Index_hopping

Dane Dewees

Fri Sep 15 09:45:27 2017

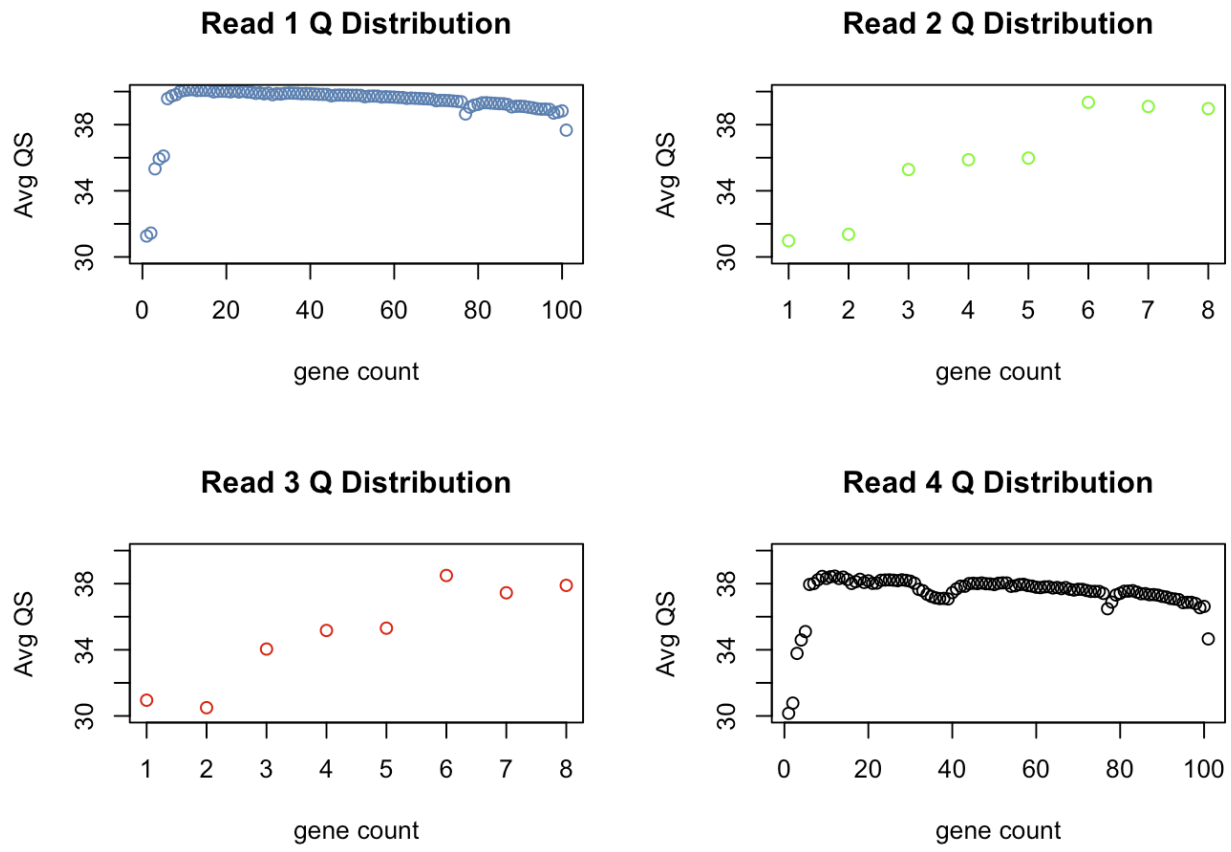
Part 1: Generate per base call distribution of quality scores for read1, read2, index1, and index2. Next, average the Quality scores for each read (for each of the four files) and plot frequency of the Quality Scores.

see part 1 python scripts for both mean QS and frequency distribution on github

```
## function ()
## .Internal(getwd())
## <bytecode: 0x7fd21c4ef2a0>
## <environment: namespace:base>
```

