

# Python Unit 4 programming problem

October 6, 2016

**These are the exercises for this week. YOU ONLY HAVE TO UPLOAD A TEXT FILE CONTAINING THE SCRIPT TO SOLVE PROBLEM 2, problems 1 is optional and won't be marked.**

**Exercise 1.** Write a function that counts all the occurrences (including the overlapping ones!) of a given pattern within a string. For example, it should return 3 for the number of occurrences of the pattern “AA” in the string “ATGCAAAAGCTGA”. Do not use a loop structures (as we did for the Unit2 problem), instead make a function that calls *itself* until all the occurrences are found. This is called *recursion* and it is widely used in computing.

sep

**Exercise 2.** Modify the script you wrote for Unit3 so that it:

1. Reads a FASTA file and extracts the DNA sequence from it. The program must accept the file name as an argument (do not hard-code the input filename). Thus, once you have written the script you should be able to analyze the “Human\_HRAS.fasta” file we used in the previous assignment as well as any other file you provide without changing the script. To help you test it, I am sending you two more FASTA files: “hg19\_chr14\_34419875\_34420045.fasta” and “hg19\_chrM\_12900\_14157.fasta”.
2. Reads the standard and the mitochondrial genetic codes from files “GeneticCode\_standard.csv” and “GeneticCode\_mito.csv” respectively (both are text files with the same tab-delimited format). Since you always have to read these files, regardless the input sequence it is ok to hard-code these filenames.
3. Translates the DNA imported in step #1 sequence in all six coding frames according to both genetic codes in step 2.
4. Saves all translations to a single text file whose name was also given as an argument (do not hard-code the output filename).

**IMPORTANT:** starting with this problem we will take into account the modularity of your program. Try to use functions for specific tasks, particularly those that are repeated along the execution of the program.

**Happy coding!**