#3: Minimum fragment size for successful genome reconstruction

Izabela Dusza (70083252)

January 18, 2020

1 Introduction

Aim of the project is to determine the minimum length k of a fragment (*k-mer*) that is required to correctly and unambiguously reconstruct a given circular genome.
Genomes for the fragmentation to k-mers are saved in file `genomes.txt`.

2 Solution

In first step, according to description of exercise, `composition(G, k)` function was implemented. It returns a list of k-mers of a circular genome G that "wrap around" the genome sequence ends. For example:

```
composition("ATACGGTC", 3)
>>> ["ATA", "TAC", "ACG", "CGG", "GGT", "GTC", "TCA", "CAT"]
```

Then, each *k-mer* has been divided into **prefixes** and **suffixes**. Prefix and suffix refer to the first $k-1$ nucleotides and last $k-1$ nucleotides of a k-mer, respectively. For the purpuses of implementation only unique suffixes/prefixes were needed (redundant ones were removed).

In the final step de Bruijn directed graph was implemented and at the end, the recursive reconstruction function. In de Bruijn graph, k-mers are assigned not to the nodes but to the edges. Therefore, solving the String Reconstruction Problem reduces to finding a path in the de Bruijn graph that visits every edge exactly once – so called Eulerian Path.

3 Results

For each genome, following minimum lengths k of mers was obtained:

Genome name	Genome length	**min** k
Homo sapiens S100A4	306	9
Danio rerio VAMP2	333	10
Bos taurus RPL39	156	8
Canis lupus familiaris RPL35	372	10
Mus musculus MNS1	1476	14

As it is shown in the table above, the length k of mers depends on length of genome itself – for longer genome chains the minimum k was bigger.

3.1 Conclusion

Plenty of ways for assembling genome of k-mers exist. It can be done not only by solving Eulerian path problem, but also for example a Hamiltonian path, which can be tricky and is not reccomended.

I chose a recursive way of solving the Eulerian path problem, but it was also possible to solve it iteratively, which is probably more efficient but can be less intuitive for some people. As it is typical to the recursive methods, using it on a bigger data, for example running it on the longest genome (*Mus musculus MNS1*), was a little bit problematic and required increasing the default recursion limit in python.