Demultiplexing Summary

Mitchell Rezzonico

October 16, 2018

Demultiplexing Script

The runtime for my script was about 14 hours. This is likely attributed to Python's sluggish rate of gzipping files. It is also a probably a consequence of checking the index lines for each filtering step.

```
#!/usr/bin/env python3
# module load python3/3.6.1
import argparse
import gzip
def get_arguments():
    parser = argparse.ArgumentParser(description="Allows users to know how to navigate this program")
    parser.add_argument("-r1", "--read1", help="Forward Read File", required=True, type=str)
    parser.add_argument("-i1", "--index1", help="Forward Index File", required=True, type=str)
    parser.add_argument("-i2", "--index2", help="Reverse Index File", required=True, type=str)
    parser.add_argument("-r2", "--read2", help="Reverse Read File", required=True, type=str)
    parser.add_argument("-qs", "--quality_score", help="Cutoff for quality score", required=True, type=int)
    parser.add_argument("-I", "--Index_library", help="Input index library. Indexes must be in the 5th column", required=True, t
    return parser.parse_args()
################################
###### Argparse Variables ######
################################
args = get_arguments()
r1 = args.read1
i1 = args.index1
i2 = args.index2
r2 = args.read2
qscore = args.quality_score
indexes = args.Index_library
################################
#### Higher order functions ####
################################
def reverse_complement(seq):
   """Returns the reverse complement of a DNA sequence"""
   complement = {'A': 'T', 'C': 'G', 'G': 'C', 'T': 'A', 'N':'N'}
   return ''.join([complement[base] for base in seq[::-1]])
def convert_phred(letter):
    """Converts a single character into a phred score"""
    return(ord(letter)-33)
################################
### Isolate Indexes from file ###
################################
index_list = []
with open(indexes) as i:
    for col in i:
        col = col.strip("\n").split("\t")
        index_list.append(col[4])
index_list = index_list[1:] # remove the header line
################################
# Open and Create Output Files #
################################
ctr_dict = {}
F_index_dict = {}
R_index_dict = {}
for index in index_list:
    Forward = gzip.open(index + "_forward.gz","wt")
    Reverse = gzip.open(index + "_reverse.gz","wt")
    F_index_dict[index] = Forward
    R_index_dict[index] = Reverse
    ctr_dict[index] = 0
bad_ctr = 0
LQ_Fz = gzip.open("bad_reads_forward.gz", "wt")
```

```
LQ_Rz = gzip.open("bad_reads_reverse.gz", "wt")
################################
### Bad Read Filter Counters ###
################################
undetermined = 0
low_qscore = 0
false_read = 0
index_hopping = 0
################################
# Open Input Files, Sort Reads #
################################
total_count = 0
with gzip.open(r1, "rt") as r1, gzip.open(i1, "rt") as i1, gzip.open(i2, "rt") as i2, gzip.open(r2, "rt") as r2:
    while True: # keep looping until the break

        r_header1 = r1.readline()
        r_seq1 = r1.readline()
        r_plus1 = r1.readline().strip()
