Bi624_demultiplexing

October 16, 2018

```
In [ ]: #Test file Index 1
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Good quality score
        CCTTCGAC
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Low quality score
        CCTTCGAC
        +
        ########
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with no Ns
        TTTTTTTT
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Ns
        NTTTTTTT
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key in dictionary
        AGAGTCCA
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key in dictionary
        AGAGTCCA
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key not in dictionary
        GGGGGGGG
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key not in dictionary
        CCCCCCCC
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key in dictionary and match
        AGGATAGC
        +
        JJJJJJJJ
```

```
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key in dictionary and match
AGGATAGC
+
JJJJJJJJ
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key in dictionary, match, bad
AGGATAGC
+
########
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key in dictionary and match,
TACCGGAT
+
JJJJJJJJ
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key in dictionary and dont ma
TACCGGAT
+
JJJJJJJJ
```

```
In [ ]: #Test file Index 1
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read with low quality score
        CCTTCGAC
        +
        ########
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read with good quality score
        CCTTCGAC
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read with N
        NTTTTTTT
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read with no N
        TTTTTTTT
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read not in dictionary
        GGGGGGGG
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read not in dictionary
        TTTTTTTT
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read in dictionary
        TACGCTAC
        +
        JJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read  in dictionary
        TACGCTAC
```

```
+
JJJJJJJJ
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read  in dictionary and match
GCTATCCT
+
JJJJJJJJ
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read  in dictionary and match
GCTATCCT
+
JJJJJJJJ
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-read  in dictionary, match, bad QS
GCTATCCT
+
########
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key not in dictionary and matc
ATCCGGTA
+
JJJJJJJJ
@K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1-reads with Key in dictionary and dont mat
CGCATGAT
+
JJJJJJJJ
```

```
In [ ]: #Test file R1
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
        1)Good quality score
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
        2)Bad quality score
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
        3)No Ns
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
        4)Has Ns
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
        5)Key in dictionary
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
        6)Key in dictionary
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

```
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
7)Key not in dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
8)Key not in dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
9)Key in dictionary and matches
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
10)Key in dictionary and matches
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
11)Key in dictionary and matches, bad quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
12)Key in dictionary and matches, Good quality score, and match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
13)Key in dictionary, Good quality score, and dont match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

In [ ]:
```
#Test file R2
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
1)Bad quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
2)Good quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
3)Has Ns
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
4)No Ns
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
5)Key not in Dictionary
```

```
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
6)Key not in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
7)Key in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
8)Key in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
9)Key in dictionary and matches
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
10)Key in dictionary and matches
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
11)Key in dictionary and matches, bad quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
12)Key in dictionary and matches, Good quality score, and match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1
13)Key in dictionary, Good quality score, and dont match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

In [ ]:
```
#Barcode file, tab sperated
B1          GTAGCGTA
A5          CGATCGAT
C1          GATCAAGG
B9          AACAGCGA
C9          TAGCCATG
C3          CGGTAATC
B3          CTCTGGAT
C4          TACCGGAT
A11          CTAGCTCA
C7          CACTTCAC
B2          GCTACTCT
A1          ACGATCAG
```

```
B7        TATGGCAC
A3        TGTTCCGT
B4        GTCCTAAG
A12        TCGACAAG
C10        TCTTCGAC
A2        ATCATGCG
C2        ATCGTGGT
A10        TCGAGAGT
B8        TCGGATTC
A7        GATCTTGC
B10        AGAGTCCA
A8        AGGATAGC
```

In [8]:
```python
#Demultiplexing Python Assignment
#######################################################################################
##Making Dictionary for Barcodes and output files

#dictionary for R1 barcodes for making output file
file_dict_r1={}
#dictionary for R2 barcodes for making output files
file_dict_r2={}
#file that contains the barcodes and samples
barcodes_file="/Users/mandiedriskill/Bi624_Assignments/demultiplexing/Barcodes.txt"
#dictionary for checking if index matches
barcode_dict = {}
#Reading each line in the barcode file
for line in open(barcodes_file).readlines():
    #striping the lines and spliting in the barcode file
    sample, barcode = line.strip().split()
    #setting the barcodes keys and the samples as values
    barcode_dict[barcode] = sample
    #creating R1 files that are named with all the barcodes in the dictionary
    file_dict_r1[barcode]=open(barcode + "_R1.fq", "w")
    #creating R2 files that are named with all the barcodes in the dictionary
    file_dict_r2[barcode]=open(barcode + "_R2.fq", "w")


#######################################################################################
##Creating Functions

#making a function called convert_phred
def convert_phred(letter):
    #creating a variable that can be called later that converts the letter to a phred
    phred_score= ord(letter)-33
    #returning the phred_score
    return(phred_score)

#making a function called reverse_complement
```

```
def reverse_complement(bases):
    #a dictionary that has the compliment of each base as values
    complement = {'A': 'T', 'C': 'G', 'G': 'C', 'T': 'A', 'N':'N'}
    #returning the compliment base and making the reverse.
    return ''.join([complement[base] for base in bases[::-1]])

