Title: Calculating genomic coverage from next-generation sequencing data

Problem Definition: Calculate read coverage at a number of positions of interest ("loci")

Approach that were taken

Approach 1:

1. Loop through the entire reads.csv file for every position in loci.csv file and check the coverage for that position
2. The time complexity for this approach was length of data present in loci.csv time length of data present in reads.csv file

O(n\*k)

Where n = length of data in loci.csv and k = length of data in reads.csv

1. This approach takes time to parse millions of data

Approach 2:

1. Pre-process the data of reads.csv and stored it in dictionary (Map interface) where key is pattern which is sliced from the start position of DNA.

For instance if start is “101843359”, so key stored is “101843” which is seen as a common pattern in entire reads.csv

The value is set of start position and maximum length of base pair

Example:

Start Length

101843359 151

Then dictionary would be {“101843”:[(“ 101843359”,” 101843510”)]}

1. A second dictionary is used to store the count of this sit(“ 101843359”,” 101843510”). This count is used to add while computing the coverage
2. For computing the coverage for a position, we get a list of the pattern and from dictionary we return the list and check if position lies in that coverage and increment the count value

Merits of approach 2:

1. We are pre-processing millions of data and storing it in dictionary. The time complexity of getting a key from dictionary is O(1) and putting a value is also O(1)
2. Thus the search complexity is drastically reduced comparing to Approach 1

Time complexity for approach 2:

1. Let “n” be size of data in reads.csv, ”k” be size of data in loci.csv
2. Time taken to read entire reads.csv file is O(n)
3. Time taken to read loci.csv file is O(k)
4. Time to write a row in csv is constant
5. Time to compute is size O(k\*m) where m is length of dictionary value list for that position

Total time complexity for average case is O(n + k + k\*m)

Worst case time complexity is O(n) where n is time required to read “read.csv”

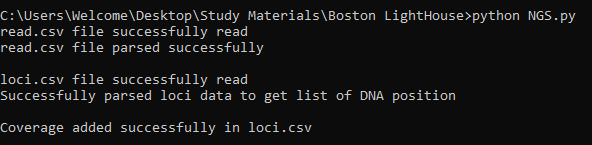
Pre-requisites:

1. Install python on machine to run python program

Steps to run:

1. Copy the python files to any directory
2. The reads.csv and loci.csv must be present inside “data” folder and data folder must be present where all python files are present
3. To run the NGS.py on cmd type python NGS.py/python3 NGS.py as per python version installed
4. If you want to run the python script on IDE then skip step 3 and install IDE of your preference run he NGS.py file

Output:



Best Approach (Future scope) to improve performance:

1. Find a pattern with the data and analyse a formula to provide us the coverage for that particular position which would help us get solution in lesser time compared to Approach 2.
2. In this approach the time complexity will just depend on time required to traverse number of positions in loci.csv file.