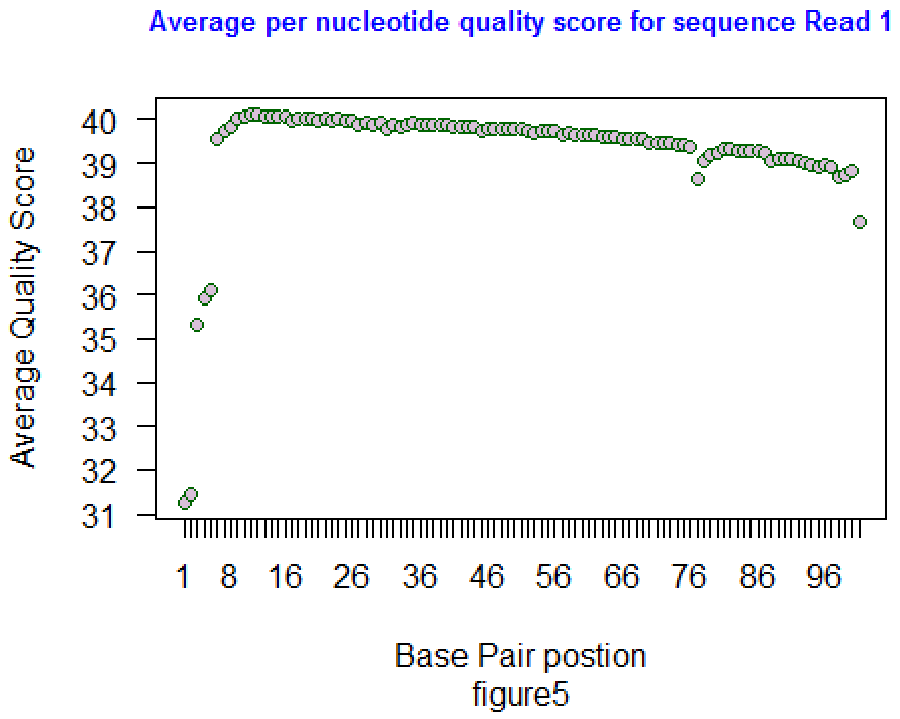


figure4

```
# repeate with sequence reads 1 and 4

#the Base pair files start @ pos 0 and go to 101 thats confusing for non
programers so lets change that so BP pos 1 ==1 not 0
names2=seq(1:101) # create a seq of all numbers from 1 -8
meanBp1$V1=names2 # replace the column holding postions  with above names ie
1-8
meanBp4$V1=names2

plot(meanBp1$V1,meanBp1$V2, yaxt='n',xaxt='n',col = "darkgreen",main="Average
per nucleotide quality score for sequence Read 1",xlab = "Base Pair
postion",ylab="Average Quality
Score",cex.main=.8,col.main="blue",sub="figure5",cex=1,pch=21,bg="thistle")
axis(1,at=seq(1:101),labels = seq(1, 101, by=1))
axis(2,at=seq_along(1:40),labels = seq(1,40,by=1),las=2)
```



```
plot(meanBp4$V1,meanBp4$V2, yaxt='n',xaxt='n',col = "darkgreen",main="Average
per nucleotide quality score for sequence Read 4",xlab = "Base Pair
postion",ylab="Average Quality
Score",cex.main=.8,col.main="blue",sub="figure6",cex=1,pch=21,bg="thistle")
axis(1,at=seq(1:101),labels = seq(1, 101, by=1))
axis(2,at=seq_along(1:40),labels = seq(1,40,by=1),las=2)
```