        r_qscore1 = r1.readline()

        # set condition for when the loop finally ends so that there isn't an error
        if r_header1 == '':
            break

        total_count += 1

        i_header1 = i1.readline()
        i_seq1= i1.readline()
        i_plus1 = i1.readline()
        i_qscore1 = i1.readline()

        i_header2 = i2.readline()
        i_seq2 = i2.readline()
        i_plus2 = i2.readline()
        i_qscore2 = i2.readline()

        r_header2 = r2.readline()
        r_seq2 = r2.readline()
        r_plus2 = r2.readline().strip()
        r_qscore2 = r2.readline()

        isBad = False

        # filter out indexes with undetermined nucleotides
        if "N" in i_seq1 or "N" in i_seq2:
            isBad = True
            undetermined += 1
            r_plus1 += "Undetermined | "
            r_plus2 += "Undetermined | "

        # calculate the mean qscore of the index sequence
        converted_NT = []
        for NT in i_qscore1:
            converted_NT.append(convert_phred(NT))
        converted_NT_mean = sum(converted_NT) / len(converted_NT)

        # filter out indexes with a low mean qscore
        if converted_NT_mean < qscore:
            isBad = True
            low_qscore += 1
            r_plus1 += "Low Quality | "
            r_plus2 += "Low Quality | "

        # filter out indexes that do not appear in the index library
        if i_seq1.strip() not in index_list and reverse_complement(i_seq2.strip()) not in index_list:
            isBad = True
            false_read += 1
```

```
            r_plus1 += "Index not recognized | "
            r_plus2 += "Index not recognized | "

        # filter and track occurrences of index hopping
        else:
            if reverse_complement(i_seq1.strip()) != i_seq2.strip():
                isBad = True
                index_hopping += 1
                r_plus1 += "Index hopping | "
                r_plus2 += "Index hopping | "

        # build final bad output files
        if isBad:
            LQ_Fz.write(r_header1.strip()+':'+i_seq1.strip()+'\n')
            LQ_Fz.write(r_seq1)
            LQ_Fz.write(r_plus1 + '\n')
            LQ_Fz.write(r_qscore1)

            LQ_Rz.write(r_header2.strip()+':'+i_seq2.strip()+'\n')
            LQ_Rz.write(r_seq2)
            LQ_Rz.write(r_plus2 + '\n')
            LQ_Rz.write(r_qscore2)

            bad_ctr += 1

        # once all the bad reads have been filtered out, write good reads to their respective files
        if isBad == False:

            F_index_dict[i_seq1.strip()].write(r_header1.strip()+':'+i_seq1.strip()+'\n')
            F_index_dict[i_seq1.strip()].write(r_seq1)
            F_index_dict[i_seq1.strip()].write(r_plus1 + '\n')
            F_index_dict[i_seq1.strip()].write(r_qscore1)

            R_index_dict[i_seq1.strip()].write(r_header2.strip()+':'+i_seq2.strip()+'\n')
            R_index_dict[i_seq1.strip()].write(r_seq2)
            R_index_dict[i_seq1.strip()].write(r_plus2 + '\n')
            R_index_dict[i_seq1.strip()].write(r_qscore2)

            ctr_dict[i_seq1.strip()] += 1
################################
