Deduper_psuedo

Present Problem

PCR Dupliactes are a present within amplification based sequencing kits. A PCR duplicate is defined as a duplicate amplicon that has come from the same parent. You can tell if a PCR is a duplicate based on a descending order of importance with the UMI (*Unique Molecular Indentifier*), then the chromosome, then position, and then strand. If all of these paramters have been viewed before then you know that this particular sequence is a pcr duplicate.

Another thing to take into account is soft clipping. Soft clipping is when the first couple of nucleotides do not match to the sequence the majority of the sequence has bound to. These are clipped off and the starting position is shifted over to match with the new starting position. We need to now shiftover the position by that many in the reported position to correct for it.

Samfile view

Here is the bad output first test.sam file output:

```
samtools view test.sam | head -n 10
NS500451:154:HWKTMBGXX:1:11101:24260:1121:CTGTTCAC  0   2   76814284    36  71M *   0   0   TCCACCACAATC
NS500451:154:HWKTMBGXX:1:11101:24260:1121:CTGTNNNNN 0   2   76814284    36  71M *   0   0   TCCACCACAATC
NS500451:154:HWKTMBGXX:1:11101:24260:1121:CTGTTCAC  0   2   76814284    36  71M *   0   0   TCCACCACAATC
NS500451:154:HWKTMBGXX:1:11101:18996:1145:TTCGCCTA  0   2   130171653   36  40M1I30M    *   0   0   GTCT
```

Here is a corrected sam file output:

```
NS500451:154:HWKTMBGXX:1:11101:24260:1121:CTGTTCAC  0   2   76814284    36  71M *   0   0   TCCACCACAATC
NS500451:154:HWKTMBGXX:1:11101:18996:1145:TTCGCCTA  0   2   130171653   36  40M1I30M    *   0   0   GTCT
```

Pseudo Code

1. Open the sam file (TextIOWrapper)
2. Load in the UMI file as the keys of a dictionary (UMI Dict)
3. Read the file one line at a time until it reaches a line that starts with "^NS500"

a. Then take the line and break it up into its SAM parts. Which will be stored in a list, by using split()
b. After breaking up the sam file, we pull out the important parts. QNAME, UMI, BitFLAG, CIGAR string, & Chroms, begin_pos.

c. Once those are saved into the separate variables. I will first run the UMI to check that it is in the keys of UMI Dict a. If it isn't then throw the read away. If it is then we go into that value. Which will be another dict. The keys of that dictionary will be (pos, chrom, and strand). b. First the cigar string needs to be consulted to see if there is soft clipping. If it is then the amount that was soft clipped off needs to be added back on to the beginning position. If not then do nothing to the begin_pos c. Then we will check to see if the pos is in the values for the "pos" key. We will do this by running a bool check, this will return the positions of the matching pos values in the list. This will be put into a small sub set list. Which will be used for the checking of everything going forward. If it isn't then everything will be added to the dict and the read will be kept, and added onto the the end of the list values of the inner dict. d. Once you have your subset of matched pos values. We will use the index numbers on the "chrom" and "strand" list value to pull out those values associated with the returned indexes. This will be compared in a for loop to the variables from the line from 3b. e. If there are no matches then the read will be kept and stored into the values of the inner dict at the same index. f. Then move onto the next line in the sam file till we have run through every file in the sam file.

(Side note Everything that gets added to the dictionary will be added to a new output file. Where as the things that match things within the dictionary will not be added to the new output file as those are duplicates.)

Functions:

add_cigar_to_pos():

Add the soft clipped ends to the beginning of the start position. This will be used for comparison with the positions in the UMI dict After comparison it would remove them so it adds the original if the position isn't found in the dict.

line_split_samline():

split a samfile line into the correct parts and then store those values in the multiple variables.