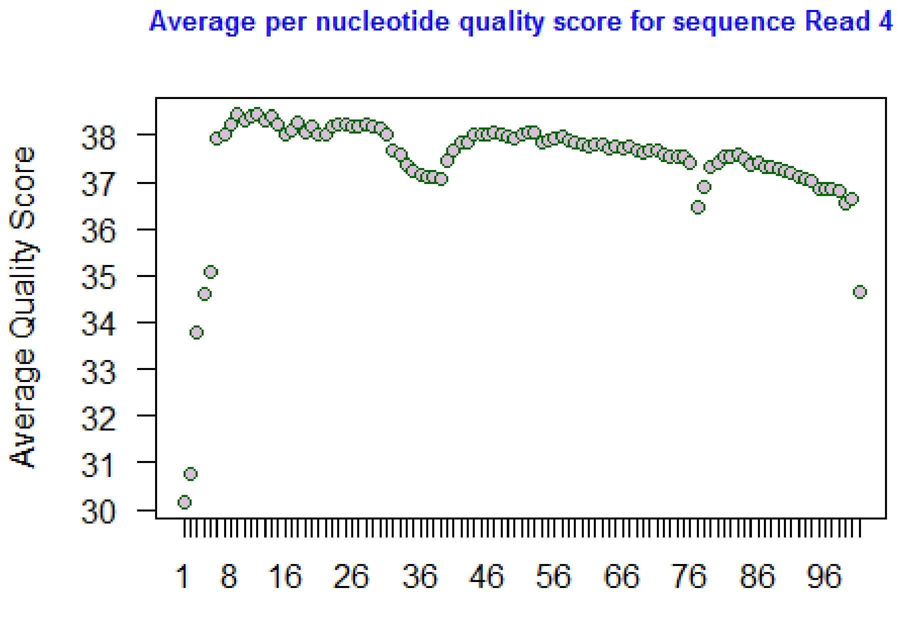
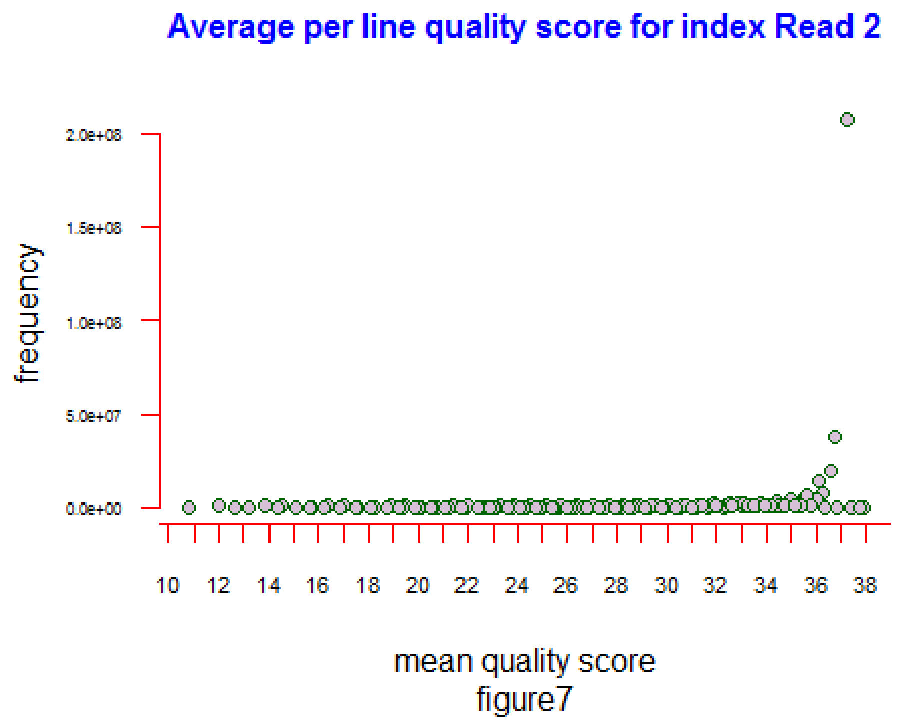