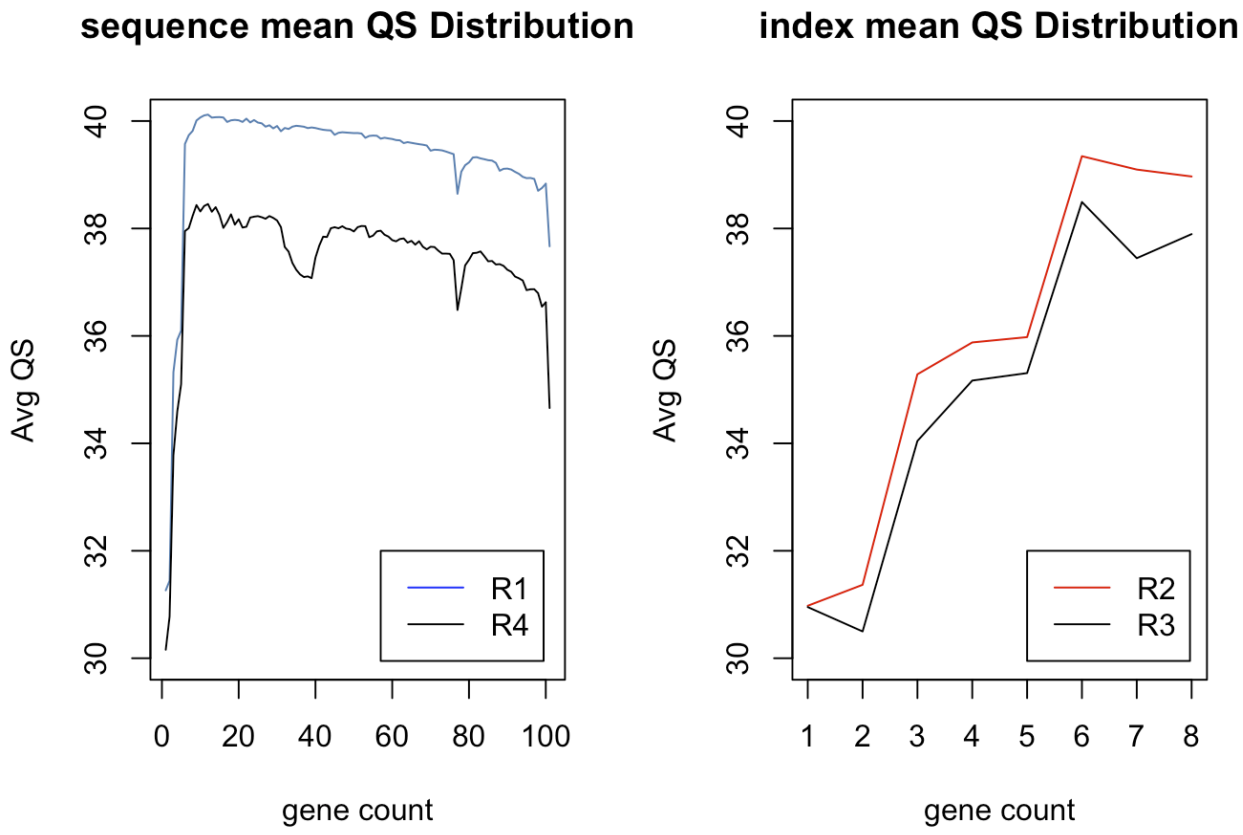
PART 1 plots QS distribution-individual

```
par(mfrow=c(2,2))


plot(index_read_1, main = "Read 1 Q Distribution", ylim = c(30,40), col = 'stee
l blue')
plot(index_read_2, main = "Read 2 Q Distribution", ylim = c(30,40), col = 'gree
n')
plot(index_read_3, main = "Read 3 Q Distribution", ylim = c(30,40), col = 'red'
)
plot(index_read_4, main = "Read 4 Q Distribution", ylim = c(30,40), col = 'blac
k')
```



