E0 259 Assignment-5

Aditya Gupta
SR No: 22205

Implementation Summary

We are given about 3 million reads of length 100 each from the X chromosome. We are also given the reference sequence of the X chromosome, whose length is about 150 million. We have to match the reads to the reference sequence, with a maximum of two mismatches allowed. Then we have to find the number of reads corresponding to the red and green exons.

Instead of finding the substring directly in the reference sequence, we will use the last column of the Burrows-Wheeler Transform (BWT) of the reference sequence. To match a read, we will read it from right to left and keep finding the first and last occurrence of the character in the BWT. This will be done by keeping track of the rank of each base.

The ranks will be stored in a data structure called SuccinctRank. This data structure will have 4 columns (corresponding to each of the 4 bases) where the rank of every base at every 32^{nd} index will be stored. This is done to save memory since our RAM is limited. Using this data structure, we can quickly calculate the ranks of the first and last occurrence of the character in the BWT.

To select the corresponding range of the character in the first column, we need to select the rows corresponding to it. However since the last column is alphabetically sorted, we simply need to shift the range by the no of characters in the reference sequence that are lexicographically smaller than the character. Then we select the last column of the shifted range and repeat the process for the next character. Doing this for all characters of the read would give us the location of an exact match of the read in the first column. Since we are given a map from the first column to the reference sequence, we can easily find the location of the read in the reference sequence.

The above process finds exact matches. Since we are allowed to have two mismatches, we will split the read into three equal parts and match them separately. If the total read has less than three mismatches, one of these three subreads will have an exact match (by the pigeonhole principle). Thus we get the indices of exact match and then directly compare the remaining read with the reference. If the full read has less than 3 mismatches with the reference at that index, we will consider it a match. We try to run this for every possible match for every subread and keep track of the number of matches. If the subreads in a read have no exact matches, we will try to run this process for the reverse complement of the read.

Once we have obtained all the valid matches, we will check which of these lie in the red and green exon ranges. If there is a match only to one of red or green exon, we will add 1 to the corresponding exon number. If there is a match to both red and green exon, we will add 0.5 to both the corresponding exon numbers. We do this for all the reads and obtain the final counts of the number of reads corresponding to the red and green exons. The approximate runtime was about 40 minutes.

Results

On matching the reads to the exons, we obtain the following results:

Colour	1	2	3	4	5	6
Red	97	143	85.5	160.5	279.5	235
Green	97	193	132.5	133.5	334.5	235

Table 1: No. of reads per exon for red and green exons

We know 4 possible configurations of the exons, and want to find which one causes colour blindness. Let us calculate the probability of each of these configurations using the counts we have. We will only be using exon 2, 3, 4 and 5 for this calculation, since exon 1 and 6 are the same for both red and green.

We will calculate the logarithm of the probability of the configuration j by using the KL divergence metric, which in this case is given by:

$$\log(P_j) = \sum_{i=2}^{5} r_i \log(C_{i,j}) + g_i \log(1 - C_{i,j})$$

where r_i is the number of reads corresponding to the red exon i, g_i is the number of reads corresponding to the green exon i and $C_{i,j}$ is the probability of exon i for the configuration j.

The configurations given to us are [50%, 50%, 50%, 50%], [100%, 100%, 0%, 0%], [33%, 33%, 100%, 100%] and [33%, 33%, 33%, 100%]. Using this, we can construct the class probability and get the following log probabilities for each configuration:

Configuration	Log Probability
1	-1058.92
2	-inf
3	-1039.78
4	-1096.90

Table 2: Log probabilities of each configuration

From the above table, we can see that the configuration 3 has the highest probability of being the cause of colour blindness.