**Bioinformatics 529 Homework 2**

**Amelia Lauth**

**02/20/22**

**Resources used (Including websites, partners in class, etc.):**

Partners in Class: Crystal Wen, Matthew Pun, Noah Helton, Tutor Catherine Barnier, Brad Crone

<https://www.mat.univie.ac.at/~kriegl/Skripten/CG/node44.html>

<https://iq.opengenus.org/run-length-encoding/>

<https://stackoverflow.com/questions/43244825/getting-last-element-of-range-in-python/43244842>

<https://note.nkmk.me/en/python-check-int-float/>

<https://stackoverflow.com/questions/11059910/skipping-elements-in-a-list-python/11059982>

<https://www.youtube.com/watch?v=GWFb_C4IR14&t=775s>

<https://www.youtube.com/watch?v=0qMGAsrYS0g>

<https://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/bwt.pdf>

<https://bamnostic.readthedocs.io/en/latest/_modules/bamnostic/utils.html>

Class\_07 BWT PTT, video, class code, class code solutions, Class\_08 class code, class code solutions, Class\_09 HW code, video, class code, class code solutions **(pileup function in the class\_09 solutions was used in hw\_02) I used the pileup function since BAM files are not human readable and I’m not familiar with BAM.**

**Statement of Objective (What was the purpose of this homework assignment):**

We explored the BWT, reversing the BWT string back to its original string and performing run-length encode and decode functions to compress and decompress strings. We also focused on INDEL identifications regarding matches and deletion states in sequences, utilizing the package bamnostic to decode a CIGAR string and build a pileup from the CIGAR string rather than a sequence.

**Procedure (Explain in general terms how you went about implementing the homework assignment):**

I performed a ton of google searches, revisited class videos and PPTs, looked through available PPTs found online, and thoroughly explored bamnostic documentation on both GitHub and the official online documentation website. I decided to use a parse cigar method found on bamnostic documentation to complete the decode\_CIGAR function more efficiently. I also had to create multiple versions of the get\_cigar\_pileup function to check edge-cases. Manually created deletions in the SAM file from class, converted it to a BAM file, and used file as a test.

**Difficulties and Roadblocks (What were the pain points in the implementation of this homework assignment):**

The reverseBWT function was very difficult, and I tried multiple approaches from counting the occurrences of each character to see which ones had more than one occurrence, to trying to slice the columns and keep track of which character repeated itself or not. After google searching, a PPT explained a far easier way of rebuilding the entire matrix, adding the BWT string to an empty matrix, sorting each time around in the loop, and finally taking string in a row that began with the “$”. But then I struggled with how to make the for loop, since I had the for loop outside of the sorted command. I realized you have to keep the for loop inside the sorted method for this function to work, since you are sorting each time you add the BWT to the matrix. The second roadblock was with the pileup function, where I was confused if I was to either include or exclude the “-“s from the pileup. Since the docstring said we’re building the pileup from a CIGAR string, and the string may contain deletions, and deletions won’t count as allele frequencies (which we were returning per the docstring) I chose to skip over “-“s and exclude them from the Counter, but also not lose the correct index in the sequence. I hope this was the correct interpretation of the get\_cigar\_pileup function docstring.