DeDuperAlgorithm

Trevor Enright

October 15, 2018

Part 1

Problem: PCR duplicates can mis represent expression analysis and other down stream analyses. By removing the PCR duplicates we can a better representation of what is happening biologically.

Solution: We use a reference-based approach. After alignment to a reference, the outputted SAM file will contain clues to which eludes to a PCR duplicate or not. If the following are the same position +/- soft clipping, chromosome, UMI, then we call a read a PCR duplicate. We will also be considering stranded to see if the reverse compliment needs to be applied.

General Workflow:

1) Sort the SAM file by chromosomal coordinates using samtools sort
2) Run python script on output of that file

General Algorithm

1) Loop through lines (reads) sorted SAM file
2) If UMI correction and umi list, correct UMI if hamming distance is within parameter specified by user.
3) Determine standedness with FLAG, Chromosome, and either right aligned (reverse stranded) or left aligned (forward stranded) position using position +/- CIGAR string information.
4) Keeping first unique read, write to output file, and store read in appropriate forward or reverse stranded dictionary, dictionary key is the position +/- CIGAR as described above and the value is the read line. If dictionary is not empty, check if PCR duplicate. If PCR duplicate then do nothing, else add to dictionary and write to output.
5) Purge dictionaries when position of current read could not be a PCR duplicate simply by being out of range of the sequence read length.

Functions:

```
def Forward_Parse_Cigar(string):
  pass
  #given a DNA sequence (string), return the Integer of the amount to SUBTRACT from the Position

  #sample input: "5S90M5S"
  #returns: 5
def Reverse_Parse_Cigar(string):
  pass
  #given a DNA sequence (string), return the Integer of the amount to ADD to the Position

  #sample input: "5S90M5S"
  #returns: 95
  #sample input: "5S85M1000N10S"
  #returns: 95

  #global variables
forward_dictionary = {} # dictionary to keep track of forward stranded read
reverse_dictionary = {} #reverse stranded reads

def dedupe(sam):
  pass
"""
  # given a sorted SAM file (string), loop through file and return a file without duplicates
  # for single-end only
  # writes files
  # returns status report
  with open(sam,"r") as fh:
    for line in fh:

      if line does not start with "@":
         access "dupeData" from the tab seperations, for UMI, current_position, chromosome, stranded, a



       if stranded is True: #forward stranded (flag does not have 16 flipped)
          #easy mode
          if forward_dictionary == {} : #empty
            #keeps first read
            #add the read and dupeData to the dictionary
            # write read to file

          else:
            #check if pos - left side soft clipping is a key in the forward_dictionary
            #check umi give or take hamming distance == umi in the value
            #check if chrom of the current read == chrom in the value
            #if checks are TRUE: PCR duplicate, remove (keep out of dictionary)
            #if FALSE: add to dictionary, write to output file
          #every max_length of read we'll purge the dictionary
          #purge forward_dictionary of positions < current_position - read_length

      else:
          #reverse compliment is True
          #hard mode

          if reverse_dictionary == {} : #empty
            #keeps first read
            #calculate where the read started to sequence from using CIGAR and position data
            #and use this to "right_align"
            #add the read and dupeData to the reverse_dictionary
            #write read to file

          else:
            #check if pos + deletions/matches/insertions/ right side soft clipping/splice is key in rev
            #check umi give or take hamming distance == umi in dictionary
            #check chromosome matches
            #if checks are TRUE: PCR duplicate (keep out of reverse_dictionary; do nothing)
            #if FALSE, add to dictionary, write to output file
           #every max_length read or so we'll purge the reverse dictionary
           #purge reverse_dictionary of reads with "right_align" < current_position + read_length
    else:
     #write header information to file
"""
```