figure6

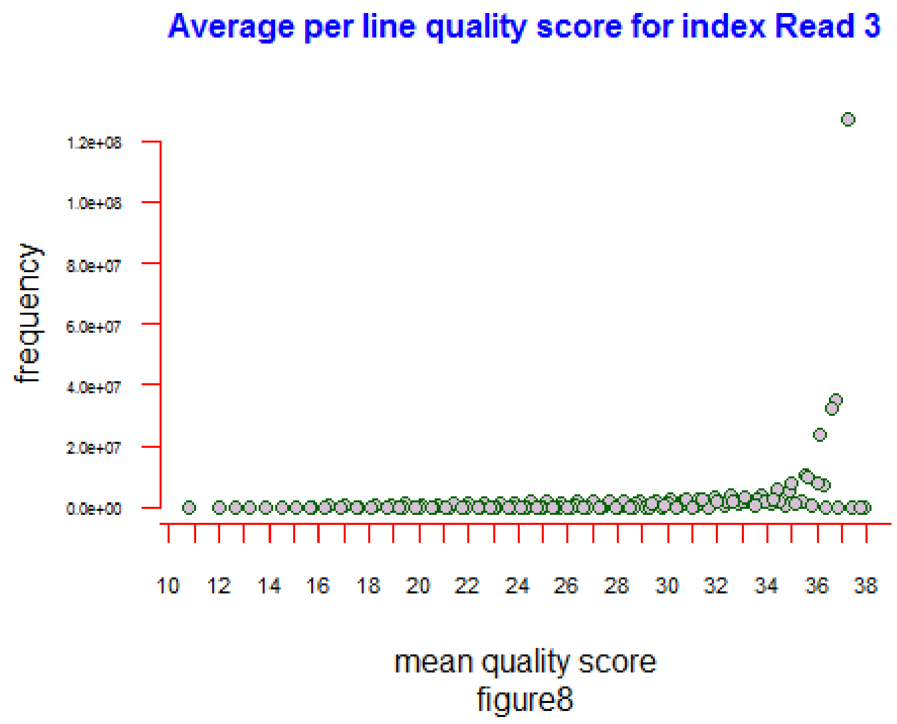
```
# plot the per read quality score distributions for index reads 2 and 3

# read in the data
meanLineQ2=read.table("meanLineQual2.tsv")
meanLineQ3=read.table("meanLineQual3.tsv")

#make the plots
plot(meanLineQ2$V1,meanLineQ2$V2,axes=F, col = "darkgreen",main="Average per
line quality score for index Read 2",xlab = "mean quality
score",ylab="frequency",cex.main=1,col.main="blue",sub="figure7",cex=1,pch=21
,bg="thistle")
axis(1,col = "red",cex.axis=.7,at=seq_along(1:40))
axis(2,col="red",las=1,cex.axis=.5)
```