######################################################################################
#Creating Counters

line_ctr=0

cutoff=30

ctr_low_qs=0
ctr_good_qs=0

ctr_yes_N=0
ctr_No_N=0

barcodes_notin_dict=0
barcodes_in_dict=0


reads_No_match=0
reads_Do_match=0

######################################################################################
##Creating Variables for Files and Reading in Files

#setting variables for each file
r1="/Users/mandiedriskill/Bi624_Assignments/demultiplexing/test_r1.txt"
r2="/Users/mandiedriskill/Bi624_Assignments/demultiplexing/test_r2.txt"
i1="/Users/mandiedriskill/Bi624_Assignments/demultiplexing/test_i1.txt"
i2="/Users/mandiedriskill/Bi624_Assignments/demultiplexing/test_i2.txt"
#opening R1,R2,I1,I2 files and setting as variables
#creating R1 and R2 undetermined files and setting as variables
#creating a count data file and setting as variables
with \
open(r1, "r") as r_1, open(r2, "r") as r_2, \
open(i1, "r") as i_1, open(i2, "r") as i_2, \
open('R1_undetermine.fq', 'w') as undetermined_r1, \
open('R2_undetermine.fq', 'w') as undetermined_r2, \
open('count_data', 'w') as count_data:

######################################################################################
##Creating Variables for Individual Lines

    while r_1 and r_2 and i_1 and i_2:
        #setting each line as a variable and stripping the new ling at the end
        line1_i1 = i_1.readline().strip('\n')
        if not line1_i1:
            break
        line2_i1 = i_1.readline().strip('\n')
        line3_i1 = i_1.readline().strip('\n')
        line4_i1 = i_1.readline().strip('\n')

        #setting each line as a variable and stripping the new ling at the end
        line1_i2 = i_2.readline().strip('\n')
        if not line1_i2:
            break
        line2_i2 = i_2.readline().strip('\n')
        line3_i2 = i_2.readline().strip('\n')
        line4_i2 = i_2.readline().strip('\n')

        #setting each line as a variable and stripping the new ling at the end
        line1_r1 = r_1.readline().strip('\n')
        if not line1_r1:
            break
        line2_r1 = r_1.readline().strip('\n')
        line3_r1 = r_1.readline().strip('\n')
        line4_r1 = r_1.readline().strip('\n')
        #creating a variable that prints all 4 lines in fastq format with the index in
        lines_r1 = line1_r1+":"+line2_i1+'\n'+line2_r1+'\n'+line3_r1+'\n'+line4_r1+'\n
        #print(lines_r1)

        #setting each line as a variable and stripping the new ling at the end
        line1_r2 = r_2.readline().strip('\n')
        if not line1_r2:
            break
        line2_r2 = r_2.readline().strip('\n')
        line3_r2 = r_2.readline().strip('\n')
        line4_r2 = r_2.readline().strip('\n')

        rev_i2 = reverse_complement(line2_i2)
        #creating a variable that prints all 4 lines in fastq format with the index in
        lines_r2 = line1_r2+":"+rev_i2+'\n'+line2_r2+'\n'+line3_r2+'\n'+line4_r2+'\n'
        print(line2_i2)
        print(lines_r2)

######################################################################################
##Low Average Quality Score Removal

#removing average read quality scores that are below a cutoff of 30. I choose 30 becau
##score were above 30 in the demultiplex index graphs, which any score above 30 is 99..
##demultiplex index graphs, any cutoff above 30 will remove a good portion of good rea
##to do that. Low read quality scores are most likely due to Ns, which will be removed
```

```
##is reads with average quality scores above 30 (99.9% accurate).


        #creating a list to store characters in for index1
        phred_list_i1=[]
        #looking at each character in the quality score line for index1
        for char in line4_i1:
            #converting each character to a phred score
            phred_score_i1=(convert_phred(char))
            #print(phred_score_i1)#checking right characters are printed

        #creating a list to store characters in for index2
        phred_list_i2=[]
        #looking at each character in the quality score line for index2
        for char in line4_i2:
            #converting each character to a phred score
            phred_score_i2=(convert_phred(char))
            #print(phred_score_i2)#checking right characters are printed

            #appending characters to a list
            phred_list_i1.append(phred_score_i1)
            phred_list_i2.append(phred_score_i2)
            #print(phred_list_i1)#making sure list look correctly
            #print(phred_list_i2)#making sure list look correctly

        #creating variables that has the sum of each line
        phred_sum_i1 = sum(phred_list_i1)
        phred_sum_i2 = sum(phred_list_i2)
        #print("1",phred_sum_i1)#checking to make sure sums are correct
        #print("2",phred_sum_i2)#checking to make sure sums are correct

        #creating variables that are the average of each index line
        phred_ave_i1 = (phred_sum_i1/len(line4_i1))
        phred_ave_i2 = (phred_sum_i2/len(line4_i2))
        #print("1",phred_ave_i1)#checking to make sure average are correct
        #print("2",phred_ave_i2)#checking to make sure average are correct

        #writing reads to undetermined file (R1 and R2) if they are below the set cuto
        if phred_ave_i1 < cutoff or phred_ave_i2 < cutoff:
            #counting number of reads that are below the cutoff level
            ctr_low_qs +=1
            undetermined_r1.write(lines_r1)
            undetermined_r2.write(lines_r2)
        #if reads are above the cutoff level
        else:
            #counting the number of reads that are above the cutoff level
            ctr_good_qs +=1
#######################################################################################
```

