Deduper Part 1

Laura Paez

10/15/2021

Strategy for Reference Based PCR Duplicate Removal Tool

A PCR duplicate is a strand of DNA that amplified more than expected. There are several reasons this can skew our data and make it more difficult to draw conclusions. In transcriptomics, the amount of expression is now reflecing not only the actual expression levels but the amplified PCR duplicates as well. Removing PCR duplicates can also minimize data we need to analyze. Regarding genome assembly, with PCR duplicates it is difficult to tell if a strand is a duplicate or if it is actually high coverage. With a reference genome, it is more feasible to identify PCR duplicates. Additionally, SAM files provide ways to identify whether strands are duplicates or not based on several factors. To start, the duplicates will have the same start position on the same chromosome of the same strand. What makes this tricky is that, the start position in the SAM file is based on whether soft clipping has occurred. Soft clipping is where the first or last several bases do not align and the start position is adjusted accordingly. In the SAM file, we can look at the CIGAR string to check for soft clipping, which we then use to adjust the start position. The new start position can be used to compare against other strands and find duplicates. Also, if the Unique Molecular Identifier (UMI) matches another read we can confirm they are PCR duplicates since only one molecule would have this UMI. In the SAM file, the UMI can be found in the QNAME. To retain a single copy of each read, we take the factors into account and can possibly create a set where there are no PCR duplicates.

Pseudocode

Python Script Part One: UMI Filter + Undo Soft Clipping

1. Reading line by line, if UMI is found in line, write line to new UMI-filtered SAM file. Else, continue (ignore that line).
2. Use pysam to sort new SAM file by chromosome name (RNAME)
3. Create a dictionary (chr_dict) that gets populated for one chromosome and then starts fresh when the next chromosome is reached.

- This dictionary holds chr_dict = {RNAME: POS, CIGAR}
- fixcigar function

3. Replace the start position to the new one from the chromosome dictionary.
4. Sort by start position using pysam to output the UMI filtered and soft clip-fixed SAM file

Python Script Part Two

Read in the UMI filtered and soft clip-fixed SAM file

5. Create a dictionary (output_dict) that gets populated for one POS and then starts fresh as the next POS is encountered. output_dict = {(UMI, FLAG): } The nature of dictionaries is to discard duplicates, so this will hold only the relevant lines we want.
6. After getting the full dictionary for one start position, I can rewrite the original start position in the line using chr_dict where the old start position is chr_dict[RNAME][0][1]. I can write the new lines (values of output_dict) out to a new final file, free of PCR duplicates.
7. Do this for all start positions of each chromosome.

High level functions

def fixcigar(chr_dict): '‘'This function takes the chromosome dictionary, extracts the CIGAR string, checks for 'S' indicating soft clipping, and adds the new pos (position) to the dictionary in a list with the old one' ' ' input: {RNAME:('15', '3S97M')} output: {RNAME: (['12', '15'], '3S97M')}