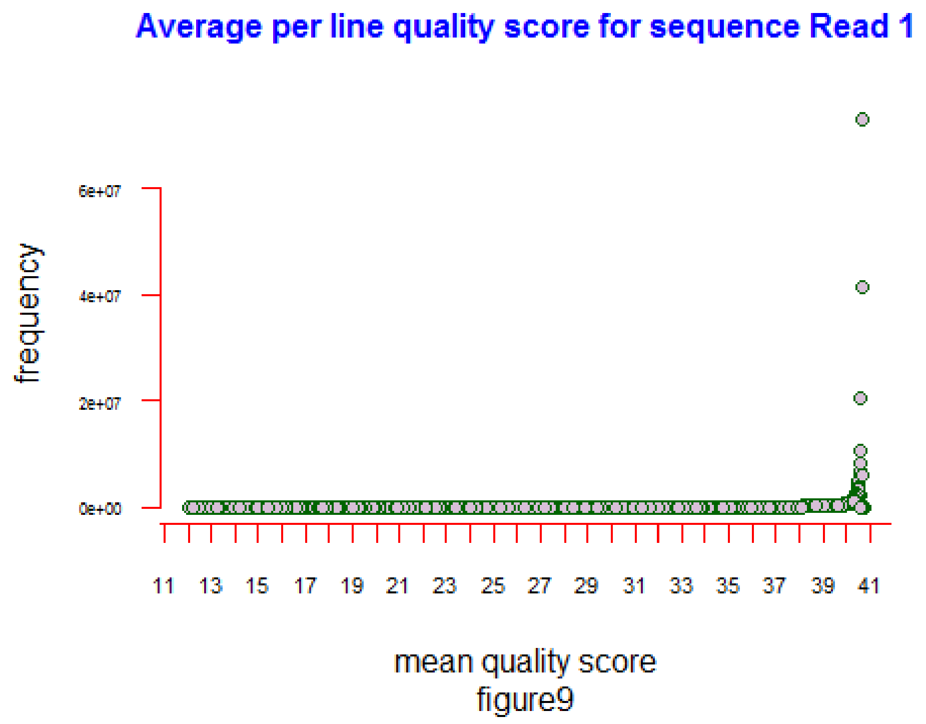
```
plot(meanLineQ3$V1,meanLineQ3$V2,axes=F, col = "darkgreen",main="Average per
line quality score for index Read 3",xlab = "mean quality
score",ylab="frequency",cex.main=1,col.main="blue",sub="figure8",cex=1,pch=21
,bg="thistle")
axis(1,col = "red",cex.axis=.7,at=seq_along(1:40))
axis(2,col="red",las=1,cex.axis=.5)
```



```
# plot the per read quality score distributions for sequence  reads 1 and 4

# read in the data
meanLineQ1=read.table("meanLineQual1.tsv")
meanLineQ4=read.table("meanLineQual4.tsv")

#make the plots
plot(meanLineQ1$V1,meanLineQ1$V2,axes=F, col = "darkgreen",main="Average per
line quality score for sequence Read 1",xlab = "mean quality
score",ylab="frequency",cex.main=1,col.main="blue",sub="figure9",cex=1,pch=21
,bg="thistle")
axis(1,col = "red",cex.axis=.7,at=seq_along(1:42))
axis(2,col="red",las=1,cex.axis=.5)
```



```
plot(meanLineQ4$V1,meanLineQ4$V2,axes=F, col = "darkgreen",main="Average per
line quality score for sequence Read 4",xlab = "mean quality
score",ylab="frequency",cex.main=1,col.main="blue",sub="figure10",cex=1,pch=2
1,bg="thistle")
axis(1,col = "red",cex.axis=.7,at=seq_along(1:42))
axis(2,col="red",las=1,cex.axis=.5)
```

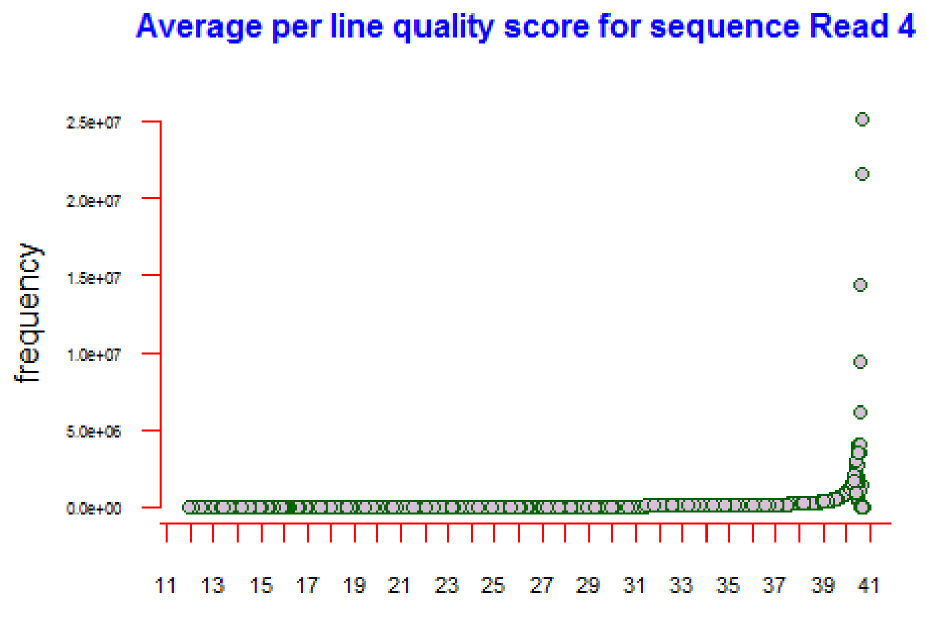


figure10

b)