```
##Removing Reads with Ns

            #if index1 and index2 sequence lines have Ns. Reads are witten to undeterm
            if "N" in line2_i1 or "N" in line2_i2:
                #counting number of reads that have Ns
                ctr_yes_N +=1
                undetermined_r1.write(lines_r1)
                undetermined_r2.write(lines_r2)
            else:
                #counting number of reads that don't have Ns
                ctr_No_N +=1
                #making the reverse compliment of index2 sequence
                reverse_i2 = reverse_complement(line2_i2)
                #print("original", line2_i2)#viewing origional index2
                #print("reverse", reverse_i2)#making sure origional was reversed compl
                #finding indexes that are not in the dictionary and writing them to un
                if line2_i1 not in barcode_dict.keys() and reverse_i2 not in barcode_d
                    #print("1",line2_i1)#viewing index1
                    #print("2",reverse_i2)#making sure indexes are reversed compliment
                    #counting the indexes that are not in the dictionary
                    barcodes_notin_dict+=1
                    undetermined_r1.write(lines_r1)
                    undetermined_r2.write(lines_r2)

                else:
                    #finding the indexes that are in the dictionary
                    if line2_i1 in barcode_dict.keys() and reverse_i2 in barcode_dict.
                        #counting the indexes that are in the dicitonary
                        barcodes_in_dict += 1
                        #print("R1",line2_i1)#checking indexes that are found in the d
                        #print("R2",reverse_i2)#checking indexes that are found in the

                    #finind index reads that don't match and writing them to underterm
                    if reverse_i2 != line2_i1:
                        #counting index reads that don't match
                        reads_No_match+=1
                        #print("R1",line2_i1)#checking indexes that don't match
                        #print("R2",reverse_i2)#checking indexes that don't match
                        undetermined_r1.write(lines_r1)
                        undetermined_r2.write(lines_r2)

                    #finding indexes that match and if they do writing them to files t
                    if reverse_i2==line2_i1:
                        #counting reads that do match
                        reads_Do_match+=1
                        #print("R1",line2_i1)#checking index reads that match
                        #print("R2",reverse_i2)#checking index reads that match
                        #print(lines_r1)#checking associated reads that match
```

```
                                #print(lines_r2)#checking associated reads that match
                                file_dict_r1[line2_i1].write(lines_r1)
                                file_dict_r2[reverse_i2].write(lines_r2)

            #writing count data to a file called count_data
            count_data.write("{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n
                                                                       "# of Read Pairs with Go
                                                                       "# of Read Pairs with Ns
                                                                       "# of Read Pairs withou
                                                                       "# of Read Pairs Not in
                                                                       "# of Read Pairs in Dic
                                                                       "# of Read Pairs that Do
                                                                       "# of Read Pairs that Do

        #closing files that were created from the dictionary barcodes
        for barcode in file_dict_r1:
            file_dict_r1[barcode].close()
            file_dict_r2[barcode].close()
```

```
CCTTCGAC
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GTCGAAGG
1)Bad quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<

CCTTCGAC
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GTCGAAGG
2)Good quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<

NTTTTTTT
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AAAAAAAN
3)Has Ns
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<

TTTTTTTT
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AAAAAAAA
4)No Ns
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<

GGGGGGGG
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:CCCCCCCC
5)Key not in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<

TTTTTTTT
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AAAAAAAA
6)Key not in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<

TACGCTAC
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GTAGCGTA
7)Key in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<

TACGCTAC
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GTAGCGTA
8)Key in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<

GCTATCCT
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AGGATAGC
9)Key in dictionary and matches
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F

GCTATCCT
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AGGATAGC
10)Key in dictionary and matches
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F

GCTATCCT
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AGGATAGC
11)Key in dictionary and matches, bad quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F

ATCCGGTA
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:TACCGGAT
12)Key in dictionary and matches, Good quality score, and match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F

CGCATGAT
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:ATCATGCG
13)Key in dictionary, Good quality score, and don't match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

```
In [ ]: #test output R1_undetermine.fq output
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:CCTTCGAC
        1)Good quality score
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:CCTTCGAC
        2)Bad quality score
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:TTTTTTTT
        3)No Ns
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:NTTTTTTT
        4)Has Ns
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AGAGTCCA
        5)Key in dictionary
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AGAGTCCA
        6)Key in dictionary
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GGGGGGGG
        7)Key not in dictionary
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:CCCCCCCC
        8)Key not in dictionary
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AGGATAGC
        11)Key in dictionary and matches, bad quality score
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:TACCGGAT
```

```
13)Key in dictionary, Good quality score, and don't match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

In [ ]:
```
#test output R2_undetermine.fq output
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:CCTTCGAC
1)Bad quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:CCTTCGAC
2)Good quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:NTTTTTTT
3)Has Ns
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:TTTTTTTT
4)No Ns
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GGGGGGGG
5)Key not in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:TTTTTTTT
6)Key not in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:TACGCTAC
7)Key in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:TACGCTAC
8)Key in Dictionary
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GCTATCCT
11)Key in dictionary and matches, bad quality score
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
@K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:CGCATGAT
13)Key in dictionary, Good quality score, and don't match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

