ace04_process_reads_full

August 21, 2018

In [5]:
```
# import packages
from Bio.SeqIO.QualityIO import FastqGeneralIterator
```

In [6]:
```
read_pair1 = open("../data/c_burnetii_1k_R1.fastq", "r")
read_pair2 = open("../data/c_burnetii_1k_R2.fastq", "r")

filtered_read_pair1_out = open("../data/c_burnetii_1k_trim_R1.fastq", "w")
filtered_read_pair2_out = open("../data/c_burnetii_1k_trim_R2.fastq", "w")
```

In [7]:
```
# instantiate a fastq iterator using Biopython package
f_iter1 = FastqGeneralIterator(read_pair1)
f_iter2 = FastqGeneralIterator(read_pair2)
```

In [12]:
```
# settings, constants
LENGTH = 60
QUAL = 35
BARCODE_LENGTH = 0
```

In [13]:
```
# iterate through reads, in lock step, quality trim
for (title1, seq1, qual1), (title2, seq2, qual2) in zip(f_iter1, f_iter2): #zip does

    # do not use reads with "Ns"
    if "N" not in seq1 and "N" not in seq2:

        # set flag to zero
        seq1_flag = 0
        seq2_flag = 0
        qual1_len = qual1[BARCODE_LENGTH:(LENGTH + BARCODE_LENGTH - 1)]
        qual2_len = qual2[BARCODE_LENGTH:(LENGTH + BARCODE_LENGTH - 1)]

        # check if quality score meets cutoff for both reads
        for i in qual1_len:
            if ord(i)-33 < QUAL:
                seq1_flag += 1

        for i in qual2_len:
            if ord(i)-33 < QUAL:
                seq2_flag += 1
```

```
        # only keep those reads that pass quality
        if seq1_flag == 0 and seq2_flag == 0:

            filtered_read_pair1_out.write("@%s\n%s\n+\n%s\n" % (title1, seq1[BARCODE_
            filtered_read_pair2_out.write("@%s\n%s\n+\n%s\n" % (title2, seq2[BARCODE_

# clean up
filtered_read_pair1_out.close()
filtered_read_pair2_out.close()
read_pair1.close()
read_pair2.close()
```