based on the above graphs the majority of the quality scores are above 34 for reads and per base pair. The first couple of NT's in a read tend to have the lowest quality scores with an average just above 30. The script I wrote for detecting index swapping looks at quality scores on a per nucleotide basis, removing whole reads if one nucleotide is below the cutoff. I feel like 30 being the low average is a good place to start for quality score cutoff, for index reads and pairs being utilized for sample identification and downstream analysis.

c)

```
# use 1 line command to determine the number of undermined ("N") base pair
calls for both index files R2 and R3
#R2
cat 1294_S1_L008_R2_001.fastq | awk 'NR%4==2' | grep -c 'N'

3976613 #number of N calls in R2

#R3
cat 1294_S1_L008_R3_001.fastq | awk 'NR%4==2' | grep -c 'N'
3328051 # number of N calls in R3
```

d)

The average quality score across reads tells me that the vast majority of my reads are high quality 38+. you can see from the graphs reads with quality scores of 10 to ~ 35 all seem to occur at low frequencies then you see a upturn and the quality scores 36+ occur in much larger frequencies. This tells me that overall the data that I am working with is of pretty high quality which is good.

2)

a)

```
#determine the number of reads retained for each  expected index pair ie: UDI
# calculate the percentage of correctly indexed reads

# read in the data
indexes=read.table("R2_3_index.tsv",sep="\t")

#clean up the data table ie : remove ()
indexes$V1=substr(indexes$V1,2,19)
indexes$V1=gsub(",","",indexes$V1) #remove the comma seperator

# read in the index file ie the file of indexes that where used in the
experiment
udi=read.table("indexes.txt",sep = "\t",header = TRUE)
# make a column in the table with index index pairs sep by comma ie correct
udi's **for merging with indexes data frame
udi$V1=paste(udi$index.sequence,udi$index.sequence)

# merge the indexes data frame and udi data frame by correct index column
(V1) to get a data frame containing only correct indexed read counts
cindex=merge(indexes,udi,by="V1")
colnames(cindex)=c("sequence pair","counts")

# how many reads are retained for each expected index pair
cindex[,1:2]

##        sequence pair   counts
## 1  AACAGCGA AACAGCGA  6368144
## 2  ACGATCAG ACGATCAG  5933528
## 3  AGAGTCCA AGAGTCCA  7602663
## 4  AGGATAGC AGGATAGC  5861709
## 5  ATCATGCG ATCATGCG  6927867
## 6  ATCGTGGT ATCGTGGT  4730009
## 7  CACTTCAC CACTTCAC  2577666
## 8  CGATCGAT CGATCGAT  4237854
## 9  CGGTAATC CGGTAATC  2393021
## 10 CTAGCTCA CTAGCTCA 13034311
## 11 CTCTGGAT CTCTGGAT 24515042
## 12 GATCAAGG GATCAAGG  4628196
## 13 GATCTTGC GATCTTGC  2636332
## 14 GCTACTCT GCTACTCT  4301318
## 15 GTAGCGTA GTAGCGTA  5774439
## 16 GTCCTAAG GTCCTAAG  6200133
## 17 TACCGGAT TACCGGAT 49686878
## 18 TAGCCATG TAGCCATG  7148153
## 19 TATGGCAC TATGGCAC  7651472
## 20 TCGACAAG TCGACAAG  2644260
## 21 TCGAGAGT TCGAGAGT  7448072
## 22 TCGGATTC TCGGATTC  2874320
## 23 TCTTCGAC TCTTCGAC 30089661
## 24 TGTTCCGT TGTTCCGT 11450554
```

```
# what is the pecentage of reads correctly indexed out of the total number of
sequenced reads
perct=sum(cindex$counts)
(perct/363246735)*100
```

```
## [1] 62.41367
```

```
# what is the percentage of reads correctly indexed out of the total number
of retained reads
totID=(perct/sum(indexes$V2)*100)
totID
```

```
## [1] 99.85423
```

```
# how many indexes were undetermined even after quality filtering ie index
combos we did not use (seq err and such)
ud=read.table("ud.tsv",sep="\t")
sum(ud$V2)
```

```
## [1] 5181567
```