###### Close Output Files ######
################################

for index in index_list:
    F_index_dict[index].close()
    R_index_dict[index].close()
LQ_Fz.close()
LQ_Rz.close()
################################
## Relevant Output Statements ##
################################
print("INDEX", "\t", "RAW COUNT", "\t", "PROPORTION (%)")
for k in ctr_dict:
    print(k, "\t", ctr_dict[k], "\t", (ctr_dict[k]/total_count) * 100)

bad_proportion = (bad_ctr / total_count) * 100
print("Bad reads compose {}% of the entire file.".format(bad_proportion))
print("There are {} instances of indexes with undetermined NTs in their sequences.".format(undetermined))
print("There are {} instances of indexes with a mean phred quality score below {}.".format(low_qscore, qscore))
print("There are {} instances of indexes that are not included in the list of indexes.".format(false_read))
print("There are {} instances of index hoppping.".format(index_hopping))
```

Sbatch script submitted to Talapas

```
#!/usr/bin/env bash
```

```
#SBATCH --partition=long        ### Partition (like a queue in PBS)
#SBATCH --job-name=Demult          ### Job Name
#SBATCH --time=1-8:00:00       ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1               ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=28     ### Number of tasks to be launched per Node
#SBATCH --mail-type=ALL
#SBATCH --mail-user=mrezzoni@uoregon.edu

R1="/projects/bgmp/shared/2017_sequencing/1294_S1_L008_R1_001.fastq.gz"
I1="/projects/bgmp/shared/2017_sequencing/1294_S1_L008_R2_001.fastq.gz"
I2="/projects/bgmp/shared/2017_sequencing/1294_S1_L008_R3_001.fastq.gz"
R2="/projects/bgmp/shared/2017_sequencing/1294_S1_L008_R4_001.fastq.gz"
I_table="/projects/bgmp/mrezzoni/Bi624/HW2/indexes.txt"

./demultiplexing_script.py -r1 $R1 -i1 $I1 -i2 $I2 -r2 $R2 -qs 30 -I $I_table
```

Test files

```
############## Altered Index List ##############

# sample    group   treatment   index   index sequence
# 1 2A  control B1  AAAAAAAA
# 2 2B  control A5  CGATCGAT
# 3 2B  control C1  GATCAAGG
# 4 2C  mbnl    B9  AACAGCGA
# 6 2D  mbnl    C9  TAGCCATG
# 7 2E  fox C3  CGGTAATC
# 8 2F  fox B3  CTCTGGAT
# 10    2G  both    C4  TACCGGAT
# 11    2H  both    A11 CTAGCTCA
# 14    3B  control C7  CACTTCAC
# 15    3C  mbnl    B2  GCTACTCT
# 16    3D  mbnl    A1  ACGATCAG
# 17    3E  fox B7  TATGGCAC
# 19    3F  fox A3  TGTTCCGT
# 21    3G  both    B4  GTCCTAAG
# 22    3H  both    A12 TCGACAAG
# 23    4A  control C10 TCTTCGAC
# 24    4A  control A2  ATCATGCG
# 27    4C  mbnl    C2  ATCGTGGT
# 28    4D  mbnl    A10 TCGAGAGT
# 29    4E  fox B8  TCGGATTC
# 31    4F  fox A7  GATCTTGC
# 32    4G  both    B10 AGAGTCCA
# 34    4H  both    A8  AGGATAGC


############## Test Read 1 ##############