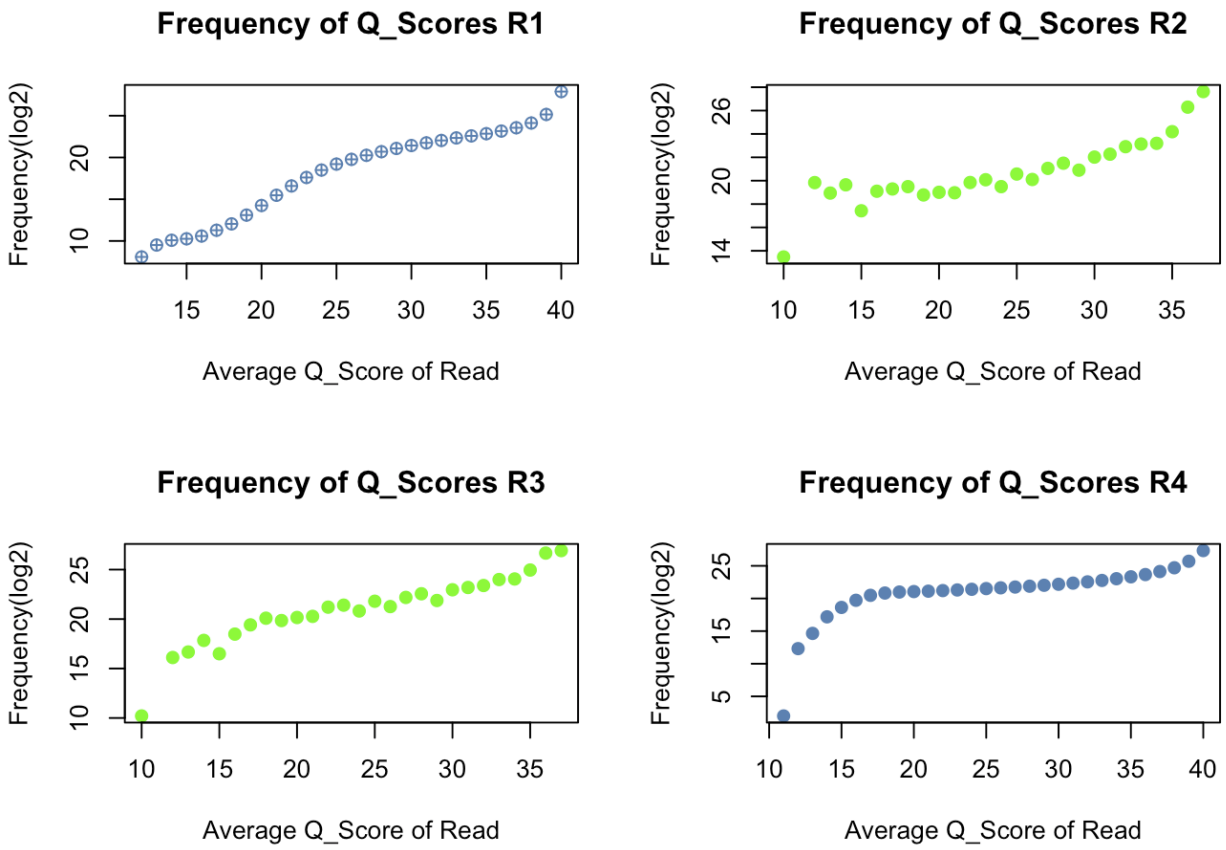
PART 1 plots (Mean quality score R1/R4 & R2/R3)

```
par(mfrow=c(1,2))

plot(index_read_1, main = "sequence mean QS Distribution", pch = 19, type = "l"
, ylim = c(30,40), col = 'steel blue')
points(index_read_4, main = "Read 4 Quality Distribution", type = "l", ylim = c
(30,40), col = 'black')
legend(57, 32, legend=c("R1", "R4"), col=c("blue", "black"), lty=1)


plot(index_read_2, main = "index mean QS Distribution", pch = 19, type = "l", y
lim = c(30,40), col = 'red')
points(index_read_3, main = "Read 4 Quality Distribution", type = "l", ylim = c
(30,40), col = 'black')
legend(5, 32, legend=c("R2", "R3"), col=c("red", "black"), lty=1)
```



Frequency per Base Pair Position plots

```
par(mfrow=c(2,2))

plot(R1_freq_table$Mean_QS, log2(R1_freq_table$Frequency), xlab='Average Q_Scor
e of Read', main='Frequency of Q_Scores R1', pch=10, ylab='Frequency(log2)', co
l = "steel blue")
plot(R2_freq_table$Mean_QS, log2(R2_freq_table$Frequency), xlab='Average Q_Scor
e of Read', main='Frequency of Q_Scores R2', pch=19, ylab='Frequency(log2)', co
l = "green")
plot(R3_freq_table$Mean_QS, log2(R3_freq_table$Frequency), xlab='Average Q_Scor
e of Read', main='Frequency of Q_Scores R3', pch=19, ylab='Frequency(log2)', co
l = "green")
plot(R4_freq_table$Mean_QS, log2(R4_freq_table$Frequency), xlab='Average Q_Scor
e of Read', main='Frequency of Q_Scores R4', pch=19, ylab='Frequency(log2)', co
l = "steel blue")
```



What is a good quality score cutoff for index reads and pairs to utilize for sample identification and downstream analysis, respectively?