In [ ]:
```
#output files, code makes them all, but they are filled only if those reads are presen
AACAGCGA_R1.fq
```

```
AACAGCGA_R2.fq
ACGATCAG_R1.fq
ACGATCAG_R2.fq
AGAGTCCA_R1.fq
AGAGTCCA_R2.fq
AGGATAGC_R1.fq
AGGATAGC_R2.fq
ATCATGCG_R1.fq
ATCATGCG_R2.fq
ATCGTGGT_R1.fq
ATCGTGGT_R2.fq
GTCCTAAG_R1.fq
GTCCTAAG_R2.fq
TACCGGAT_R1.fq
TACCGGAT_R2.fq
TAGCCATG_R1.fq
TAGCCATG_R2.fq
TATGGCAC_R1.fq
TATGGCAC_R2.fq
TCGACAAG_R1.fq
TCGACAAG_R2.fq
ATCATGCG_R1.fq
ATCATGCG_R2.fq
ATCGTGGT_R1.fq
ATCGTGGT_R2.fq
GATCTTGC_R1.fq
GATCTTGC_R2.fq
CACTTCAC_R1.fq
CACTTCAC_R2.fq
CGATCGAT_R1.fq
CGATCGAT_R2.fq
TGTTCCGT_R1.fq
TGTTCCGT_R2.fq
TCTTCGAC_R1.fq
TCTTCGAC_R2.fq
TCGGATTC_R1.fq
TCGGATTC_R2.fq
GTCCTAAG_R1.fq
GTCCTAAG_R2.fq
GTAGCGTA_R1.fq
GTAGCGTA_R2.fq
GCTACTCT_R1.fq
GCTACTCT_R2.fq
TCGAGAGT_R1.fq
TCGAGAGT_R2.fq
```

In [ ]:
```
#test output
head TACCGGAT_R1.fq
```

```
12)Key in dictionary and matches, Good quality score, and match
+
A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

```
In [ ]: #test output
        head TACCGGAT_R2.fq
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:ATCCGGTA
        12)Key in dictionary and matches, Good quality score, and match
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

```
In [ ]: #test output
        head AGGATAGC_R1.fq
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AGGATAGC
        9)Key in dictionary and matches
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:AGGATAGC
        10)Key in dictionary and matches
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

```
In [ ]: #test output
        head AGGATAGC_R2.fq
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GCTATCCT
        9)Key in dictionary and matches
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:GCTATCCT
        10)Key in dictionary and matches
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F
```

```
In [ ]: #test output
        cat count_data
        # of Read Pairs with Low SQ R1
        3
        # of Read Pairs with Good SQ R1
        10
        # of Read Pairs with Ns
        2
        # of Read Pairs without Ns
        8
        # of Read Pairs Not in Dictionary
        4
        # of Read Pairs in Dictionary
        4
        # of Read Pairs that Don't Match
        1
```

```
# of Read Pairs that Do Match
3
```

```
In [ ]: #!/usr/bin/env python3
        import argparse

        def get_arguments():
            parser = argparse.ArgumentParser(description='k-mer size program')
            parser.add_argument("-r1", "--read1", help="input string", required=True, type=str
            parser.add_argument("-r2", "--read2", help="input string", required=True, type=str
            parser.add_argument("-i1", "--index1", help="input string", required=True, type=st
            parser.add_argument("-i2", "--index2", help="input string", required=True, type=st
            parser.add_argument("-b", "--barcode_file", help="input string", required=True, ty
            parser.add_argument("-c", "--cutoff", help="input string", required=True, type=str
            return parser.parse_args()

        args = get_arguments()
        read1 = args.read1
        read2 = args.read2
        index1 = args.index1
        index2 = args.index2
        barcode_file = args.barcode_file
        cutoff = args.cutoff

        #actual code used

        #opening nano to make script
        #nano demultiplexing.py

        ######################################################################################
        #!/usr/bin/env python

        #Demultiplexing Python Assignment
        ######################################################################################
        ##Making Dictionary for Barcodes and output files

        #dictionary for R1 barcodes for making output file
        file_dict_r1={}
        #dictionary for R2 barcodes for making output files
        file_dict_r2={}
        #dictionary for checking if index matches
        barcode_dict = {}
        #Reading each line in the barcode file
        for line in open(barcode_file).readlines():
            #striping the lines and spliting in the barcode file
            sample, barcode = line.strip().split()
            #setting the barcodes keys and the samples as values
            barcode_dict[barcode] = sample
```

```
    #creating R1 files that are named with all the barcodes in the dictionary
    file_dict_r1[barcode]=open(barcode + "_R1.fq", "w")
    #creating R2 files that are named with all the barcodes in the dictionary
    file_dict_r2[barcode]=open(barcode + "_R2.fq", "w")


######################################################################################
##Creating Functions

#making a function called convert_phred
def convert_phred(letter):
    #creating a variable that can be called later that converts the letter to a phred
    phred_score= ord(letter)-33
    #returning the phred_score
    return(phred_score)