# @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
# GNCTGGCATTCCCAGAGACATCAGTACCCAGTTGGTTCAGACAGTTCCTCTATTGGTTGACAAGGTCTTCATTTCTAGTGATATCAACACGGTGTCTACAA
# +
# A#A-<FJJJ<JJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F--JJJJJJF-A-F7JJJJ
# @K00337:83:HJKJNBBXX:8:1101:1286:1191 1:N:0:1
# CNACCTGTCCCCAGCTCACAGGACAGCACACCAAAGGCGGCAACCCACACCCAGTTTTACAGCCACACAGTGCCTTGTTTTACTTGAGGACCCCCCCACTCC
# +
# A#AAFJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJAJJJJJJJJJJJJJJFJJJJJFFFFJJJJJJJJJJJJJJJJ77F
# @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1
# GNGGTCTTCTACCTTTCTCTTCTTTTTTGGAGGAGTAGAATGTTGAGAGTCAGCAGTAGCCTCATCATCACTAGATGGCATTTCTTCTGAGCAAAACAGGT
# +
# A#AFFFJFJJFJJJJFJJJJJJJJAJJFJJJJJFJFJ7<FAFJJFJFJJFJFJJJFJAAJJJFJJJJJJJJJJJJJJAJJJFAJJJJJFFJJJAJJJ<F-
# @K00337:83:HJKJNBBXX:8:1101:1367:1191 1:N:0:1
# GNGCTCTTCCCCACACCATTGGGACCCACGATGCAAATCCGGGAGTCCATGTCGATGCCGAAATCTAGATTCTTAAAGAGTGGCTTCTGCCCCTCGTAGCC
# +
# A#<AAFJFJJJJFJJFJJ7JFJJJFJFAJJ<FF<<JJ<JJ<F<JJFAJJFFFJJJJJJA--77FJ--<<-AA<<AFJJJJJJFJJJFFFJ-<7--7-FFFA
# @K00337:83:HJKJNBBXX:8:1101:1407:1191 1:N:0:1
```

```
# TNCTAAAATAAATAAAATCCCTAGGTTTAAATTAATTAATGGGTGTGGTATTGGTAGGGGAACTCATAGACTTAATGCTAGTGTGAGTGATAGGGTAGGTG
# +
# A#A-FJJJJJJJJJJJJJJJJJJJJJJJJJJJJJFJJFJFJFJJJJJJJJJJFFJJJJJJFJJJJJJJJJJJJJJJJJFJ<FJJAFJJJF<J7FJJJF


############### Test Index 1 ###############

# @K00337:83:HJKJNBBXX:8:1101:1265:1191 2:N:0:1
# AAAAAAAA
# +
# #AA<FJJJ
# @K00337:83:HJKJNBBXX:8:1101:1286:1191 2:N:0:1
# AAAAAAAA
# +
# #AAAFJJJ
# @K00337:83:HJKJNBBXX:8:1101:1347:1191 2:N:0:1
# AAAAAAAA
# +
# #AAFFJAJ
# @K00337:83:HJKJNBBXX:8:1101:1367:1191 2:N:0:1
# AAAAAAAN
# +
# #AAAFJJ<
# @K00337:83:HJKJNBBXX:8:1101:1407:1191 2:N:0:1
# AAAAAAAA
# +
# #AAFFJJF


############### Test Index 2 ###############

# @K00337:83:HJKJNBBXX:8:1101:1265:1191 2:N:0:1
# TTTTTTTT
# +
# #AA<FJJJ
# @K00337:83:HJKJNBBXX:8:1101:1286:1191 2:N:0:1
# TTTTTTTT
# +
# #AAAFJJJ
# @K00337:83:HJKJNBBXX:8:1101:1347:1191 2:N:0:1
# TTTTTTTT
# +
# #AAFFJAJ
# @K00337:83:HJKJNBBXX:8:1101:1367:1191 2:N:0:1
# TTTTTTTT
# +
# #AAAFJJ<
# @K00337:83:HJKJNBBXX:8:1101:1407:1191 2:N:0:1
# TTTTTTTT
# +
# #AAFFJJF


############### Test Read 2 ###############

# @K00337:83:HJKJNBBXX:8:1101:1265:1191 4:N:0:1
# NTTTTGATTTACCTTTCAGCCAATGAGAAGGCCGTTCATGCAGACTTTTTTAATGATTTTGAAGACCTTTTTGATGATGATGATGTCCAGTGAGGCCTCCC
# +
# #AAFAFJJ-----F---7-<FA-F<AFFA-JJJ77<FJFJFJJJJJJJJJJAFJFFAJJJJJJJJFJF7-AFFJJ7F7JFJJFJ7FFF--A<A7<-A-7--
# @K00337:83:HJKJNBBXX:8:1101:1286:1191 4:N:0:1
# NTGTGTAGACAAAAGTTTTCATGAGTCTGTAAGCTGTCTATTGTCTCCTGAAAAGAAACCAGAAGTTTTCCCCTAAATGTGTTTAGAATGCTTATTCTAAT
# +
# #A-AFFJJFJJJJJJJJJJJJJJJ<JAJFJJJJF<JFJJJAJJJJJJJJJJJJJJJJJJFJJJAJJFJJJFJJJF<JJA-JJJ-<AFAF--FF<JAFJF
# @K00337:83:HJKJNBBXX:8:1101:1347:1191 4:N:0:1
# NAAATGCCATCTAGTGATGATGAGGCTACTGCTGACTCTCAACATTCTACTCCTCCAAAAAAGAAGAGAAAGATTCCAACCCCCAGAACCGATGACCGGCA
# +
# #AAFFJJJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJFJJJJJJJJJJJFJJFJJJFJFJAJAF<7AJF<J--7AA7<FJ----7A-F-77-77------
# @K00337:83:HJKJNBBXX:8:1101:1367:1191 4:N:0:1
```

```
# NAGAAATTCAAGGTGCTTAAGGTTGGGGGCAATTCAACAAAGCAATAGGAAAAGCAAATAAAGGAAGGCCTGACTCGAAAACATCAGAATCATGTATGGAA
# +
# #A-7AF----<7---77--7<<<F-7---77F--77-<--7--<-----77<--<-<<<<<-7-7-7-F<<A-7-F-A-7-7------7-----7--77-7
# @K00337:83:HJKJNBBXX:8:1101:1407:1191 4:N:0:1
# NTAAGACTTAAAACCTTGTTCCCAGAGGTTCAAATCCTCTCCCTAATAGTGTTCTTTATTAATATCCTAACACTCCTCGTCCCCATTCTAATCGCCATACC
# +
# #AAFFJJJJJJJJJJJJJJJJJJFJJJF7FFJJFF7AJJFJJJJJJJJJJFJJJJJJJJJJJJFJJJJJJJJJFJJJJJJJFFFJFF7AFJF7FFJAJ-F
```

Output from test files

```
# INDEX      RAW COUNT   PROPORTION (%)
# AAAAAAAA   1   20.0
# CGATCGAT   0   0.0
# GATCAAGG   0   0.0
# AACAGCGA   0   0.0
# TAGCCATG   0   0.0
# CGGTAATC   0   0.0
# CTCTGGAT   0   0.0
# TACCGGAT   0   0.0
# CTAGCTCA   0   0.0
# CACTTCAC   0   0.0
# GCTACTCT   0   0.0
# ACGATCAG   0   0.0
# TATGGCAC   0   0.0
# TGTTCCGT   0   0.0
# GTCCTAAG   0   0.0
# TCGACAAG   0   0.0
# TCTTCGAC   0   0.0
# ATCATGCG   0   0.0
# ATCGTGGT   0   0.0
# TCGAGAGT   0   0.0
# TCGGATTC   0   0.0
# GATCTTGC   0   0.0
# AGAGTCCA   0   0.0
# AGGATAGC   0   0.0
# Bad reads compose 80.0% of the entire file.
# There are 1 instances of indexes with undetermined NTs in their sequences.
# There are 4 instances of indexes with a mean phred quality score below 30.
# There are 0 instances of indexes that are not included in the list of indexes.
# There are 1 instances of index hoppping.
```

Output from assigned samples

```
# INDEX      RAW COUNT   PROPORTION (%)
# GTAGCGTA   6423487     1.768353678388878
# CGATCGAT   4370218     1.203099045060928
# GATCAAGG   4877454     1.3427385658401032
# AACAGCGA   7247905     1.995311809203185 4
# TAGCCATG   8048298     2.2156559782980567
# CGGTAATC   3693609     1.0168319888683928
# CTCTGGAT   28060203    7.7248328192130895
# TACCGGAT   61542468    16.942332048765696
# CTAGCTCA   13196603    3.632958462792515
# CACTTCAC   3197050     0.880131792512877 7
# GCTACTCT   5162621     1.4212436073238208
# ACGATCAG   6186925     1.7032293490538877
# TATGGCAC   8622836     2.3738234013307786
# TGTTCCGT   10905086    3.0021153528055797
# GTCCTAAG   6461033     1.7786899034343697
# TCGACAAG   2947217     0.811354023595009
# TCTTCGAC   34754032    9.56761029111521
# ATCATGCG   7264829     1.999970901321384
# ATCGTGGT   4814723     1.3254690369068285
# TCGAGAGT   5608742     1.544058475845626
```

