Problem Set 6

Phys 589: Biostatistics

October 17, 2018

Jonathan Monroe

1 Noisy sequences

Opening the *.FASTQ* file reveals sequenced DNA strands organized in the following format:

1. "@(identifier) [optional comments]"
2. Sequence letters
3. "+ [optional description and repeat of sequence]"
4. Quality flags

The quality flags assign an alphanumeric character to quantify the quality, Q. Q is given by the character's ASCII decimal code, e.g. 33 for '!' and 126 (' ') for a total of 94 options. Q corresponds to the occurrence probability, p, via $p = 10^{-(Q-33)/10}$.

Question 1.1. *Which sequence seem noisier?*

Solution. Sequence 3 and 6 contain a number of '/' corresponding to bottom-third quality. More quantitatively, the sum of the quality factors for these sequences seems "statistically lower", as in Figure 1.

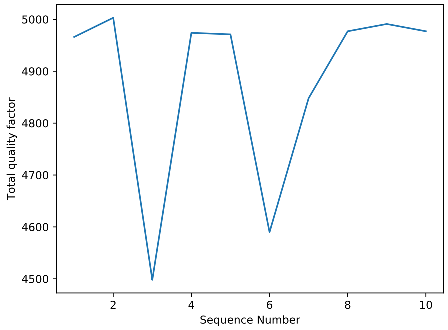

Figure 1:

2 Identification

Searching the National Medical Library's *Basic Local Alignment Search Tool* for the first sequence indicates that it belongs to *Lactobacillus salivarius.* I was surprised that so many other *Lactobacillus* species fit so closely. The score metric maximizes for many of the other fits, and the documentation does not offer a clarifying definition. Nonetheless, because the *ident* field maximizes for *L. salivarius* and not for others I believe this is the represented organism.

3 Difference identification

Here we list the full sequence with highlighted differences:

```
GCAAGCGTTGTCCGGATTTATTGGGCGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCAGAAGAGGAGAGTGGAACTC
GCAAGCGTTGTCCGGATTTATTGGGCGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCGGAAGAGGAGAGTGGAACTC
GCAAGCGTTGTCCGGATTTATTGGGCGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCAGAAGAGGAGAGTGGAACTC
GCAAGCGTTGTCCGGATTTATTGGGCGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCGGAAGAGGAGAGTGGAACTC
GCAAGCGTTGTCCGGATTTATTGGGCGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAAACGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCGGAAGAGGAGAGTGGAACTC
GCAAGCGTTGTCCGGATTTATTGGGAGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCAGAAGAGGAGAGTGGAACTC
GCAAGCGTTGTCCGGATTTATTGGGCGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCAGAAGAGGAGAGTGGAACTC
GCAAGCGTTATCCGGATTTATTGGGCATAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGGAGACTTGAGTGCAGAAGAGGAGAGTGGAACTC
GCAAGCGTTGTCCGGATTTATTGGGCGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCAGAAGAGGAGAGTGGAACTC
GCAAGCGTTGTCCGGATTTATTGGGCGTAAAGGGAACGCAGGCGGTCTTTTAAGTCTGATGTGAAAGCCTTCGGCTTAACCGGAGTAGTG
  CATTGGAAACTGGAAGACTTGAGTGCGGAAGAGGAGAGTGGAACTC
```

We note that 4 of the sequences seem to have the same substitution error and 3 seem to have substitutions in dissimilar positions, whereas 2 others have a close, but unique error.

4 Error sources

The two major sources of substitution errors in 16S data are: the Illumina sequencer itself, and a molecular machine called DNA polymerase, used in a reaction called Polymerase chain reaction (PCR), to selectively amplify the 16S-encoding DNA prior to its sequencing.

5 Error source evaluation

This process of DNA multiplication can include a number of differences among nucleotides. First, biological diversity clearly creates differences in DNA sequences. Second, sequencing errors can also induce nucleotide differences. Of these, errors due to contaminated specimens and nucleotide misidentification can both occur. However, only **artifacts of sequencing** (misidentification) can be flagged by the sequence itself.

6 Errors in sequencing

One can identify which sequences seem to have which type of error by evaluating the read quality of each differing nucleotide. In these comparisons quality flag "E" seems to be the highest achieved accuracy, and I'll assume these nucleotides are read correctly.

(a) No differences

`Seq 3`, `Seq 7`, and `Seq 9` match the reference (`Seq 1`).

(b) Real biological

`Seq 2`, `Seq 4`, and `Seq 10` disagree in nucleotide number 117 with quality flag "E" suggesting an accurate detection of a legitimate biological difference. The repeated substitution in multiple sequences seems to suggest a biological mechanism which produces these mutations.

(c) Artifact of sample prep

Likewise, `Seq 8` contains 3 separate differences from the reference, however, these are all high quality reads. Because these nucleotides are distant from each other, I suspect this sequence comes from an organism which contaminated the sample.