Roughly around 30 could be e a good quality score cut off. When viewing the output tables and plots from part 1, you can see a siginficant increase occurance of quality scores over 30. Setting a higher cutoff could potentially lead to loss data. Trimming this value however, could get rid of some bad avg reads when looking at posiitons 1-2.

How many indexes have Undetermined (N) base calls? (Utilize your command line tool knowledge. CHALLENGE: use a one line command)

[daned@ln1 (mailto:daned@ln1) new_index_data]$ cat 1294_S1_L008_R2_001.fastq | awk 'NR %4 == 2' | grep -c "N" 3976613

[daned@ln1 (mailto:daned@ln1) new_index_data]$ cat 1294_S1_L008_R3_001.fastq | awk 'NR %4 == 2' | grep -c "N" 3328051

What do the averaged Quality Scores across the reads tell you? Interpret your data specifically.

The average QS in the majority of the figures is around 39-40 for the base pair positions for reads 1&4. For reads2&3 (index files), the avg QS is between 30-40 (slightly lower) which could be caused by index hopping. This shows that the spread is quite significant for both high and low quality scores.

Write a program to de-multiplex the samples and document index swapping and number of reads retained per sample.

see part_2_idx_hopping script

How many reads are retained for each expected index pair? What is the percentage?

see stats_cov30.tsv and Dual_idx_pairs_cov30.tsv files for raw data - see below for output of index pair with added percent column and total for cov_cutoff of 30

From the data below, you can see that roughly 62% were greater than the cutoff score of 30. Also, you can see infleunces from swapping/sequence error when looking at the total percent of those that did not align to said index pairs (see below).

```
##      Exp_Index_pairs Counts_per_read Percent_captured Percent_total
## 1  GTAGCGTA_GTAGCGTA         5774439         2.486537     1.5896740
## 2  CGATCGAT_CGATCGAT         4237854         1.824867     1.1666599
## 3  GATCAAGG_GATCAAGG         4628196         1.992952     1.2741191
## 4  AACAGCGA_AACAGCGA         6368144         2.742193     1.7531180
## 5  TAGCCATG_TAGCCATG         7148153         3.078074     1.9678506
## 6  CGGTAATC_CGGTAATC         2393021         1.030461     0.6587867
## 7  CTCTGGAT_CTCTGGAT        24515042        10.556447     6.7488678
## 8  TACCGGAT_TACCGGAT        49686878        21.395718    13.6785477
## 9  CTAGCTCA_CTAGCTCA        13034311         5.612718     3.5882803
## 10 CACTTCAC_CACTTCAC         2577666         1.109971     0.7096185
## 11 GCTACTCT_GCTACTCT         4301318         1.852195     1.1841312
## 12 ACGATCAG_ACGATCAG         5933528         2.555043     1.6334704
## 13 TATGGCAC_TATGGCAC         7651472         3.294808     2.1064118
## 14 TGTTCCGT_TGTTCCGT        11450554         4.930735     3.1522800
## 15 GTCCTAAG_GTCCTAAG         6200133         2.669846     1.7068654
## 16 TCGACAAG_TCGACAAG         2644260         1.138648     0.7279515
## 17 TCTTCGAC_TCTTCGAC        30089661        12.956940     8.2835324
## 18 ATCATGCG_ATCATGCG         6927867         2.983216     1.9072070
## 19 ATCGTGGT_ATCGTGGT         4730009         2.036794     1.3021477
## 20 TCGAGAGT_TCGAGAGT         7448072         3.207222     2.0504168
## 21 TCGGATTC_TCGGATTC         2874320         1.237714     0.7912858
## 22 GATCTTGC_GATCTTGC         2636332         1.135234     0.7257689
## 23 AGAGTCCA_AGAGTCCA         7602663         3.273791     2.0929749
```