```
# TCGGATTC   2840779     0.7820521772893567
# GATCTTGC   2773588     0.7635548327777811
# AGAGTCCA   7709563     2.122403935716036
# AGGATAGC   6771157     1.8640654815520914
# Bad reads compose 30.218113040988513% of the entire file.
# There are 4205183 instances of indexes with undetermined NTs in their sequences.
# There are 99116240 instances of indexes with a mean phred quality score below 30.
# There are 7867981 instances of indexes that are not included in the list of indexes.
# There are 23623721 instances of index hoppping.
```

Files created after script completed

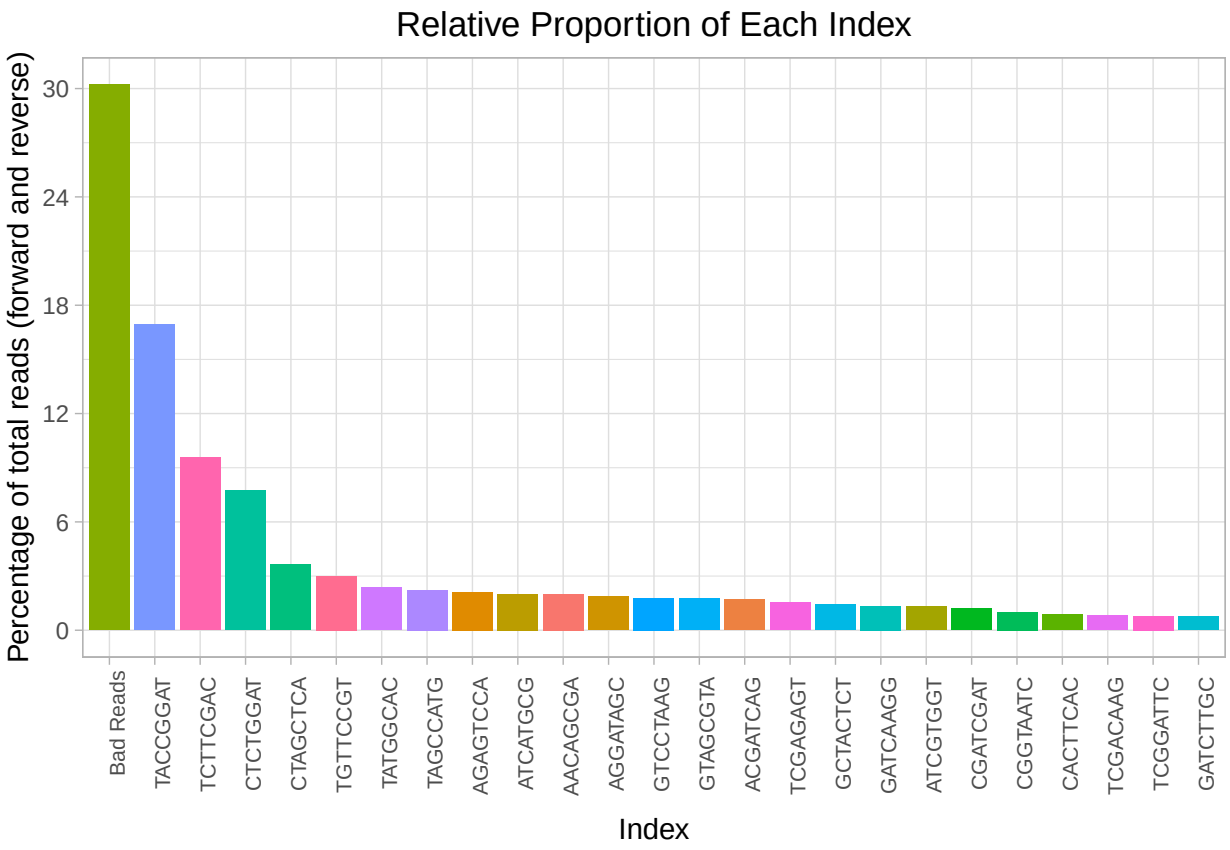
```
[mrezzoni@talapas-ln1 zipped_output]$ ls
AACAGCGA_forward.gz  ATCATGCG_forward.gz   CGATCGAT_forward.gz  GATCAAGG_forward.gz  GTCCTAAG_forward.gz  TCGACAAG_forward.gz  TGTTCCGT_forward.gz
AACAGCGA_reverse.gz  ATCATGCG_reverse.gz   CGATCGAT_reverse.gz  GATCAAGG_reverse.gz  GTCCTAAG_reverse.gz  TCGACAAG_reverse.gz  TGTTCCGT_reverse.gz
ACGATCAG_forward.gz  ATCGTGGT_forward.gz   CGGTAATC_forward.gz  GATCTTGC_forward.gz  TACCGGAT_forward.gz  TCGAGAGT_forward.gz
ACGATCAG_reverse.gz  ATCGTGGT_reverse.gz   CGGTAATC_reverse.gz  GATCTTGC_reverse.gz  TACCGGAT_reverse.gz  TCGAGAGT_reverse.gz
AGAGTCCA_forward.gz  bad_reads_forward.gz  CTAGCTCA_forward.gz  GCTACTCT_forward.gz  TAGCCATG_forward.gz  TCGGATTC_forward.gz
AGAGTCCA_reverse.gz  bad_reads_reverse.gz  CTAGCTCA_reverse.gz  GCTACTCT_reverse.gz  TAGCCATG_reverse.gz  TCGGATTC_reverse.gz
