Report

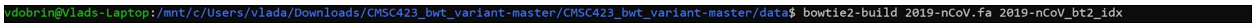
How to use:

1. User will be prompted to enter which command they want by selecting either a 1 or a 2 as indicated by the statement
2. The user needs to enter the appropriate locations of required files as prompted with \\ slashes because otherwise, java will interpret the \ slash as part of the code
    a. Depending on the tool chosen the user will have to enter a different amount of file locations as requested by the documentation of the project.
3. The user will then get the necessary files outputted to the specified locations upon completion of the program.

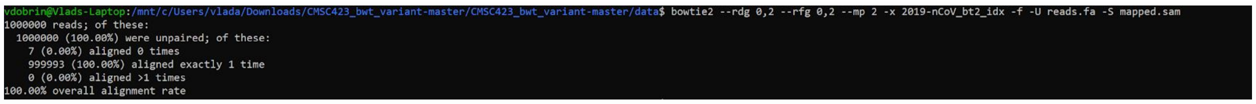
The code is heavily commented to help with what is going on.

Discovered Variants

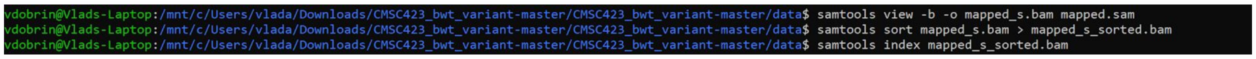
These were discovered using Bowtie 2 because I could not get the header to properly function, the rest of the SAM file was nearly identical as compared to the provided smaller Reads.fa file. Commands Run was as follow:



```
vdobrin@Vlads-Laptop:/mnt/c/Users/vlada/Downloads/CMSC423_bwt_variant-master/CMSC423_bwt_variant-master/data$ bowtie2-build 2019-nCoV.fa 2019-nCoV_bt2_idx
```



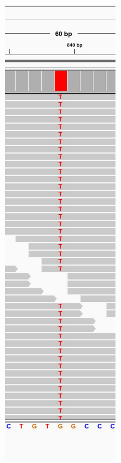
```
vdobrin@Vlads-Laptop:/mnt/c/Users/vlada/Downloads/CMSC423_bwt_variant-master/CMSC423_bwt_variant-master/data$ bowtie2 --rdg 0,2 --rfg 0,2 --mp 2 -x 2019-nCoV_bt2_idx -f -U reads.fa -S mapped.sam
1000000 reads; of these:
  1000000 (100.00%) were unpaired; of these:
    7 (0.00%) aligned 0 times
    999993 (100.00%) aligned exactly 1 time
    0 (0.00%) aligned >1 times
100.00% overall alignment rate
```



```
vdobrin@Vlads-Laptop:/mnt/c/Users/vlada/Downloads/CMSC423_bwt_variant-master/CMSC423_bwt_variant-master/data$ samtools view -b -o mapped_s.bam mapped.sam
vdobrin@Vlads-Laptop:/mnt/c/Users/vlada/Downloads/CMSC423_bwt_variant-master/CMSC423_bwt_variant-master/data$ samtools sort mapped_s.bam > mapped_s_sorted.bam
vdobrin@Vlads-Laptop:/mnt/c/Users/vlada/Downloads/CMSC423_bwt_variant-master/CMSC423_bwt_variant-master/data$ samtools index mapped_s_sorted.bam
```

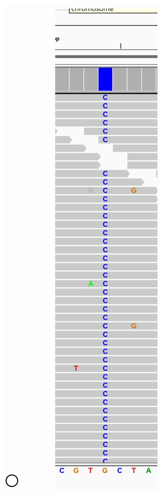
Then imported into IGV as instructed from the project file, and the following mutations were observed. Hopefully, none of which if actually enacted would cause COVID-19 to be more contagious or deadly.