#making a function called reverse_complement
def reverse_complement(bases):
    #a dictionary that has the compliment of each base as values
    complement = {'A': 'T', 'C': 'G', 'G': 'C', 'T': 'A', 'N':'N'}
    #returning the compliment base and making the reverse.
    return ''.join([complement[base] for base in bases[::-1]])

######################################################################################
#Creating Counters
line_ctr=0



ctr_low_qs=0
ctr_good_qs=0

ctr_yes_N=0
ctr_No_N=0

barcodes_notin_dict=0
barcodes_in_dict=0


reads_No_match=0
reads_Do_match=0

######################################################################################
##Creating Variables for Files and Reading in Files
#setting variables for each file

#opening R1,R2,I1,I2 files and setting as variables
#creating R1 and R2 undetermined files and setting as variables
```

```
#creating a count data file and setting as variables
with \
open(read1, "r") as r_1, open(read2, "r") as r_2, \
open(index1, "r") as i_1, open(index2, "r") as i_2, \
open('R1_undetermine.fq', 'w') as undetermined_r1, \
open('R2_undetermine.fq', 'w') as undetermined_r2, \
open('count_data', 'w') as count_data:

######################################################################################
##Creating Variables for Individual Lines
    while r_1 and r_2 and i_1 and i_2:
        #setting each line as a variable and stripping the new ling at the end
        line1_i1 = i_1.readline().strip('\n')
        if not line1_i1:
            break
        line2_i1 = i_1.readline().strip('\n')
        line3_i1 = i_1.readline().strip('\n')
        line4_i1 = i_1.readline().strip('\n')

        #setting each line as a variable and stripping the new ling at the end
        line1_i2 = i_2.readline().strip('\n')
        if not line1_i2:
            break
        line2_i2 = i_2.readline().strip('\n')
        line3_i2 = i_2.readline().strip('\n')
        line4_i2 = i_2.readline().strip('\n')

        #setting each line as a variable and stripping the new ling at the end
        line1_r1 = r_1.readline().strip('\n')
        if not line1_r1:
            break
        line2_r1 = r_1.readline().strip('\n')
        line3_r1 = r_1.readline().strip('\n')
        line4_r1 = r_1.readline().strip('\n')
        #creating a variable that prints all 4 lines in fastq format with the index in
        lines_r1 = line1_r1+":"+line2_i1+'\n'+line2_r1+'\n'+line3_r1+'\n'+line4_r1+'\n
        #print(lines_r1)

        #setting each line as a variable and stripping the new ling at the end
        line1_r2 = r_2.readline().strip('\n')
        if not line1_r2:
            break
        line2_r2 = r_2.readline().strip('\n')
        line3_r2 = r_2.readline().strip('\n')
        line4_r2 = r_2.readline().strip('\n')

        rev_i2 = reverse_complement(line2_i2)
        #creating a variable that prints all 4 lines in fastq format with the index in
        lines_r2 = line1_r2+":"+rev_i2+'\n'+line2_r2+'\n'+line3_r2+'\n'+line4_r2+'\n'
        #print(lines_r2)
#######################################################################################
##Low Average Quality Score Removal
#removing average read quality scores that are below a cutoff of 30. I choose 30 becau
##score were above 30 in the demultiplex index graphs, which any score above 30 is 99..
##demultiplex index graphs, any cutoff above 30 will remove a good portion of good rea
##to do that. Low read quality scores are most likely due to Ns, which will be removed
##is reads with average quality scores above 30 (99.9% accurate).


        #creating a list to store characters in for index1
        phred_list_i1=[]
        #looking at each character in the quality score line for index1
        for char in line4_i1:
            #converting each character to a phred score
            phred_score_i1=(convert_phred(char))
            #print(phred_score_i1)#checking right characters are printed

        #creating a list to store characters in for index2
        phred_list_i2=[]
        #looking at each character in the quality score line for index2
        for char in line4_i2:
            #converting each character to a phred score
            phred_score_i2=(convert_phred(char))
            #print(phred_score_i2)#checking right characters are printed

            #appending characters to a list
            phred_list_i1.append(phred_score_i1)
            phred_list_i2.append(phred_score_i2)
            #print(phred_list_i1)#making sure list look correctly
            #print(phred_list_i2)#making sure list look correctly

        #creating variables that has the sum of each line
        phred_sum_i1 = sum(phred_list_i1)
        phred_sum_i2 = sum(phred_list_i2)
        #print("1",phred_sum_i1)#checking to make sure sums are correct
        #print("2",phred_sum_i2)#checking to make sure sums are correct

        #creating variables that are the average of each index line
        phred_ave_i1 = (phred_sum_i1/len(line4_i1))
        phred_ave_i2 = (phred_sum_i2/len(line4_i2))
        #print("1",phred_ave_i1)#checking to make sure average are correct
        #print("2",phred_ave_i2)#checking to make sure average are correct
        #writing reads to undetermined file (R1 and R2) if they are below the set cuto
        if phred_ave_i1 < cutoff or phred_ave_i2 < cutoff:
            #counting number of reads that are below the cutoff level
            ctr_low_qs +=1
            undetermined_r1.write(lines_r1)
            undetermined_r2.write(lines_r2)
        #if reads are above the cutoff level
        else:
            #counting the number of reads that are above the cutoff level
            ctr_good_qs +=1
######################################################################################
##Removing Reads with Ns

