

HOMEWORK 2 – 07/12/2023

Submission date: Friday, Dec 22th, 13:30

Use Spider to do your homework, save solution of each question as a separate *.py file such as:

groupNoXXX_HW2_Q1.py

1. (30 points) **Ribosome Function**

- Make a dictionary of RNA codons and aminoacids (use the table in the slides).
You must have done this step last week in the lab, you can use the same code. If not, use codon-aa.txt file, read the text into Python and make your dictionary using this information.
- Read the sequence in seq.txt as a string into Python (remove non-nucleotide characters such as numbers, spaces, line breaks!)
- Write a function that takes an RNA sequence and a dictionary as optional arguments, the default values will be the dictionary and the sequence you prepared in parts a and b.
- The function should find one gene (start codon -> stop codon) and find the aminoacid sequence of the gene region using the RNA sequence and the dictionary.
- Test it with default values and write the aminoacid sequence in a file.

2. (30 points) **Missense Mutation Function**

- Write a function that takes three arguments: an RNA sequence, a codon-aminoacid dictionary and a point mutation on RNA
- The function should call the function you wrote in Q1c to convert the RNA sequence to aminoacid sequence (conversion can run either on whole sequence or a part of it)
- Then function should return the index of the aminoacid effected by the mutation and if the mutation changes the aminoacid that is coded by the codon (missense mutation or not)

Note: Mutation information should be provided as follows:

r.123A>G (on RNA, A in position 123 is replaced by G)

p.Pro252Arg (on protein, proline at position 252 is replaced by arginine)

3. (40 points) **Data Extraction and Standardization**

The data file 'gene_expression.txt' gives a table of gene expression values of genes in 5 replicates from an RNA-seq experiment. Each line of the file gives expression values for one gene.

a/ Write a Python script that reads the file, standardizes the expression values and saves them in a new file.

- Write a function for standardization, it should take a list of numbers and return their standardized values as a tuple or list.
- Call the function for each gene in your script, save the standardized values as a table in your output file.

(Standardization means converting an X variable distributed with normal distribution into z-value, which is distributed by standard normal distribution. The formula is as follows:

$$z[i] = (x[i] - \text{meanX}) / \text{stdX}$$

You can calculate mean and std of a gene by using:

```
import numpy as np
```

```
x=[x0, x1, x2 ,x3 ,x4, x5] # x0 ... x5 are expression values of a gene
```

```
meanX = np.mean(x)
```

```
stdX = np.std(x)
```

Have fun!