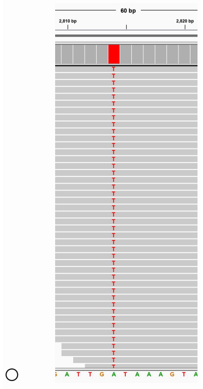
- T instead of G at 839 bases

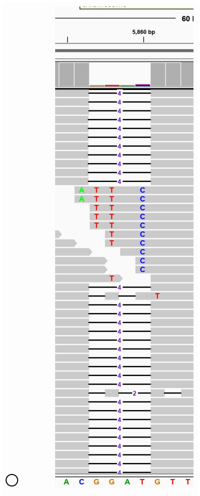


    ○

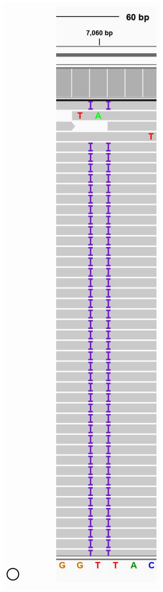
- C instead of G at 1544 bases

- 

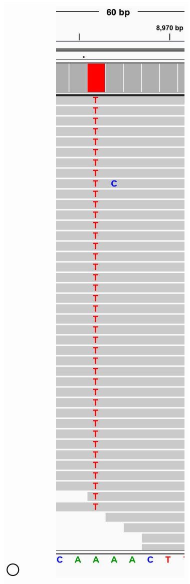
- T instead of A at 2814 bases
  - 

- 4 base deletion of GGAT around the 5860 base mark
  - 

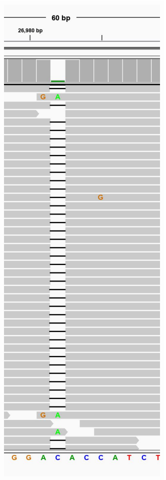
- T insertion around the 7060 base mark

- ○ 

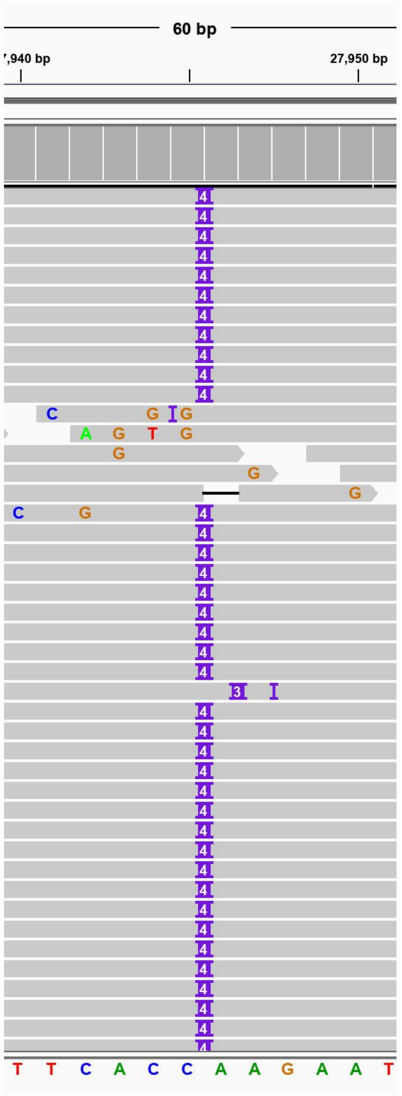
- T instead of A 8966 bases

- ○ 

- Deletion of a C around 26, 980 bases



- 4 Base insertion cannot tell what the insertions were from IGV. Occurred around 27, 945 bases



My evidence for all the variants was the large sets of specific colors for various changes. They extensively stood out from the other random errors because they all aligned perfectly with one another as shown in the above images. The random errors or misreads showed up as spots that littered the reads rather obnoxiously. The reads could also be seen as aligned as light gray boxes with white in-between. The point mutations would not cause much of a problem for the virus as the degeneracy of the amino acid code would lead to the same reading frames for translation into proteins with variations in the codon. The larger errors such as the 4 base insertion and 4 base deletion would cause a much larger problem since those are significantly impacting the reading frame of rRNA. The deletion/insertion could also have the same impact since it offsets the reading frame by one base pair. Rather interestingly T errors seem to have been the most prevalent with it accounting for 4 of the mutations out of the 8. Work by Kimsey et al published here cited here: Isaac J. Kimsey et al, Dynamic basis for dG•dT misincorporation via tautomerization and ionization, Nature (2018) supports this. The work indicates that while a T-G base pairing is not preferred over regular A-T, G-C base pairing, it can occur as it is energy efficient enough for nature to allow it to happen. This error normally occurs in DNA so that may explain why we did not see an extensive amount of it (well over 50%) since COVID-19 is an RNA based virus (Gorbalenya et al, 2020). The poly-A tail of the virus is also rather short at only about 12 A's. This would indicate that it survives a low amount of time in an open-cell environment before it is degraded (BSCI222 Straney Spring 2020).