Sample input

@HD VN:1.0 SO:coordinate @PG ID:GSNAP PN:gsnap VN:2017-10-12 CL:gsnap.avx2 –gunzip -t 26 -A sam -m 5 -d mm10_chroms -D /projects/bgmp/coonrod/mmu/INTEL -s /projects/bgmp/coonrod/mmu/INTEL/mm10_chroms/mm10
–split-output=/projects/bgmp/coonrod/deduper/gsnap//Datset1 /projects/bgmp/coonrod/deduper//Dataset1.fastq_dups.gz @SQ SN:1 LN:195471971 @SQ SN:2 LN:182113224 @SQ SN:3 LN:160039680 @SQ SN:4 LN:156508116 @SQ SN:5 LN:151834684 @SQ SN:6 LN:149736546 @SQ SN:7 LN:145441459 @SQ SN:8 LN:129401213 @SQ SN:9 LN:124595110 @SQ SN:10 LN:130694993 @SQ SN:11 LN:122082543 @SQ SN:12 LN:120129022 @SQ SN:13 LN:120421639 @SQ SN:14 LN:124902244 @SQ SN:15 LN:104043685 @SQ SN:16 LN:98207768 @SQ SN:17 LN:94987271 @SQ SN:18 LN:90702639 @SQ SN:19 LN:61431566 @SQ SN:X LN:171031299 @SQ SN:Y LN:91744698 @SQ SN:MT LN:16299 NS500451:154:HWKTMBGXX:1:22103:21539:19173:GTGATGTC 0 2 3286131 36 71M * 0 0 CCAGTTAAGAGGTTTCCAGATTTATTACACATCAGCACATTAATTATATATTAGGATGCTTAATCAAAATT EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU NS500451:154:HWKTMBGXX:1:22103:66244:99484:GTGATGT
0 2 3286131 36 71M * 0 0 CCAGTTAAGAGGTTTCCAGATTTATTACACATCAGCACATTAATTATATATTAGGATGCTTAATCAAAATT EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU NS500451:154:HWKTMBGXX:1:22103:84824:14967:GTGATGT
0 2 3286131 36 10S61M * 0 0 TTTAATTTTTGGTTTCCAGATTTATTACACATCAGCACATTAATTATATATTAGGATGCTTAATCAAAATT EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU NS500451:154:HWKTMBGXX:1:23109:26371:3459:GAAGACCA
0 2 3305412 36 28M4027N43M * 0 0 CAGCAGATTTCAGCAGGAGCAGCCATGGAATCCCACCATCCCTATTCAGAAGCTACAAGACATCCAGAGAG EEAEEEEEEEEEEEEEEEEEEEEAAEEAEEEEEA/EE/EEAEEEEEEEEEAE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU XS:A:+ XG:Z:A NS500451:154:HWKTMBGXX:1:23109:26371:3
0 2 3305412 36 28M4027N43M * 0 0 CAGCAGATTTCAGCAGGAGCAGCCATGGAATCCCACCATCCCTATTCAGAAGCTACAAGACATCCAGAGAG EEAEEEEEEEEEEEEEEEEEEEEAAEEAEEEEEA/EE/EEAEEEEEEEEEAE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU XS:A:+ XG:Z:A NS500451:154:HWKTMBGXX:1:13103:80160:6
16 2 181863757 36 71M * 0 0 CAGAGGATTAAAGAGGCAAGGAGTTTCAAAGAAATCTGAGAAGCCAGAGCTGGGAAATATAAATGTCTATA 6EAEEAEEEEEEEEAEAEEEE<EEAEEEEEEEEEEEEEEEE<EEEA6A6EEEEEEE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU NS500451:154:HWKTMBGXX:1:13103:67557:25136:AGCATGG
16 2 181863760 36 3S33M30M5S * 0 0 CAGAGGATTAAAGAGGCAAGGAGTTTCAAAGAAATCTGAGAAGCCAGAGCTGGGAAATATAAATGTCTATA 6EAEEAEEEEEEEEAEAEEEE<EEAEEEEEEEEEEEEEEEE<EEEA6A
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU

Notices:

This file has been sorted by coordinate Also this file has the first 5 reads as forward stranded and the last two as reverse stranded

Stepping through the meaty part of the algorithm:

1st read is forward stranded: the forward_dictionary is empty, add the read to the forward_dictionary write to output file 2nd read is forward stranded: the forward_dictionary is not empty, do PCR duplicate checks, checks conclude PCR duplicate: do nothing 3rd read is forward stranded: the forward_dictionary is not empty, do PCR duplicate checks, since adjusted position is in forward_dictionary and other checks are positive for PCR duplicate, do nothing 4th read is forward stranded: the forward_dictionary is not empty, do PCR duplicate checks, since adjusted position is NOT in forward_dictionary, add the read to the forward_dictionary write to output file 5th read is reverse stranded: the reverse_dictionary is empty, add the read to the forward_dictionary write to output file 6th read is reverse stranded: the reverse_dictionary is not empty, do PCR duplicate checks, since adjusted position is NOT in reverse_dictionary, add the read to the reverse_dictionary write to output file 7th read is reverse stranded: the reverse_dictionary is not empty, do PCR duplicate checks, since adjusted position is in reverse_dictionary, do nothing

Sample output

```
@HD VN:1.0 SO:coordinate @PG ID:GSNAP PN:gsnap VN:2017-10-12 CL:gsnap.avx2 –gunzip -t 26 -A sam -m
5 -d mm10_chroms -D /projects/bgmp/coonrod/mmu/INTEL -s /projects/bgmp/coonrod/mmu/INTEL/mm10_chroms/mm10
–split-output=/projects/bgmp/coonrod/deduper/gsnap//Datset1 /projects/bgmp/coonrod/deduper//Dataset1.fastq_dups.gz
@SQ SN:1 LN:195471971 @SQ SN:2 LN:182113224 @SQ SN:3 LN:160039680 @SQ SN:4 LN:156508116 @SQ
SN:5 LN:151834684 @SQ SN:6 LN:149736546 @SQ SN:7 LN:145441459 @SQ SN:8 LN:129401213 @SQ SN:9
LN:124595110 @SQ SN:10 LN:130694993 @SQ SN:11 LN:122082543 @SQ SN:12 LN:120129022 @SQ SN:13
LN:120421639 @SQ SN:14 LN:124902244 @SQ SN:15 LN:104043685 @SQ SN:16 LN:98207768 @SQ SN:17
LN:94987271 @SQ SN:18 LN:90702639 @SQ SN:19 LN:61431566 @SQ SN:X LN:171031299 @SQ SN:Y
LN:91744698 @SQ SN:MT LN:16299 NS500451:154:HWKTMBGXX:1:22103:21539:19173:GTGATGTC 0 2
3286131 36 71M * 0 0 CCAGTTAAGAGGTTTCCAGATTTATTACACATCAGCACATTAATTATATATTAG-
GATGCTTAATCAAAATT EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU NS500451:154:HWKTMBGXX:1:23109:26371:3459:GAAGACCA
0 2 3305412 36 28M4027N43M * 0 0 CAGCAGATTTCAGCAGGAGCAGCCATGGAATCCCACCATCCC-
TATTCAGAAGCTACAAGACATCCAGAGAG EEAEEEEEEEEEEEEEEEEEEEEAAEEAEEEEEA/EE/EEAEEEEEEEEEEAE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU XS:A:+ XG:Z:A NS500451:154:HWKTMBGXX:1:23109:26371:3
16 2 3305412 36 28M4027N43M * 0 0 CAGCAGATTTCAGCAGGAGCAGCCATGGAATCCCACCATCCC-
TATTCAGAAGCTACAAGACATCCAGAGAG EEAEEEEEEEEEEEEEEEEEEEEAAEEAEEEEEA/EE/EEAEEEEEEEEEEAE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU XS:A:+ XG:Z:A NS500451:154:HWKTMBGXX:1:13103:80160:6
16 2 181863757 36 71M * 0 0 CAGAGGATTAAAGAGGCAAGGAGTTTCAAAGAAATCTGAGAAGCCA-
GAGCTGGGAAATATAAATGTCTATA 6EAEEAEEEEEEEEAEAEEEE<EEAEEEEEEEEEEEEEEEEE<EEEA6A6EEEEEEE
MD:Z:71 NH:i:1 HI:i:1 NM:i:0 SM:i:36 XQ:i:40 X2:i:0 XO:Z:UU
```