b)

```
library(dplyr)
# how many reads are indicative of index swapping
#make a talble of swapped read and their counts ** huge table so just gonna
head to save space and ill provide the sum
swap=anti_join(indexes, udi, by = "V1")
head(swap)
```

```
##                   V1  V2
## 1 GTAGCGTA CGATCGAT  71
## 2 GTAGCGTA GATCAAGG  94
## 3 GTAGCGTA AACAGCGA 146
## 4 GTAGCGTA TAGCCATG 101
## 5 GTAGCGTA CGGTAATC  58
## 6 GTAGCGTA CTCTGGAT 346
```

```
sum(swap$V2)
```

```
## [1] 330975
```

```
max(swap$V2)
```

```
## [1] 58741
```

c)

```
#plot the distribution of swapped indexes

# bar plot of swapped indexes and counts
plot(swap$V2, xaxt='n', ylim=c(0, 50000), col = "red",main="swapped index
counts",xlab = "swapped
indexes",ylab="counts",cex.main=.1,sub="figure11",type = "h")
```

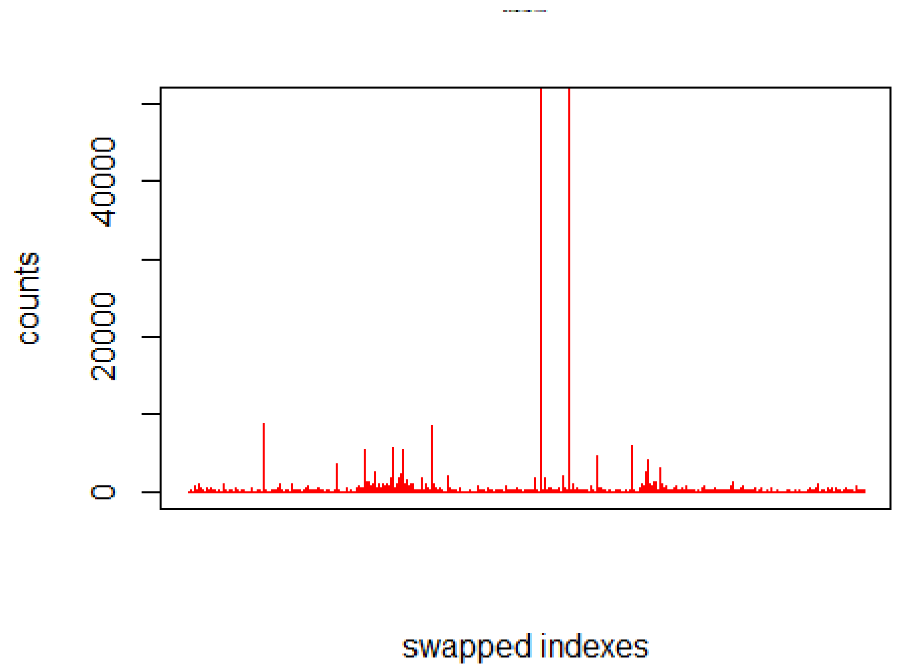


figure11

```
#histogram of swappped index counts and frequency
hist(swap$V2, labels = TRUE, ylim=c(0, 600), col = "thistle", border =
"pink",main="histogram of swapped index frequency",xlab = "swapped index
counts",cex.main=1,sub="figure12")
```