Design

There were a few design decisions that I took based on the advantages that java provides, but a lot of the disadvantages of java are explored secondly when looking at what changes I would make. I decided to return a fmIndex object from my first method call because it would make everything easier to access all the stored information rather than attempting to take it from the file and rewrite it back to the appropriate data structures. This saved me a lot of time! I also decided to allow a user to input the file locations they wanted because they could enter it as a string and then it could all easily be saved as necessary for future use. This would allow a user to easily call any tool depending only on the location of the reads and the genome itself. I reused a lot of other code that we had designed for Rosalind and found that that helped a lot with preparing the first task.

There are some changes that I would make if I started this project again. I would learn python because java is not a great language to do text manipulation in, this is because of its object-oriented nature. My code is littered with calls to other methods in which I need to open files and access text information in extremely "laborious" ways. My limited experience in python tells me that this would have been a lot easier in python. In terms of design with the tool, I would not change much other than trying to reuse a lot more of my code, I had to rewrite some large chunks twice to allow for compatibility with other objects that were being passed around. This was done because it was easier to make a new class rather than try to figure out object compatibility. Given more time I would also tidy up the align function with more method calls and reducing the redundancy of the code. It was done in this manner because of the time constraint and the direct ability to debug without needing to worry about passing objects around and ensuring that they would have the correct return values. Furthermore, the code itself has the instructions for use at the top but if the user does not appropriately use the tool it will not function. I would work to improve the tool to have more try catches and deal with other exceptions.

Roadblocks

Oh boy, this project led to lots of roadblocks!
The first roadblock that was hit was a minor one, and that was returning values from method calls and ensuring that any parsed information was appropriately assigned to variables for future use.
The second roadblock was parsing the Fasta file. I have never used git before and had no experience in how to add a git library as an import. The same issue arrived when working with SAM writing. I got over this by writing my own FASTA parser as recommended by Dr. Patro and that was much simpler!
The third roadblock was how to store information, I played around a lot with how to send information to the output file but eventually was able to convert all my data structures to string and use file output commands to easily store them.
The fourth roadblock was moving into the align function. I was stuck with the FM index backtracking first because I overlooked the fact that I needed a rank array that could assist me. That was a major stepping stone toward getting it to work. I then ran into trouble on how to exactly use said ranks for my benefit in calculating the range for my fmIndex backtrace. I was able to fix this with a newly learned data structure called a wavelet tree. I found help on how to build this tree type on the internet and it was super beneficial in completing the backtrace with ranks.
The fifth roadblock occurred with dynamic programming and backtracking. This was because the format of left, up and diagonal was not making sense as insertion and deletion into the reference. I asked on Piazza and that was cleared up. Another problem was the type of fitting alignment to use because four modes were given. I again asked on Piazza and had it cleared up.
A major roadblock that occurred in this process was the SAM file. I did not understand what it was, how to write it and what was expected in it. Luckily, Dr. Patro took time out of his Spring Break to have a Zoom meeting and explain everything again. After this, I was able to fly on the align function over the next few days and wrap most of it up.
Smaller issues like index off by one occurred numerous times but these were corrected by debugging.
The only bug I was not able to fix was the SAM header. I did not know what was expected for each component, I tried copying over what Dr. Patro did in his example but did not know what to change the bowtie information to.

Code Validity and Testing

In this section, I will explain how I know that my code worked.

The name and reference genome that was put in the output file were character compared using an online tool. The occurrence table created was correct based on the total count of the alphabet is the same as the length of the and an in-depth debugging procedure. The suffix array table was only able to be tested with the align function and intended output. The FM index was tested against the smaller reads, and I manually tested a few by ensuring that the 100M reads that existed in the reference genome. This was easily done with a ctr+f to ensure I was looking at the correct ones. To ensure the backtracking algorithm was working I first used small sample data sets that I created on random strings, I manually confirmed these worked before going to the 10000 read file given. Here I first confirmed all my 100M for perfect FM index search matched before going to others. Then in order to test the backtracking algorithm, I created a print for the DP table and printed out the alignment to see what they looked like to ensure it was working as intended. I then proceeded forward and used the 10000 read as a test file and ensured that it worked for all processes. The CIGAR was similar for about 95% of the reads and the spot in the genome was always correct.