Dedup

Rachel Richardson

October 22, 2018

Problem:

PCR duplicates can be a source of error for Illumina sequencing, especially considering RNA-seq differential expression analysis. PCR duplicates can occur disproportionally over reads, so there is a need to remove duplicates from the data. To do this, we will design a program to compare alignment read outputs (SAM files) in order to determine if reads are PCR duplicates. Important components inculde strand, start pososion of read, CIGAR string, and UMI or randomer.

Consider threading for forward/reverse?

Functions:

CIGAR reader - Reads CIGAR string for important characters and integers. Returns S, I, D, N, and other for appropriate CIGAR entries. To be used to determine reverse string start position.

FWDCLIP - Only takes soft clipping into account for forward strand reads. POS and CIGAR input. Returns adjusted POS

Window fill fwd - Fills forward window calling FWDCLIP function if S is in CIGAR string, checking for duplicates with each addition. Returns fwd window (list of saved variables).

Window fill rvs - Fills reverse window calling CIGAR reader function. Adds output to POS, then checking for duplicates with each addition. Returns rvs window.

Sliding window fwd - Takes fwd window. Reads in next line and calls FWDCLIP if S is in CIGAR string. Checks for duplicates in window. If none, writes window[0] to out and sets window as window[1:] + [newline]. At EOF, writes window to out. Returns null.

Sliding window rvs - Takes rvs window. Reads in next line and calls CIGAR read. Checks for duplicates in window. If none, writes window[0] to out and sets window as window[1:] + [newline]. At EOF, writes window to out. Returns null.

Test examples:

For CIGAR read function and FWDCLIP

Forward - soft clip (before M), no soft clip

Reverse - soft clip, insertion, deletion, splicing

Different UMIs (Non UMIs)

Different start points (POS)

Different chromosome (consider same start positions)

Header lines

For Window functions:

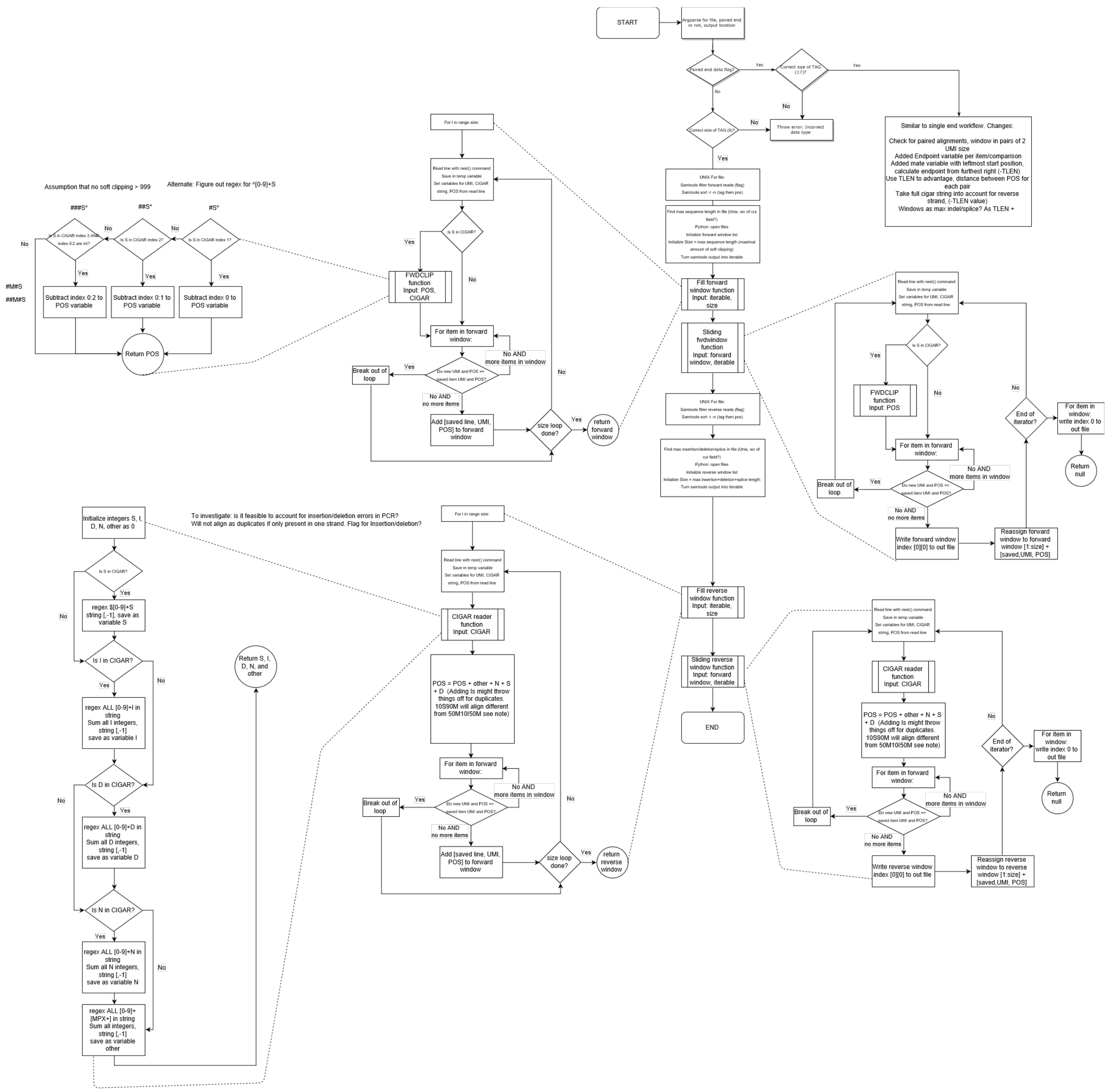
Comparison against splice samples large distances apart

Main function:

Paired end - error message for single??