            #if index1 and index2 sequence lines have Ns. Reads are witten to undeterm
            if "N" in line2_i1 or "N" in line2_i2:
                #counting number of reads that have Ns
                ctr_yes_N +=1
                undetermined_r1.write(lines_r1)
                undetermined_r2.write(lines_r2)
            else:
#counting number of reads that don't have Ns
                ctr_No_N +=1
                #making the reverse compliment of index2 sequence
                reverse_i2 = reverse_complement(line2_i2)
                #print("original", line2_i2)#viewing origional index2
                #print("reverse", reverse_i2)#making sure origional was reversed compl
                #finding indexes that are not in the dictionary and writing them to un
                if line2_i1 not in barcode_dict.keys() or reverse_i2 not in barcode_di
                    #print("1",line2_i1)#viewing index1
                    #print("2",reverse_i2)#making sure indexes are reversed compliment
                    #counting the indexes that are not in the dictionary
                    barcodes_notin_dict+=1
                    undetermined_r1.write(lines_r1)
                    undetermined_r2.write(lines_r2)

                else:
                    #finding the indexes that are in the dictionary
                    if line2_i1 in barcode_dict.keys() and reverse_i2 in barcode_dict.
                        #counting the indexes that are in the dicitonary
                        barcodes_in_dict += 1
                        #print("R1",line2_i1)#checking indexes that are found in the d
                        #print("R2",reverse_i2)#checking indexes that are found in the

                    #finind index reads that don't match and writing them to underterm
                    if reverse_i2 != line2_i1:
                        #counting index reads that don't match
                        reads_No_match+=1
                        #print("R1",line2_i1)#checking indexes that don't match
                        #print("R2",reverse_i2)#checking indexes that don't match
                        undetermined_r1.write(lines_r1)
                        undetermined_r2.write(lines_r2)
                        #finding indexes that match and if they do writing them to fil
                    if reverse_i2==line2_i1:
                        #counting reads that do match
                        reads_Do_match+=1
                        #print("R1",line2_i1)#checking index reads that match
                        #print("R2",reverse_i2)#checking index reads that match
                        #print(lines_r1)#checking associated reads that match
                        #print(lines_r2)#checking associated reads that match
                        file_dict_r1[line2_i1].write(lines_r1)
                        file_dict_r2[reverse_i2].write(lines_r2)

    #writing count data to a file called count_data
    count_data.write("{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n{}\n
                                                                "# of Read Pairs with G
                                                                "# of Read Pairs with N
                                                                "# of Read Pairs withou
                                                                "# of Read Pairs Not in
                                                                "# of Read Pairs in Dic
                                                                "# of Read Pairs that D
                                                                "# of Read Pairs that D

#closing files that were created from the dictionary barcodes
for barcode in file_dict_r1:
    file_dict_r1[barcode].close()
    file_dict_r2[barcode].close()

######################################################################################
#chaning file permissions to run the script
#chmod 755 demultiplexing.py
```

In [ ]:

```
#creating a batch script to run code on talapas
nano demultiplexing.slurm

######################################################################################
#!/usr/bin/env bash
#SBATCH --partition=long    ### Partition
#SBATCH --job-name=job  ### Job Name
#SBATCH --output=job.out    ### File in which to store job output
#SBATCH --time=2-00:00:00   ### Wall clock time limit in Days-HH:MM:SS
#SBATCH --nodes=1           ### Number of nodes needed for the job
#SBATCH --ntasks-per-node=20 ### Number of tasks to be launcged per Node

cd /projects/bgmp/mdriskil/Bi624_Assignments/demultiplex

./demultiplexing.py

######################################################################################
#chaning file permission
chmod 755 demultiplexing.slurm
```

```
#submitting script to talapas
sbatch demultiplexing.slurm
```

```
In [ ]: #actual output
        head R1_undetermine.fq
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 1:N:0:1:NCTTCGAC
        GNCTGGCATTCCCAGAGACATCAGTACCCAGTTGGTTCAGACAGTTCCTCTATTGGTTGACAAGGTCTTCATTTCTAGTGATATCAA
        +
        A#A-<FJJJ<JJJJJJJJJJJJJJJFJJJJFFJJFJJJAJJJJ-AJJJJJJJFFJJJJJJFFA-7<AJJJFFAJJJJJF<F--J
        @K00337:83:HJKJNBBXX:8:1101:1286:1191 1:N:0:1:NACAGCGA
        CNACCTGTCCCCAGCTCACAGGACAGCACACCAAAGGCGGCAACCCACACCCAGTTTTACAGCCACACAGTGCCTTGTTTTACTTGA
        +
        A#AAFJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJAJJJJJJJJJJJJJJFJJJJJFFFFJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 1:N:0:1:NTCCTAAG
        GNGGTCTTCTACCTTTCTCTTCTTTTTTGGAGGAGTAGAATGTTGAGAGTCAGCAGTAGCCTCATCATCACTAGATGGCATTTCTTC
```

