Assignment Report

Paarth Gupta
15477

In this assignment we try to find out the most probable configuration (of out of the 4 discussed in class) that leads to color blindness.

I have tried to follow the similar approach discussed in the class. We have 4 characters A, C, G and T so for each one of them we create a binary column from the BWT column.

Now I have use the concept of Δ- milestones (with $\Delta = 100$). So to keep a Δ- rank milestone array we need $\frac{\Delta}{n}$ bytes and for 4 chars we need $\frac{4\Delta}{n}$ bytes.

This is a one time process done so that we can obtain the rank of a character in $O(\Delta)$ time.

Now our aim is to find for each read either a perfect match or a match with at most 2 mismatched characters, using the rank array. We try to recursively search for the range of indices in which the suffix has an exact match.

Now we note down the count of these matches in 2 different sets i.e Red and green and further in them on which number they lie.

Finding probable configuration :-

eg for Red : $[R_1 \quad R_2 \quad R_3 \quad R_4 \quad R_5 \quad R_6]$
Green : $[G_1 \quad G_2 \quad G_3 \quad G_4 \quad G_5 \quad G_6]$

$$P(\text{(Red, Green)} / \text{Config1}) = \prod_{i=2}^{\epsilon} \binom{R_i + G_i}{R_i} P_{\text{config1}}^{R_i} (1 - P_{\text{config1}})^{G_i} \quad -①$$

Running 3M reads was not possible system was crashing + Google Colab was also crashing. Results are obtained by partitioning and that too only an around 1.7 M reads (high density), 1M from the back and 0.7 M from the front. After analysing that backside contains much more matches.

It is observed that all the exons are present after 2.8M only.

$R = [97 \quad 237 \quad 107.5 \quad 167.5 \quad 303.5 \quad 235]$
$G = [97 \quad 287 \quad 154.5 \quad 140.5 \quad 358.5 \quad 235]$

Now using equation 1 we get :-

$P(R, G / \text{Config 1}) = 2.1294 \times e^{-44}$
$P(R, G / \text{Config 2}) = 0$
$P(R, G / \text{Config 3}) = 1.266 \times e^{-36}$
$P(R, G / \text{Config 4}) = 3.0427 \times e^{-62}$

$\log P_1 = -100.55$
$\log P_2 = -\infty$
$\log P_3 = -82.657$
$\log P_4 = -141.647$

Maximum likelihood is obtained on Config 3. i.e. [33% 33% 100% 100%

Solution for Select query

Select query :- to find the i^{th} 1 from a binary array.

- Considering that we have a array of size $\frac{n}{\Delta}$ which stores Δ-rank milestones and will contain the indices of the corresponding ranked 1 in the original binary array.
  Now as we have this Δ-rank milestone array, if we are given i using this Δ-rank milestone array we can directly jump to the index in the original array till we reach i^{th} (desired) 1.
  With the i in hand, we can use the binary search as the array is sorted to find the correct location in $O(\log n)$ time.
  If we analyse the technique we can observe that traversal to the required 1 may take more time than $O(\Delta)$ but in amortized setting we say that its $O(\Delta)$ ∴ Also to improve this time complexity we use another array of size $\frac{n}{\Delta}$ which will contain a reference to the next one.

This setting has a time complexity of $O(\Delta)$ and a total space complexity of $O(2\frac{n}{\Delta})$