SNU 2024 Fall Algorithm Course Homework Assignment 1

Introduction

In this assignment, you will write a program in Python or Java that counts the occurrences of k-mers in the whole DNA sequence of the bacteria E. coli (~5Mb). We will provide two example files, one from a harmless strain and one from a harmful strain. The files are in a .fasta format, meaning there are annotation lines starting with ">", followed by sequence with line-break every 80 characters. There may be multiple ">" annotation lines in the file.

Example fasta:

>NC_017214.2 Bifidobacterium animalis subsp. lactis BB-12, complete sequence

ATGAACGCAGAGAATCTCGACCCCGCCACGCAGGCGCAGACCATTTGGTCCGACACGCTTGCGTTGATCAAGCAGAACTC

CAGACTCACCGCGCGCGAACAGGGCTGGCTCGCCGGGGTCACTGCCGAGGCGGTGGTCGGCACCACGATCATTCTCGATG

The k-mer length will be provided as an argument, and the result should be sorted alphabetically. We will provide a paper, *Kim, Sun, and Yanggon Kim. "A fast multiple string pattern matching algorithm." Proceedings of 17th AoM/IAoM Conference on Computer Science. 1999*, which presents an encoding scheme for strings. You are encouraged to implement this scheme for faster performance, as discussed in class. The encoding used is as follows:

ENCODE(A) = 00

ENCODE(C) = 01

ENCODE(G) = 10

ENCODE(T) = 11

Example: P=AC is encoded as 00 01 and T=ATAC as 00 11 00 01.

Please refer to the Example 4 of the provided paper for detailed implementation.

Additionally, you MUST provide a ***run.sh*** file to execute your code. The program should take input via a file and output the k-mers and their counts in the specified format. Your algorithm will also be graded on performance time, and we will compare Python and Java submissions separately to account for framework limitations. We will open the server accounts for test runs, so make sure your codes work without any errors on the server environment. Carefully read the **Example run command** and **Example output** format below.

Task Instructions

1. Write a Python or Java program that counts the number of k-mers in the DNA sequence of E. coli. The k-mer length, k, will be provided as a command-line argument. Do NOT install or import extra modules or packages; only default libraries are allowed to use.

2. DNA is composed of four characters: A, T, G, C. Any occurrences of the character 'N' (e.g., in ATNC) should be removed (so ATNC becomes ATC).

3. Input files may contain multiple chromosomes, and the count should be performed for each chromosome separately, then summed at the end.

4. Your program should read the input DNA sequence from a file and generate an output file containing the k-mers and their counts, sorted alphabetically.

5. Write a shell script (run.sh) to run your code. The shell script should take two arguments: the k-mer length and the input file name.

6 . The output file must strictly follow the format provided below, as grading will be automated based on this format.

7. All students will be given an id to a linux server to perform and test their codes. The environment will also be used for testing and grading, so **make sure your code performs as expected on the server** (Server open from Oct 20 or sooner). The ID for each student is provided in the attached excel file.

Domain: snubi1.snu.ac.kr

Example Run Command

sh run.sh [k-mer] [input_file]

Example run: sh run.sh 4 ecoli.fasta

Expected Output Format

The output filename must be in the following format: [your_student_id].txt. The contents of the output file should follow this structure (example k = 2):

20201234.txt:

AA,4

AC,1

AG,3

AT,6

…

The first column is the alphabetically sorted k-mer (in this case k=2) and the second column is each k-mer count from the input file.

Grading Criteria

We will test submitted codes with 10 E.coli files *(not provided)* with different length of k-mers and the correctness of k-mer counting and the sorting results will be checked. If all answer are correct, we will measure the run time of the code, and the grades will be given accordingly.

Submission Guidelines

1. Submit your Python/Java code, the input file, and your "run.sh" script in a single zip file in ETL.

- ✓ Ensure that the output strictly follows the example format, as grading will be done automatically based on this format.
- ✓ The output file should be named as your student ID followed by .txt (e.g., 202012345.txt).

2. Late submissions will not be accepted unless you have prior approval.

Due date: Oct 31st 23:59:59