(d) Artifact of sequencing

`Seq 6` disagrees at nucleotide number 26 with quality flag "2". Because this quality score is 19 points lower (on a 94 point scale), I conclude this is likely a sequencer error.

(e) Artifact of sequencing **and** real biological

Finally, `Seq 5` contains the same shift in nucleotide 117 with high quality read, however, it also differs in nucleotide 80. Because the latter read has low quality flag "6" (15 below "E") I conclude that this sequence has both biological diversity and sequencer error.

In the end, these classifications are not air-tight. I've assumed a standard for "perfect sequencing" which is untested, and I have chosen arbitrary quality flag differences. More substantially, I've used plurality as a measure of biological significance which presently has no biological justification.

Perhaps these samples are poorly produced so that the best quality flag is actually fairly mis-representative. Systematic nucleotide mis-identification would be hard to eliminate. To the contrary, the striking disparity between overall quality flag for `Seq 3` and it's agreement with the reference indicate that even low quality reads can give accurate sequences.

7 Eliminating uncertainty

A technical replicate would be a great means to check some of these hypothesis. If the same difference in nucleotide number 117 occurs with comparable frequency then this would a strong case for a biological mechanism. Moreover, repeated "artifact of sequencing" errors would suggest that these are instead real biological differences (especially if differences were recorded higher quality). Unfortunately, "Artifact of sample prep" errors are more difficult to diagnose. If they are systematic issues (e.g. due to a dirty lab environment) contamination could persist. However, if they are sample-specific errors then they would likely be eliminated in a technical replicate.

8 Source Code

Here is my code:

```
import numpy as np
import matplotlib.pyplot as plt

def grouped(in_list, num):
    ''' Returns a list iterated [num] objects at a time'''
    iterable = iter(in_list)
    return zip(*[iterable]*num)
##END grouped

def main():
    ## load data
    data_dir = "/Users/jmonroe/Documents/classes/biostats/hw6/"
    file_name = "demo.txt"
    with open(data_dir+file_name) as open_file:
        read_lines = open_file.readlines()
    ## parse data
    sequence_list = []
    for text_block in grouped(read_lines,4):
        seq = Sequence(*text_block)
        sequence_list.append(seq)
    ## solve problems
    q1_count_errors(sequence_list)
    q3_identify_errors(sequence_list)
    q6_identify_errors(sequence_list)
##END main

def q1_count_errors(sequence_list):
    error_counts = []
    for seq in sequence_list:
        q_sum = sum(seq.quality_ints)
        error_counts.append(q_sum)
        print(seq.seq_num, q_sum)

    plt.plot(np.arange(1,len(error_counts)+1), error_counts)
    plt.xlabel("Sequence Number")
    plt.ylabel("Total quality factor")
    plt.show()
##END q1_count_errors()

def q3_identify_errors(sequence_list):
    standard = sequence_list[0].sequence
    line_length = 90
    print("%standard")
    print(standard[:line_length])
    print("\phantom{loremip}"+standard[line_length:])
    print("%Comparision")
    for Seq in sequence_list[1:]:
        to_compare = Seq.sequence
        highlighted = ’’
        new_line_flag=False
        for i,letter in enumerate(to_compare):
            if i >=line_length and new_line_flag==False:
                highlighted += ’\n’ + "\phantom{loremip}"
                new_line_flag = True
            if letter == standard[i]:
                highlighted += letter
            else:
                highlighted += r’\color{red}’ + letter + r’\color{black}’
        ##END loop through nucleotides
        print()
        print(highlighted)
    ##END loop through (10) sequences
##END q3_count_errors

def q6_identify_errors(sequence_list):
    standard = sequence_list[0].sequence
    for Seq in sequence_list[1:]:
        to_compare = Seq.sequence
        print("Sequence "+str(Seq.seq_num)+":")
        for i,letter in enumerate(to_compare):
            if letter == standard[i]:
                pass
            else:
                q_char = Seq.quality_chars[i]
                q_int = Seq.quality_ints[i]
                print("\t Error @ nucleotide {0}: quality ’{1}’ ({2})"
                      .format(i+1,q_char,q_int))
        ##END loop through nucleotides
    ##END loop through (10) sequences
##END q6_identify_errors

class Sequence():
    def __init__(self,header,sequence, comments, quality):

        self.seq_num = int(header[4:6])
        self.sequence = sequence[:-1] ## remove \n
        self.length = len(sequence)
        self.quality_chars = quality[:-1]
        self.quality_ints = np.array([ord(c) for c in self.quality_chars])
        self.quality_ints -= ord(’!’) ## subtract starting value
        self.quality_ints += 1 ## start from 1.
        self.prob = 10**(self.quality_ints/-10)
    ##END __init__
    def toString(self, do_print=False):
        out_str = ""
        out_str += "Sequence "+ str(self.seq_num) + "\n"
        out_str += self.sequence + "\n"
        out_str += "Quality: " + self.quality_chars
        if (do_print): print(out_str)
    ##END toString
##END Sequence


if __name__ == ’__main__’:
    main()
```