In [1]:

```
import pysam
```

In [2]:

```
# read our file in BAM format, and create a AlignmentFile object
data = pysam.AlignmentFile("merged-tumor.bam", "rb")
```

In [3]:

```
# take the first read out of the bam file
read_first = data.head(1)
for iter in read_first:
    print(str(iter))
```

```
C0HVYACXX120402:7:1207:5722:57044       1187    20      9483248 27      76M
        20      9483381 76      TTTTCAAACAGTATCTATGCCTGCCAAATGTGAACATATAAAAAA
AAACCAGAATGTGCCATTCTGATTTAAACTG array('B', [28, 28, 27, 29, 31, 30, 31, 31, 2
9, 31, 35, 30, 29, 31, 34, 30, 29, 23, 41, 32, 20, 30, 29, 34, 34, 29, 30, 3
1, 30, 30, 30, 33, 33, 26, 39, 12, 25, 19, 32, 30, 35, 28, 35, 33, 23, 33, 3
5, 36, 30, 38, 33, 41, 34, 35, 31, 33, 23, 30, 30, 36, 27, 32, 29, 34, 35, 4
1, 33, 31, 33, 29, 32, 32, 31, 31, 31, 34])     [('XA', 'GL000217.1,-110754,7
6M,1;'), ('BD', 'NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNN'), ('MD', '76'), ('RG', '1'), ('BI', 'NNNNNNNNNNNNNNNNNNNNNN
NNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNN'), ('NM', 0), ('MQ', 2
7), ('AS', 76), ('XS', 71)]
```

In [4]:

```
# make a dict out of AligmentSegment type
read_dict = iter.to_dict()
print('Fields in sam file types:\n')
for keys in read_dict:
    print(keys)
```

```
Fields in sam file types:

name
flag
ref_name
ref_pos
map_quality
cigar
next_ref_name
next_ref_pos
length
seq
qual
tags
```

In [5]:

```
print('Flag in the first read:\n')
print(read_dict['flag'])
```

```
Flag in the first read:

1187
```

By looking at the flag we can see that the first read is:

1. paired (0x1)

2. mapped in proper pair (0x2)

3. mate reverse strand (0x20)

4. second in pair (0x80)

5. PCR or optical duplicate (0x400)

In [6]:

```python
total_num_of_reads = data.count(until_eof = True)
print('The total number of reads is: \n')
print(total_num_of_reads)
```

```
The total number of reads is: 

2921629
```

In [7]:

```python
# read our file again as we used until_eof in the previous section
data = pysam.AlignmentFile("merged-tumor.bam", "rb")

mapping_q_with_zeros = []
mapping_q_non_zeros = []
unmapped = 0

for read in data:

    # check for unmapped reads
    if read.is_unmapped:
        unmapped = unmapped + 1

    # inspect the mapping quality
    temp = read.mapping_quality
    if not temp == 0:
        mapping_q_non_zeros.append(temp)

    mapping_q_with_zeros.append(temp)

print('Number of unmapped reads in the file: \n')
print(unmapped)
print('\n')
print('Number of reads with mapping quality 0: \n')
print(len(mapping_q_with_zeros)-len(mapping_q_non_zeros))
```

```
Number of unmapped reads in the file: 

17765


Number of reads with mapping quality 0: 

126628
```

In [8]:

```
import numpy as np

print('Average mapping quality for all the reads: \n')
print(np.mean(mapping_q_with_zeros))
print('\n')
print('Average mapping quality if reads with 0 map quality are filtered out:
print(np.mean(mapping_q_non_zeros))
```

```
Average mapping quality for all the reads:

55.91379158681681


Average mapping quality if reads with 0 map quality are filtered out:

58.446975510921106
```

In [9]:

```
# close the BAM file
data.close()
```