```
In [ ]: #actual output
        head R2_undetermine.fq
        @K00337:83:HJKJNBBXX:8:1101:1265:1191 4:N:0:1:TCTTCGAN
        NTTTTGATTTACCTTTCAGCCAATGAGAAGGCCGTTCATGCAGACTTTTTTAATGATTTTGAAGACCTTTTTGATGATGATGATGTC
        +
        #AAFAFJJ-----F---7-<FA-F<AFFA-JJJ77<FJFJFJJJJJJJJJAFJFFAJJJJJJJFJF7-AFFJJ7F7JFJJFJ7F
        @K00337:83:HJKJNBBXX:8:1101:1286:1191 4:N:0:1:AACAGCGN
        NTGTGTAGACAAAAGTTTTCATGAGTCTGTAAGCTGTCTATTGTCTCCTGAAAAGAAACCAGAAGTTTTCCCCTAAATGTGTTTAGA
        +
        #A-AFFJJFJJJJJJJJJJJJJJJJ<JAJFJJJJF<JFJJJAJJJJJJJJJJJJJJJJJJFJJJAJJFJJJFJJJF<JJA-JJJ-
        @K00337:83:HJKJNBBXX:8:1101:1347:1191 4:N:0:1:GTCCTAAN
        NAAATGCCATCTAGTGATGATGAGGCTACTGCTGACTCTCAACATTCTACTCCTCCAAAAAAGAAGAGAAAGATTCCAACCCCCAGA
```

```
In [ ]: #actual output
        head AACAGCGA_R1.fq
        @K00337:83:HJKJNBBXX:8:1101:1479:1701 1:N:0:1:AACAGCGA
        CCTTTGCTTAATCTGCTCTTGTCTTTCAGCACTAAGCTGTTCTTTTTCTTCTTTCATGGCTGAAATTTTTGTTTTCAGTTCTCTAAC
        +
        AAAAAJJFJJJJJJJJJJJJFJJJJFJJJJJJFJJJJJFFJJJJJJJJJJJJJJJFJJJJJJJAFJJJJJJJJJJJJJFF<JJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1966:1701 1:N:0:1:AACAGCGA
        ATGTTCAGTTGTCTCTGGACTTACAAGTTCCTCAGTACACTGCTCCAGAGCCTCCCTCTCTTCATCCATGTCCTGACTGGTCTCCTT
        +
        AAFAFJJJJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:4076:1701 1:N:0:1:AACAGCGA
        TGCCCCCTTCCTGGGCATCCGTTCGGGTTTTCTAACCGTTGTCTTGTTCTGGGGTGAGCGGGGAGCGCATGCAGAGATCCCAAGGCC
```

```
In [ ]: #actual output
        head AACAGCGA_R2.fq
        @K00337:83:HJKJNBBXX:8:1101:1479:1701 4:N:0:1:AACAGCGA
        CAACGAGACGCGCGCTAAGCTCGACGAGCTTTCTGCTAAGCGAGAAACGAGTGGAGAGAAATCCAGACAACTAAGGGATGCCCAGCA
        +
        A<AFFJJJJJJJJJJJJJJAJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJ7JJJJJJJJJJJJJJ
        @K00337:83:HJKJNBBXX:8:1101:1966:1701 4:N:0:1:AACAGCGA
```

```
CTTGTGAGAACTGAACCACGAGAGCAGCTCCTTGATCAGCTGCAAAAGAAACAACCACCAATGATGCTAAACAGCTCAGAAGCCAG
+
AAFFFJJJAJJJJJJJJJJJJJJJJJJJJJJJJFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
@K00337:83:HJKJNBBXX:8:1101:4076:1701 4:N:0:1:AACAGCGA
CCGAGCGGCAGAGAGTGGGGTAGGCAGGAGGCTCCCTGTAAACATTTGGACTTGGGTCAGGGCTGGTGTTGGACAAAGCCAGGGGT
```

```
In [ ]: #number of reads in original files
        1294_S1_L008_R1_001.fastq | wc -l
        1452986940/4=363246735
        cat 1294_S1_L008_R2_001.fastq | wc -l
        1452986940/4=363246735
        cat 1294_S1_L008_R3_001.fastq | wc -l
        1452986940/4=363246735
        cat 1294_S1_L008_R4_001.fastq | wc -l
        1452986940/4=363246735

        #number of reads in undetermined files
        cat R1_undetermine.fq | wc -l
        239463044/4=59865761
        cat R2_undetermine.fq | wc -l
        239463044/4=59865761
        59865761/363246735=0.1648074 proportion

        #number of reads in 48 fastq files
        cat AACAGCGA_R1.fq | wc -l
        32356964/4=8089241
         cat AACAGCGA_R2.fq | wc -l
        32356964/4=8089241
        8089241/363246735=0.02226927 proportion


        cat ACGATCAG_R1.fq | wc -l
        30192604/4=7548151
        cat ACGATCAG_R2.fq | wc -l
        30192604/4=7548151
        7548151/363246735=0.02077968 proportion

        cat AGAGTCCA_R1.fq | wc -l
        42039364/4=10509841
        cat AGAGTCCA_R2.fq | wc -l
        42039364/4=10509841
        10509841/363246735=0.02893306 proportion

        cat AGGATAGC_R1.fq | wc -l
        32629256/4=8157314
        cat AGGATAGC_R2.fq | wc -l
        32629256/4=8157314
        8157314/363246735=0.02245668 proportion
```