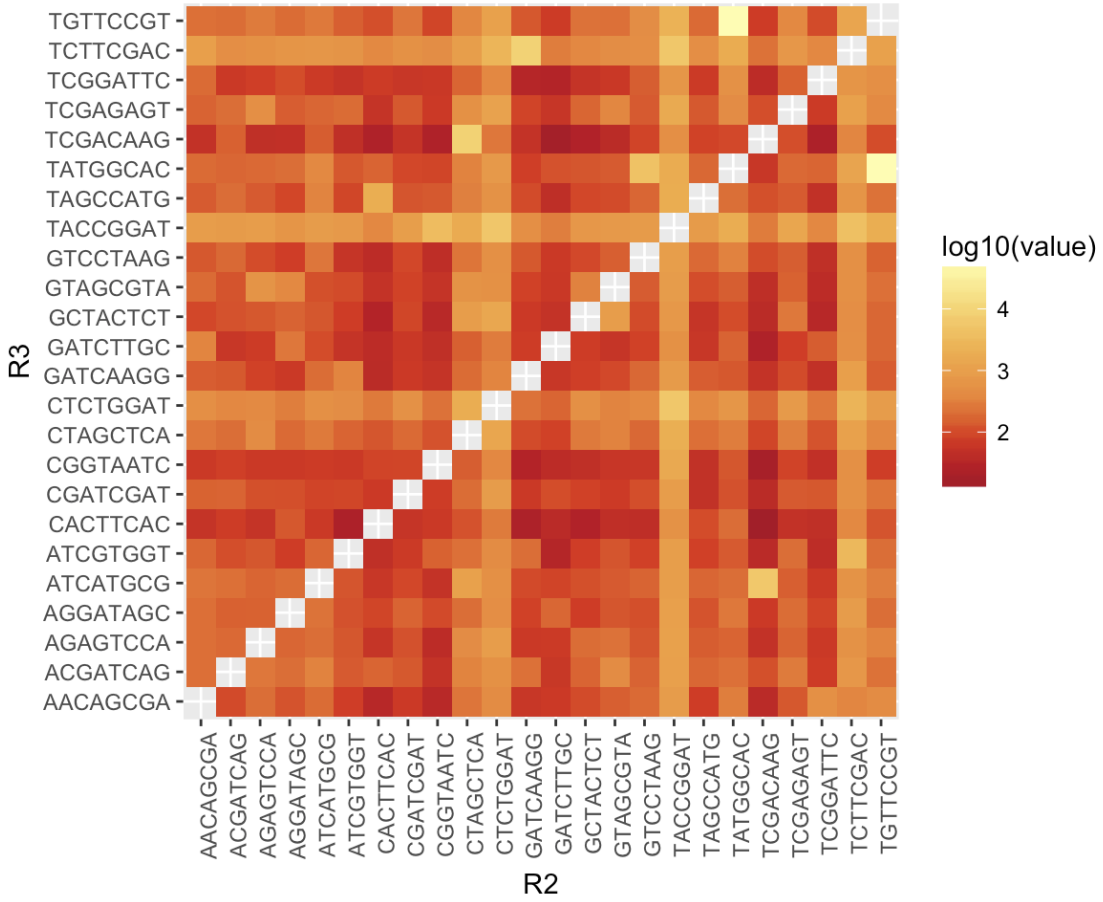
How many reads are indicative of Index Swapping?

330975 for cov_cuttoff of 30 and 179624 for raw/cov_cutoff of 0

Create a distribution of swapped idexes. What does the data tell you?

Heatmap below is for the cov_cutoff of 30

```
## Using R2, R3 as id variables
```



The heatmat shows a wide spread of possible swapped idx combos among the data set of a coverage cutoff of 30 (given out output data). The color indicators from yellow to red shows that brighter yellow –> white is indicative of a 'perfect' match to the counts. Dark red to bright yellow is the spectrum of counts given each index pair. As you can see, there was a significant amount index swapping (this could be through poor preporation of said sample-especially when looking that the indexes TACCGGAT~TACCGGAT). You can also see a consistent trend with the indexes CTTCGAC_TCTTCGAC as far as higher log value. You can reference the table above to see that those two index pairs had a significantly higher percent retained value of sampled reads. 21% for TACCGGAT~TACCGGAT and roughly 13% for TCTTCGAC_TCTTCGAC. This can show the issues with index swapping. Due to the variety in concentration each student had during the preporation of libraries, it could have had already bad coverage to begin with when Maggie began to pool the samples together.