AGGATAGC_forward.gz  CACTTCAC_forward.gz   CTCTGGAT_forward.gz  GTAGCGTA_forward.gz  TATGGCAC_forward.gz  TCTTCGAC_forward.gz
AGGATAGC_reverse.gz  CACTTCAC_reverse.gz   CTCTGGAT_reverse.gz  GTAGCGTA_reverse.gz  TATGGCAC_reverse.gz  TCTTCGAC_reverse.gz
[mrezzoni@talapas-ln1 zipped_output]$
```

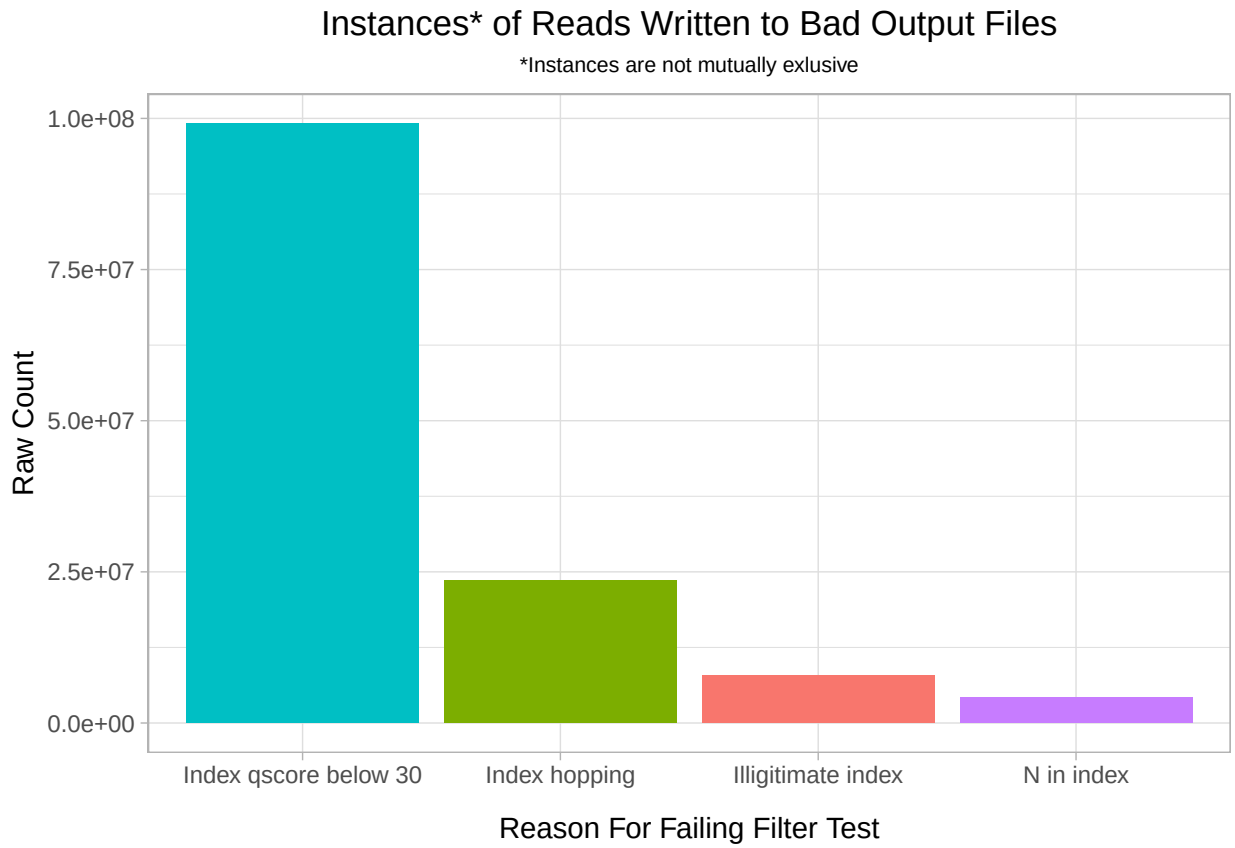
Head -8 output for AACAGCGA and bad reads

```
############### AACAGCGA Forward ###############

# @K00337:83:HJKJNBBXX:8:1101:1479:1701 1:N:0:1:AACAGCGA
# CCTTTGCTTAATCTGCTCTTGTCTTTCAGCACTAAGCTGTTCTTTTTCTTCTTTCATGGCTGAAATTTTTGTTTTCAGTTCTCTAACCTGGCGCTCAATAT
# +
# AAAAAJJFJJJJJJJJJJJFJJJJFJJJJJJFJJJJJFFJJJJJJJJJJJJJJFJJJJJJJAFJJJJJJJJJJJJFF<JJJJJJJJJFJFJJJJFJJJ
# @K00337:83:HJKJNBBXX:8:1101:1966:1701 1:N:0:1:AACAGCGA
# ATGTTCAGTTGTCTCTGGACTTACAAGTTCCTCAGTACACTGCTCCAGAGCCTCCCTCTCTTCATCCATGTCCTGACTGGTCTCCTTGCTGGCTTCTGAGC
# +
# AAFAFJJJJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ

############### AACAGCGA Reverse ###############

# @K00337:83:HJKJNBBXX:8:1101:1479:1701 4:N:0:1:TCGCTGTT
# CAACGAGACGCGCGCTAAGCTCGACGAGCTTTCTGCTAAGCGAGAAACGAGTGGAGAGAAATCCAGACAACTAAGGGATGCCCAGCAGGATGCAAGAGACA
# +
# A<AFFJJJJJJJJJJJJJJAJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJ7JJJJJJJJJJJJJJJJJJFJJJJJJJJJ
# @K00337:83:HJKJNBBXX:8:1101:1966:1701 4:N:0:1:TCGCTGTT
# CTTGTGAGAACTGAACCACGAGAGCAGCTCCTTGATCAGCTGCAAAAGAAACAACCACCAATGATGCTAAACAGCTCAGAAGCCAGCAAGGAGACCAGTCA
# +
# AAFFFJJJAJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ


############### Bad Forward ###############

