Assignment #2

APA - Master in Bioinformatics for Health Sciences

October 3^{rd}, 2019

Finding spatially close reads falling under the same genomic location

Say, Alumina, a company specialized in characterizing genetic sequences for biomedical discovery, has recently launched a new hybrid sequencer, called OptiXseq. OptiXseq combines the power of advancements in high-throughput DNA sequencing (such as droplet based microfluidics) with chromosome confirmation capture (such as Hi-C).

Challenged by the recent findings suggesting that genome topology (spatial organization of chromosomes) has only minor influence on gene expression (Ghavi-Helm et al., 2019), we are interested in investigating the relationship between copy number variation and genome topology using OptiXseq. Accordingly, one would like to analyze the number of reads that are both sequentially and spatially in close vicinity across different samples coming from individuals.

For each sample, OptiXseq provides the following info on the sequencing reads corresponding to the DNA from an individual in a single file:

- Sequence id
- Genomic loci
- Spatial position

The content of an example file is as follows (note that the reads are produced in an arbitrary order):

Read	**Loci**	**Coordinates**
seq1	5p14.1	(2,3)
seq2	8q11.2	(3,6)
seq3	2q9.4	(1,4)
seq4	5q7.3	(3,4)
seq5	11p4.1	(2,5)
seq6	2q6.2	(1,5)
seq7	11p2.3	(1,6)

Task description

- Given the input file containing the information on the sequencing reads, the objective is to find the number of pairs of reads (seq_i and seq_j) that satisfy the following two conditions:
  - seq_i and seq_j reside on the same chromosome arm
  - $d(seq_i, seq_j) \leq k$, where $d(p,q)$ for two points p, q in 2D space is defined as

$$d(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2}$$

- Write a python3 code that parses the input file and stores the information from the reads in a **linked list** (each item of the list corresponding to a read).
- To facilitate the identification of the reads that fall under the same chromosome arm, sort the list using **insertion sort** using the loci information.

- Traverse through the sorted list to group reads that fall under the same chromosome arm and iteratively check pair of reads in regards to whether they satisfy the spatial restriction above (for a user defined k).
- Please note that **no additional container** (such as list, array, dictionary, etc. other than the implemented linked list) can be used to store data (i.e. coordinates of reads on the same chromosome arm or number of close by pairs).
- Write the information on the pair of reads that are close by for each chromosome arm to a file (see specifications below).
- Time your implementation using **time** command for the provided input files.

Output specifications and evaluation criteria

The format of the output files is as follows (note that the information on the pairs of reads on the same chromosome arm are ordered *lexicographically* based on the chromosome arm):

Location	**Number**
2q	1
5p	0
5q	0
8q	0
11p	1

Note that the output above would be produced given $k = 1.5$. If $k = 1$ the output would be:

Location	**Number**
2q	1
5p	0
5q	0
8q	0
11p	0

- The python file to be executed (main.py) should accept two parameters
  - $< input\_file >$: Absolute path of the input file
  - $< k >$: The value of the spatial threshold k
- The code will be evaluated based on the correctness of the output for various inputs (sample files and k values), satisfaction of requirements above and the 'quality' of the code.
- Please generate easy to read code (meaningful function names capturing essential elements of your implementation (i.e., generating and traversal of linked lists, insertion sort procedure, iteration over reads on the same chromosome arm, etc.).
- It is important that the restriction on the (non-)use of additional container objects (such as list, array, dictionary etc.) is respected.
- In the case of not having a correctly working implementation, to obtain partial credit, please produce an output that shows your progress (e.g., printing the linked list with the input data / not-fully correct output data).

Deadline and submission instructions

- Upload your code to a repository named **APA** under a folder called **Assignment2** in your *GitHub* account by **October 15th**.
- Keep in mind that the following command will be used to clone your code:

  ```
  $> git clone https://github.com/YourUserName/APA.git YourUserName
  ```

- Your repository's *README* file should include a section containing the output of the **time** commands for example input files provided online.
- The name of the python file to be run should be **main.py** for the testing script work correctly. The input parameters will be given as is (without any option specification) as follows:

  ```
  $> cd YourUserName/Assignment2/
  $> python main.py /home/apa/Assignment2/test/input1.txt 1.5
  ```