Write up a strategy for writing a Reference Based PCR Duplicate Removal tool. That is, given a sam file of uniquely mapped reads, remove all PCR duplicates (retain only a single copy of each read). Develop a strategy that avoids loading everything into memory. You should not write any code for this portion of the assignment. Be sure to:

- Define the problem
- Write examples:
    - Include a properly formatted input sam file
    - Include a properly formatted expected output sam file
- Develop your algorithm using pseudocode
- Determine high level functions
    - Description
    - Function headers
    - Test examples (for individual functions)
    - Return statement

For this portion of the assignment, you should design your algorithm for single-end data, with 96 UMIs. UMI information will be in the QNAME, like so: NS500451:154:HWKTMBGXX:1:11101:15364:1139:GAACAGGT. Discard any UMIs with errors (or error correct, if you're feeling ambitious).

Define the problem:

In RNA library prep, amplification is required to increase the amount of DNA in your library, especially if your yield is low. The amplification step is done through PCR that uses primers to bind to the template stand. This can result in an unbalanced proportion of read for each strand. Since the number of reads can affect the level of gene expression, duplicates need to be removed.

Purpose of deduper code:

Given a sam file we can use the alignment section as well as specific fields to remove the PCR duplicates. These fields include the RNAME, POS, and bitwise flag for specific strands. The RNAME provides us with the information of the chromosome. POS will provide us the location of the strand, and the bitwise flag will tell us if the reads are opposite strands. The CIGAR field will be used to search for soft clipping. Soft clipping is important to determine if the read is a duplicate by finding the position on the reference genome. Lastly, the QNAME field will be used for known UMIs or Unique Molecular Index.

Samples of inputed and outputted sam files is located in the repository

Functions

- Argparse
    - Parse sam, umi, and single-end/paired option
- Reverse Complement Checker Function
    - Inputs

        - Bitwise flag
    - Outputs
        - **True** if bitwise flag equals 16, if not return **False**.
    - Example
        - Input: 100
        - Output: Flase
- Soft clipping of the forward strand
    - Input
        - Chromosome location, Cigar string
    - Output
        - New chromosome location
    - Location is calculated by finding the first instance of “s” in the cigar string and subtracting the prior integer from the chromosome location value. Sum the new chromosome location value with the integer after “s”. If no “s” is present sum all integer with chromosome location value
    - Example
        - Input: 100, 3S20M
        - Output: 117
- Soft clipping of the reverse strand
    - Input
        - Chromosome location, Cigar string
    - Output
        - New chromosome location
    - Location is calculated by finding the first instance of “s” in the cigar string. If true, delete any integers prior and sum remaining integers with chromosome location. Then find any instances of I in the cigar string. If true, delete all integers prior until you reach a character. Remaining integers is summed with the chromosome location to return a new chromosome location.
    - Example
        - Input: 100, 3S20M4S
        - Output: 124
- Write out function
    - Inputs
        - Sam read line
    - Outputs
        - Writes to deduper file
    - Example
        - Input: `NS500451:154:HWKTMBGXX:1:11101:24260:1121:CTGTTCAC`
        - Output: deduper file

Pseudocode

1. Read in a sam file
2. Sort by chromosome and position
3. Pls refer to pseudocode_part2.pdf to get the rest of the pseudocode