# @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:NCTTCGAC
# GNCTGGCATTCCCAGAGACATCAGTACCCAGTTGGTTCAGACAGTTCCTCTATTGGTTGACAAGGTCTTCATTTCTAGTGATATCAACACGGTGTCTACAA
# +Undetermined | Low Quality | Index not recognized |
# A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F--JJJJJJF-A-F7JJJJ
# @K00337:83:HJKJNBBXX:8:1101:1286:1191 1:N:0:1:NACAGCGA
# CNACCTGTCCCCAGCTCACAGGACAGCACACCAAAGGCGGCAACCCACACCCAGTTTACAGCCACACAGTGCCTTGTTTTACTTGAGGACCCCCCCACTCC
# +Undetermined | Low Quality | Index not recognized |
# A#AAFJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJAJJJJJJJJJJJJJJFJJJJJFFFFJJJJJJJJJJJJJJJJ77F

############### Bad Reverse ###############

# @K00337:83:HJKJNBBXX:8:1101:1265:1191 4:N:0:1:NTCGAAGA
# NTTTTGATTTACCTTTCAGCCAATGAGAAGGCCGTTCATGCAGACTTTTTTAATGATTTTGAAGACCTTTTTGATGATGATGATGTCCAGTGAGGCCTCCC
# +Undetermined | Low Quality | Index not recognized |
# #AAFAFJJ-----F---7-<FA-F<AFFA-JJJ77<FJFJFJJJJJJJJJJAFJFFAJJJJJJJJFJF7-AFFJJ7F7JFJJFJ7FFF--A<A7<-A-7--
# @K00337:83:HJKJNBBXX:8:1101:1286:1191 4:N:0:1:NCGCTGTT
# NTGTGTAGACAAAAGTTTTCATGAGTCTGTAAGCTGTCTATTGTCTCCTGAAAAGAAACCAGAAGTTTTCCCCTAAATGTGTTTAGAATGCTTATTCTAAT
```

```
# +Undetermined | Low Quality | Index not recognized |
# #A-AFFJJFJJJJJJJJJJJJJJJJ<JAJFJJJJF<JFJJJAJJJJJJJJJJJJJJJJJJFJJJAJJFJJJFJJJF<JJA-JJJ-<AFAF--FF<JAFJF
```

Output plots




Instances* of Reads Written to Bad Output Files
*Instances are not mutually exlusive
1.0e+08
7.5e+07
5.0e+07
2.5e+07
0.0e+00
Raw Count
Index qscore below 30
Index hopping
Illigitimate index
N in index
Reason For Failing Filter Test