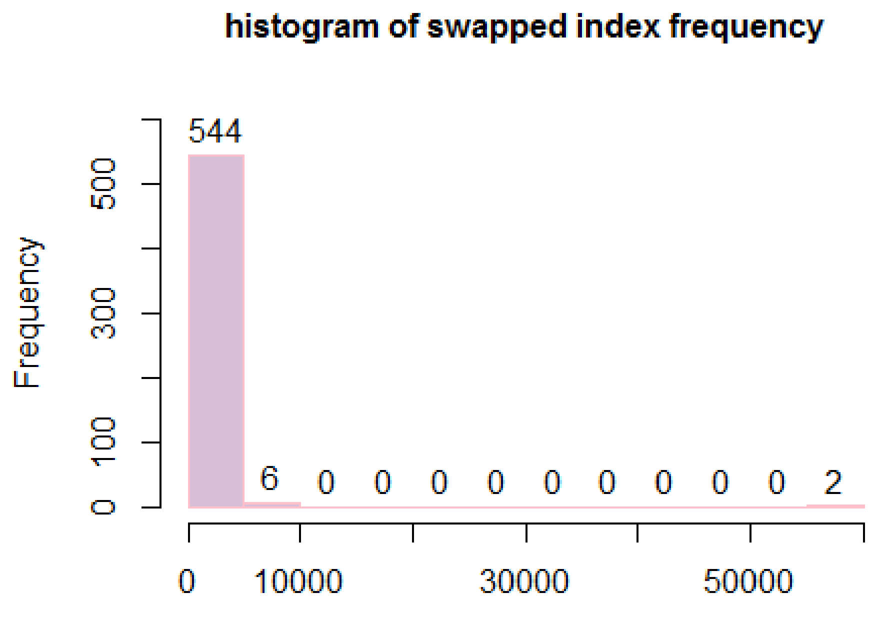


figure12

The above graphs tell me that the counts for swapped indexes for the most part are low 10000 or less except in 2 cases where swapped indexing was seen in 60000 reads. the 2 cases are complementary index swapped ie seq1-seq2, seq2-seq1. This may be due to a over abundance of one or both of the index pairs not bound to sequences in the loaded samples, this would cause them to bind during sequencing where they should not and become over expressed in the end result.

3)

```
#list all you files and the number of reads in each one. this was already
done above but put here for clarity.

# correct udi indexes
head(cindex)

##       sequence pair  counts NA NA      NA  NA       NA
## 1 AACAGCGA AACAGCGA 6368144  4 2C    mbnl  B9 AACAGCGA
## 2 ACGATCAG ACGATCAG 5933528 16 3D    mbnl  A1 ACGATCAG
## 3 AGAGTCCA AGAGTCCA 7602663 32 4G    both B10 AGAGTCCA
## 4 AGGATAGC AGGATAGC 5861709 34 4H    both  A8 AGGATAGC
## 5 ATCATGCG ATCATGCG 6927867 24 4A control  A2 ATCATGCG
## 6 ATCGTGGT ATCGTGGT 4730009 27 4C    mbnl  C2 ATCGTGGT

sum(cindex$counts)

## [1] 226715602
```

```
#index swapped reads
head(swap)

##                    V1  V2
## 1 GTAGCGTA CGATCGAT  71
## 2 GTAGCGTA GATCAAGG  94
## 3 GTAGCGTA AACAGCGA 146
## 4 GTAGCGTA TAGCCATG 101
## 5 GTAGCGTA CGGTAATC  58
## 6 GTAGCGTA CTCTGGAT 346

sum(swap$V2)

## [1] 330975

# undetermined index pairs
head(ud)

##                       V1     V2
## 1 (AACAGCGA, CAACACGA)  12444
## 2 (CTAGCTCA, CCTAGTCA)  22293
## 3 (TACCGGAT, TACCTGAT) 113934
## 4 (TCTTCTAC, TCTTCGAC)  38066
## 5 (ATCGTGGT, ATCGTGTT)  10000
## 6 (CCCCCCCC, GGGGGGGG) 107280

sum(ud$V2)

## [1] 5181567
```