```
cat ATCATGCG_R1.fq | wc -l
37543456/4=9385864
cat ATCATGCG_R2.fq | wc -l
37543456/4=9385864
9385864/363246735=0.02583881 proportion

cat ATCGTGGT_R1.fq | wc -l
24685116/4=6171279
cat ATCGTGGT_R2.fq | wc -l
24685116/4=6171279
6171279/363246735=0.01698922 proportion

cat CACTTCAC_R1.fq | wc -l
15547792/4=3886948
cat CACTTCAC_R2.fq | wc -l
15547792/4=3886948
3886948/363246735=0.01070057 proportion

cat CGATCGAT_R1.fq | wc -l
20812540/4=5203135
cat CGATCGAT_R2.fq | wc -l
20812540/4=5203135
5203135/363246735=0.01432397 proportion

cat CGGTAATC_R1.fq | wc -l
18208596/4=4552149
cat CGGTAATC_R2.fq | wc -l
18208596/4=4552149
4552149/363246735=0.01253184 proportion

cat CTAGCTCA_R1.fq | wc -l
65049000/4=16262250
cat CTAGCTCA_R2.fq | wc -l
650490004=16262250
16262250/363246735=0.04476916 proportion

cat CTCTGGAT_R1.fq | wc -l
127253488/4=31813372
cat CTCTGGAT_R2.fq | wc -l
127253488/4=31813372
31813372/363246735=0.08758061 proportion

cat GATCAAGG_R1.fq | wc -l
24661800/4=6165450
cat GATCAAGG_R2.fq | wc -l
24661800/4=6165450
6165450/363246735=0.01697317 proportion
```

```
cat GATCTTGC_R1.fq | wc -l
13813748/4=3453437
cat GATCTTGC_R2.fq | wc -l
13813748/4=3453437
3453437/363246735=0.009507138 proportion

GCTACTCT_R1.fq | wc -l
26461564/4=6615391
cat GCTACTCT_R2.fq | wc -l
26461564/4=6615391
6615391/363246735=0.01821184 proportion

cat GTAGCGTA_R1.fq | wc -l
29790372/4=7447593
cat GTAGCGTA_R2.fq | wc -l
29790372/4=7447593
7447593/363246735=0.02050285 proportion

cat GTCCTAAG_R1.fq | wc -l
32994664/4=8248666
cat GTCCTAAG_R2.fq | wc -l
32994664/4=8248666
8248666/363246735=0.02270816 proportion

cat TACCGGAT_R1.fq | wc -l
272873952/4=68218488
cat TACCGGAT_R2.fq | wc -l
272873952/4=68218488
68218488/363246735=0.1878021 proportion

cat TAGCCATG_R1.fq | wc -l
39729840/4=9932460
cat TAGCCATG_R2.fq | wc -l
39729840/4=9932460
9932460/363246735=0.02734356 proportion

cat TATGGCAC_R1.fq | wc -l
41122396/4=10280599
cat TATGGCAC_R2.fq | wc -l
41122396/4=10280599
10280599/363246735=0.02830197 proportion

cat TCGACAAG_R1.fq | wc -l
14295448/4=3573862
cat TCGACAAG_R2.fq | wc -l
14295448/4=3573862
3573862/363246735=0.009838662 proportion
```

```
        cat TCGAGAGT_R1.fq | wc -l
        38668032/4=9667008
        cat TCGAGAGT_R2.fq | wc -l
        38668032/4=9667008
        9667008/363246735=0.02661279 proportion

        cat TCGGATTC_R1.fq | wc -l
        17165336/4=4291334
        cat TCGGATTC_R2.fq | wc -l
        17165336/4=4291334
        4291334/363246735=0.01181383 proportion

        cat TCTTCGAC_R1.fq | wc -l
        157519852/4=39379963
        cat TCTTCGAC_R2.fq | wc -l
        157519852/4=39379963
        39379963/363246735=0.1084111 proportion

        cat TGTTCCGT_R1.fq | wc -l
        58108716/4=14527179
        cat TGTTCCGT_R2.fq | wc -l
        58108716/4=14527179
        14527179/363246735=0.03999259 proportion
```

```
In [ ]: cat count_data
        # of Read Pairs with Low SQ R1
        46324005
        46324005/363246735=0.1275277 proportion

        # of Read Pairs with Good SQ R1
        316922730
        316922730/363246735=0.8724723 proportion

        # of Read Pairs with Ns
        3139834
        3139834/363246735=0.008643805 proportion

        # of Read Pairs without Ns
        313782896
        313782896/363246735=0.8638285 proportion

        # of Read Pairs Not in Dictionary
        9866439
        9866439/363246735=0.02716181 proportion

        # of Read Pairs in Dictionary
        303916457
```

```
303916457/363246735=0.8366667 proportion

# of Read Pairs that Don't Match-index hopping
535483
535483/363246735=0.001474158 proportion

# of Read Pairs that Do Match
303380974
303380974/363246735=0.8351926 proportion
```