Scientific Computation
Spring 2021
Project 1 (version 1.0)

Due: Friday February 12th 12:00pm GMT (noon)

There are three main files for this assignment: 1) the one that you are reading which is the project description, 2) *project1.py*, a Python module which you will complete and submit on Blackboard (see below for details) and 3) *project1.tex*, a template file for your short report which will also be submitted on Blackboard. The discussion and figure(s) described below should be placed in this report. An example "long" gene sequence (*Sexample.txt*) has also been provided.

1. In this question, you will work with the functions *func1A* and *func1B* provided in *project1.py*. See the function documentation for a description of its input.

   (a) (4 points) Provide a brief, clear description of the functionality and correctness of *func1A* and *func1B* (analysis of *Merge* is not required, you may state results from lecture or elsewhere). Include a clear explanation of each function's output and the strategy the functions use to produce the output.

   (b) (6 points) Your analysis of the efficiency should include: i) a clear and concise discussion of the running time and how it depends on the input, ii) a critical comparison of *func1A* and *func1B*, iii) 1-3 figures illustrating key trends in the walltimes required by the functions, and iv) a description of and explanation of the trends shown in the figure(s). The code that generates your figures should be placed in test_funcA. Place your discussion and figure(s) in the appropriate section of your report. The __name__=='__main__' portion of the module should call test_funcA and generate any figures included in your submission.

2. You will now develop codes to analyze a length-N gene sequence S provided as input to the function, *gene1*. Bases are represented by the characters A,G,C, and T. A list of P smaller sequences, *L_in* is also provided as input along with an integer, x, with $x < 3$. You are tasked with efficiently finding all locations within S of 'point-x' mutations of the provided sequences in *L_in*. A 'point-x' mutation of a length-M sequence, s, is another length-M sequence which is identical to s aside from the base in its xth position (with x=0 corresponding to the first base in the pattern). So, GTAC is a point-0 mutation of ATAC (and vice-versa).

   (a) (7 pts) Complete the function *gene1*, so that it efficiently finds the locations of all point-x mutations within S of each pattern provided in L_{in}. The locations in S at which each mutation begins should be stored in a length-P list, *Lout* where the ith element of *Lout* is a list contains the locations of the point-x

mutations of the ith input pattern arrange in ascending order. If x is negative, the locations of exact matches should be collected and returned. You should design your code for input with $N \gg M$, $M \gg 1$, and $P \gg 1$, though it may be helpful to consider smaller problem sizes when developing and testing your code. Your code should be efficient with regards to running time and memory usage, and you should assume that the cost of Python integer arithmetic is independent of the length of the integer. See the function documentation for further details on the function output.

(b) (3 pts) Add a clear description of your code along with an analysis of its running time to your report.

Further guidance

- You should submit both your completed python file and a pdf containing your discussion and figure(s). You are not required to use the provided latex template, any well-organized pdf is fine. To submit your assignment, go to the module Blackboard page and click on"Project 1". There will be an option to attach your files to your submission. (these should be named $project1.py$ and $project1.pdf$). After attaching the notebook, submit your assignment, and include the message, "This is my own work unless indicated otherwise." to confirm the work as your own.

- Please do not modify the input/output of the provided functions without permission, and please do not import any modules without permission. You may create additional functions as needed, and you may use any code that I have provided during the term.

- Marking will be based on the correctness of your work and the degree to which your submission reflects a good understanding of the material covered up to the release of this assignment. Excluding figures, you should aim to keep the pdf version of your report to less than 1 page.

- Open-ended questions require sensible time-management on your part. Do not spend so much time on this assignment that it interferes substantially with your other modules. If you are concerned that your approach to the assignment may require an excessive amount of time, please get in touch with the instructor.

- Questions on the assignment should be asked in private settings. This can be a "private" question on Piazza (which is distinct from "anonymous"), using the "Chat" on Teams during a Q&A session, or by arrangement with the instructor.

- Please regularly backup your work. For example, you could keep an updated copy of your files on OneDrive.

- In order to assign partial credit, we need to understand what your code is doing, so please add comments to the code to help us.

- You have been asked to submit code in Python functions, but it may be helpful to initially develop code outside of functions so that you can easily check the values of variables in a python terminal.