Bioinformatics analysis with Python 1, BI121G, spring 2025

Assignment 2

Instructions

This assignment consists of 3 tasks related to the lecture material.

For task 1 you will get 0-3 points, as follows:

- 0 points if the answer is missing or if it is completely wrong.
- 2 point if the answer is a reasonable attempt that basically (or almost) solves the task, but with some errors or using badly chosen solutions.
- 3 points if the answer solves the task with a well chosen solution and contains no errors.

For task 2-3 you will get 0-7 points, as follows:

- 0 points if an answer is missing, is completely wrong or generates syntax errors.
- 3 - 4 points if the answer contains major errors, but is an attempt to solve the task. The program should be able to run without generating any syntax errors. The code is not described or badly described.
- 5-6 points if the answer is a reasonable attempt that basically (or almost) solves the task, but with some minor errors or using badly chosen programming constructs. The code is at least sparsely described and does not contain any major incorrect descriptions of the code.
- 7 points if the answer solves the task and contains no errors. The program is well structured, for example, have explanatory variable and function names, and in detail and correctly described.

The maximum score for the assignment is 17 and you get a grade according to this scale:

F: 0-8, E: 9, D: 10-11, C: 12-13, B: 14-15, A: 16-17

Requirements for all programming tasks

- When the task requires you to implement a Python program, you have to submit your solution as a Python script, i.e., in a file with file extension ".py". A separate Python program should be submitted for each task. Program solutions to tasks that have been written in the same file will not be corrected.
- For each task you have to provide a description of your solution. For example, you can use comments in your code which explains the most important statements in your code. From your description it should be clear what the code does.
- You are only allowed to use Python built-in modules (e.g., re and os) to solve the problem. No external modules (e.g., Pandas, NumPy or biopython) may be used.

The assignment is solely individual work since it is part of the examination for the course. It is not allowed to cooperate with other students or copy code from other sources when you answer the questions. Any undue actions will be reported to the Disciplinary Committee.

In case you fail the assignment (grade F), you will have one opportunity for re-examination. In that case you will get a new set of questions. If you fail a second time, then you will need to wait until the next time the course is given to get a new chance.

Your files must be uploaded in the assignments section of the course site. You can make a compressed archive of the Python program files and upload the zip-file as your submission, or you can upload each Python program separately. Python programs that are not in .py-format will not be corrected. The rough outline can be submitted in Word- or pdf-format.

If anything is unclear, then send me an email at angelica.lindlof@his.se.

1. Draw a rough outline for the following problem:

   Below you see a table with measured RNA expression levels for some genes. From this table, calculate the following descriptive statistics:
   - Average expression level for each row.
   - Average expression level for each column.

Gene ID	1 hour	2 hours	3 hours	4 hours
NM_4385849	56	33	21	20
NM_8694838	44	47	43	44
NM_4680808	12	53	76	89

2. Implement a Python program that asks the user for a DNA sequence. Thereafter, change all nucleotides to uppercase letters and exchange all T (thymine) and A (adenine) to a * (star) character. It should also count the number of G's respectively C's in the sequence. Finally, print out the new sequence to the shell as well as the derived numbers.

   For example,

```
Input DNA sequence: attggccaca
Output: ***GGCC*C*
Number of G's: 2
Number of C's: 3
```

3. Implement a Python program that reads in a file containing genes and annotation about their exons' start and end positions. The exons's start and end position is given in the second column in the file and is in the format start:stop, where start refers to first position of exon on the chromosome and end the last position on the chromosome. See example file in Canvas.

   Your program should store the information in a dictionary of lists, where the key is the name of each gene in the file and value is a list of all exons's start and end positions for each gene. The first row, containing 'Gene Exon', should not be included in the dictionary.

   For example, a file containing the following information:

```
Gene                    Exon
HPR11                   3:67
HPR11                   77:89
HPR11                   114:127
HPR12                   1:55
HPR12                   59:112
HPR12                   156:272
HPR12                   357:487
QRC34A                  4:89
QRC34A                  105:548
ATB8                    93:1105
ATB1                    102:885
```

   Would generate a dictionary with the following lists:

```
Key                     Value
HPR11                   [3:67, 77:89, 114:127]
HPR12                   [1:55, 59:112, 156:272, 357:487]
QRC34A                  [4:89, 105:548]
ATB8                    [93:1105]
ATB1                    [102:885]
```

   Thereafter, implement a function that takes as input a gene name given by the user, searches for this in the dictionary and, if found, prints out the information about exon start and end positions. The output should not merely be the list of items, but should be presented with two tab-separated items – first for start position and second for end position. Like this,

```
HPR11 exons:
Start           Stop
3               67
77              89
114             127
```

   You should test your program on the example file that can be downloaded from the assignment page. Your program should be generic, i.e., it should work on a file with any number of genes, not only the given as the example to this assignment.