Deduper

Part 1

Helena Klein

October 18, 2018

Purpose: Remove PCR duplicates given aligned sequences with unique molecule identifiers (UMIs). Begin with single-end, stranded reads then get progressively more complicated. PCR duplicates will have the same start position, chromosome alignment, and unique molecule identifier, with or without any soft clipping.

Part 1: The Plan

- Start by running through the file looking for the largest skipped regions (Ns) in the cigar string, and counting the number of entries per chromosome. These can be used later to make the most appropriate window size for later comparisons.
- Use samtools to sort each input dataset file by start position.
  - Write all the important information at the beggining of the SAM file to the ouput file without modifications.
- For each entry in the file, adjust the start position by the amount of soft clipping at the beginning of the read (subtract the soft clipping number (in CIGAR string, col=6) from the start position), add this modification to the end of the entry, and separate out the UMI from the QNAME and place in a separate category at the end of the entry. In addition, if the data are stranded, keep strand information.
- For entries without soft clipping, put the unmodified start position in the column at the end so they reads that are both modified and unmodified can be directly compared.
- Quality filter the UMIs. Make sure that they match one of the 96 provided, and do not have any undetermined bases. If they do not match the list of 96, then discard the read.
  - If we use random UMIs, should only filter out the undetermined bases.
- Use a shifting window approach to compare across the reads. There is only so much soft clipping that can occur at the beginning of any read, potentially no more than half the maximum read length, which could determine the size of the window in the absence of large skipped regions. A minimum of 50 entries at a time separated by column in a numpy array could encapsulate all the possible soft clipping for any start position in the file, without saving the entire file to memory. Otherwise, the window size needs to be as large as either the largest skipped region, up to the size of an entire chromosome.
- If reads are added to the end of a numpy array, then cycled through the first entry, either writing to a file if it the only one with that particular start position(POS, col=4), chromosome (RNAME, col=3), and UMI (QNAME, col=1) in the array or discarding it if the window has another entry with these identifiers (meaning there was some PCR duplication event here) then this could remove duplicates without loading everything into memory, with the largest amount of memory used the potentially the size of the largest chromosome.
  - In future iterations of this program, paired end reads, which are hopefully interleaved, could be easily incorportated into this model by doubling the window size for a similar single-end read file to hold twice as many reads, and the second UMI could also be compared across.
  - Anything modified should be at the end of the entry, which makes writing the entry to a file much easier

Functions

```
def UMI(QNAME):
    """Separates the UMI from the qname"""
    return UMI
```

```
def separate(filename):
    """Converts the raw unique molecule identifier file into an accessible format for later use."""
    return set_of_UMIs

def soft_clip(start_position, CIGAR_string):
    """Adjusts the start position given the amount of soft clipping. If there is no soft clipping,
    returns unchanged starting position"""
    return adjusted_start_position
```

Test Examples

```
UMI(NS500451:154:HWKTMBGXX:2:11107:25690:15614:ATCGTTGG)
ATCGTTGG

UMI(NS500451:154:HWKTMBGXX:2:11108:11325:4300:TGTGTGTG)
TGTGTGTG

separate(UMI_list_test.txt)
UMIS = {'AACGCCAT', 'AAGGTACG', 'AATTCCGG', 'ACACAGAG', 'ACACTCAG', 'ACACTGTG', 'ACAGGACA', 'ACCTGTAG'}

soft_clip(76886146, 70M1S)
76886146

soft_clip(52221729, 1S70M)
52221730
```

Test Files

Entries

Files should start out unorganized by starting position.

1. Unique starting positions and no soft clipping at start position (some with soft clipping at end of read)
2. Unique starting positions and soft clipping
3. Not unique starting positions and no soft clipping
4. Not unique starting positions and soft clipping
5. Not matching UMI to 96 known UMIs
6. UMI with an “N” in the sequence
7. Unmatching RNAMES with everything else the same

Output Key

Line of Test File	Test Result (corresponds to ‘Entries’ numbering)
25	1
26	3a (bad)
27	3b
28	1
29	2
30	2
31	2
32	4a (bad)
33	4b
34	2

Line of Test File	Test Result (corresponds to 'Entries' numbering)
35	2
36	5 (bad)
37	6 (bad)
38	7
39	7