A new kind of psuedocode:

Forgot to include in png: UMI check as reading lines, discards non-UMIs (Bit-wise correction?)


START
Argparse for file, paired end or not, output location
Paired end data flag?
Yes
Correct size of TAG (17)?
Yes
No
No
Correct size of TAG (8)?
No
Throw error: Incorrect data type
Yes
UNIX For file:
Samtools filter forward reads (flag)
Samtools sort -t -n (tag then pos)
Find max sequence length in file (Unix, wc of cut field?)
Python: open files
Initialize forward window list
Initialize Size = max sequence length (maximal amount of soft clipping)
Turn samtools output into iterable
Fill forward window function Input: iterable, size
Sliding fwdwindow function Input: forward window, iterable
UNIX For file:
Samtools filter reverse reads (flag)
Samtools sort -t -n (tag then pos)
Find max insertion/deletion/splice in file (Unix, wc of cut field?)
Python: open files
Initialize reverse window list
Initialize Size = max insertion+deletion+splice length
Turn samtools output into iterable
Fill reverse window function Input: iterable, size
Sliding reverse window function Input: forward window, iterable
END
For I in range size:
Read line with next() command
Save in temp variable
Set variables for UMI, CIGAR string, POS from read line
Is S in CIGAR?
Yes
No
FWDCLIP function Input: POS, CIGAR
For item in forward window:
No AND more items in window
Do new UMI and POS == saved item UMI and POS?
Yes
Break out of loop
No AND no more items
Add [saved line, UMI, POS] to forward window
No
size loop done?
Yes
return forward window
###S*
##S*
#S*
Is S in CIGAR index 3 AND index 0:2 are int?
No
Is S in CIGAR index 2?
No
Is S in CIGAR index 1?
No
Yes
Yes
Yes
#M#S
##M#S
Subtract index 0:2 to POS variable
Subtract index 0:1 to POS variable
Subtract index 0 to POS variable
Return POS
Read line with next() command
Save in temp variable
Set variables for UMI, CIGAR string, POS from read line
Is S in CIGAR?
Yes
No
FWDCLIP function Input: POS
For item in forward window:
No AND more items in window
Do new UMI and POS == saved item UMI and POS?
Yes
Break out of loop
No AND no more items
Write forward window index [0][0] to out file
Reassign forward window to forward window [1:size] + [saved,UMI, POS]
End of iterator?
No
For item in window: write index 0 to out file
Return null
Initialize integers S, I, D, N, other as 0
Is S in CIGAR?
Yes
No
regex $[0-9]+S string [,-1], save as variable S
Is I in CIGAR?
Yes
No
regex ALL [0-9]+I in string Sum all I integers, string [,-1] save as variable I
Is D in CIGAR?
Yes
No
regex ALL [0-9]+D in string Sum all D integers, string [,-1] save as variable D
Is N in CIGAR?
Yes
No
regex ALL [0-9]+N in string Sum all N integers, string [,-1] save as variable N
regex ALL [0-9]+ [MPX+] in string Sum all integers, string [,-1] save as variable other
Return S, I, D, N, and other
For I in range size:
Read line with next() command
Save in temp variable
Set variables for UMI, CIGAR string, POS from read line
CIGAR reader function Input: CIGAR
POS = POS + other + N + S + D (Adding Is might throw things off for duplicates. 10S90M will align different from 50M10I50M see note)
For item in forward window:
No AND more items in window
Do new UMI and POS == saved item UMI and POS?
Yes
Break out of loop
No AND no more items
Add [saved line, UMI, POS] to forward window
No
size loop done?
Yes
return reverse window
Read line with next() command
Save in temp variable
Set variables for UMI, CIGAR string, POS from read line
CIGAR reader function Input: CIGAR
POS = POS + other + N + S + D (Adding Is might throw things off for duplicates. 10S90M will align different from 50M10I50M see note)
For item in forward window:
No AND more items in window
Do new UMI and POS == saved item UMI and POS?
Yes
Break out of loop
No AND no more items
Write reverse window index [0][0] to out file
Reassign reverse window to reverse window [1:size] + [saved,UMI, POS]
End of iterator?
No
For item in window: write index 0 to out file
Return null

Similar to single end workflow. Changes:

Check for paired alignments, window in pairs of 2 UMI size
Added Endpoint variable per item/comparison
Added mate variable with leftmost start position, calculate endpoint from furthest right (-TLEN)
Use TLEN to advantage, distance between POS for each pair
Take full cigar string into account for reverse strand, (-TLEN value)
Windows as max indel/splice? As TLEN +

Assumption that no soft clipping > 999

Alternate: Figure out regex for ^[0-9]+S

To investigate: is it feasible to account for insertion/deletion errors in PCR?
Will not align as duplicates if only present in one strand. Flag for Insertion/deletion?

Psuedocode Flow chart