Author: Preeti Singh, Graduate Student, Carnegie Mellon University

Title: Documentation for Programming Interview Question at Bayer, Davis

Aim:
1.The total number of variants in each individual file which can be found within the ranges given
2.The number of within-range variants overlapping (aka intersecting) across all files. Rephrased another way: How many within-range variants do all 3 input files have in common?

Assumptions: Following assumptions were made by me for solving the programming question:

1. Each input file is sorted first by chromosome then by position
2. Size of Range File is small and input files are big
3. No stray lines in the input files

Algorithm:

2. To find the number of variants in each file and find the overlapping
2.1 Read the range file(ranges.txt) and load it into the memory using lists as the datastructures
2.2 Now read all three input files such that each has a marker pointing to the first variant position.
2.3 compare first variant position of file1 with the first range given in ranges.txt
2.4 If this variant position lies within-range then increase variant-count for that particular file
2.5 Else increase compare this variant-position with next range
3. To check if these variants overlap between three different input files find the file index of variant having minimum value of variant position
3.1 While reading the first variant position, compare the variant position at that index
3.2 find the file index of variant having minimum value of variant position
3.3 check if the variant position is in range
3.4 if yes then check if they are overlapping

Runtime of the Algorithm:

1. It's O(n*r), where n is total size of inputs and r is size of range file

Optimization step:

We could probably reduce it to n*log r using some more efficient search (e.g., BST) for the range file.