Lubhawan Parihar | SR.No.: 06-18-01-10-51-21-1-19239

E0 259 Data Analytics 2022

Assignment - 4 (Color Blindness)

Given the following:

(a) reads from a genome – 3 million reads of the 150m we actually generated,
(b) the reference sequence of chromosome X – 150m instead of 3b for the whole genome,
(c) the BWT last column and the pointers back to the reference for chromosome X,
(d) the locations of the exons of the red and green genes in chromosome X,

Explanation:

1. **def loadLastCol(filename):**
Used to load last column as a sub string and return the whole string as joined substring.

2. **def loadRefSeq(filename):**
Used to load sub string and return joined substring as a single string of whole DNA seq length.

3. **def loadReads(filename):**
This function loads the reads in string format and also used to pre process reads and replace all the 'N' with 'A' character.

4. **def loadMapToRefSeq(filename):**
This function loads the index of first column which maps them to Reference sequence.

5. **def MatchReadToLoc(read):**
This Function take a single read and divide it into 3 equal parts and try to match and try to match a single part with original sequence and if any one-part matches than it takes the reads position as probable position at which the read may match the reference sequence.

6. **def WhichExon(positions):**
This function takes the positions of possible match and returns the no of count of Red Exon and Green Exon match which matches at the provided Red and Green Exon locations for all the 6 positions.

7. **def ComputeProb(ExonMatchCounts):**
It computes the probability of all the given 4 cases from slide:
$\text{case1} = [0.5,0.5,0.5,0.5]$
$\text{case2} = [1,1,0,0]$
$\text{case3} = [0.33,0.33,1,1]$
$\text{case4} = [0.33,0.33,0.33,1]$

8. **def BestMatch(ListProb):**
This function returns the case no for which the probability is maximum

9. **def Rank_delta(LastCol): - > created function**
This Function takes the Last col and for the characters A,T,C,G it creates the count matrix for them after each N counts of LastCol matrix. For my case I have taken N=1000.

10. **def reverse_complement(s): -> created function**
If the original read failed to meet to the reference sequence this function takes the reverse complement of original read and then return it for matching to **MatchReadToLoc(read) function.**

Result:

My machine was taking too long time to run on whole sequence, so I just tested the program on limited reads only. So, I can't say the expected case that leads to color blindness.

From slides my intuition is that that the 3rd case to color blindness i.e.: