Class 2: Python Basics Exercise

Question 1

Suppose we want to transform a protein represented in 3 letters for each amino acid to a representation of 1 letter for each aa. Following is a dictionary in Python to assist in this:

code321 = {"GLY" : "G", "ALA" : "A", "LEU" : "L", "ILE" : "I",

"ARG" : "R", "LYS" : "K", "MET" : "M", "CYS" : "C",

"TYR" : "Y", "THR" : "T", "PRO" : "P", "SER" : "S",

"TRP" : "W", "ASP" : "D", "GLU" : "E", "ASN" : "N",

"GLN" : "Q", "PHE" : "F", "HIS" : "H", "AAA": "A"}

a. This dictionary has 2 issues:
   1. The value for Valine ("VAL") is missing. Add it to the dictionary.
   2. It has a redundant key – "AAA". Delete it from the dictionary.

   Do both with Python commands, not manually.

b. Write a function prot321(prot_seq):
   - The function gets a protein sequence prot_seq, in 3 letter representation (separated by spaces). For example: "`GLN ALA GLN ILE`".
   - The function returns the protein in 1 letter representation (no spaces). For example: '`QAQI`'.

   Help: Use the function split from class str, to separate the input protein sequence into the triplets. For example:

```
>>> "GLN ALA GLN ILE".split()
['GLN', 'ALA', 'GLN', 'ILE']
```

   Then you can simply iterate over the list of triplets and make the conversion using the dictionary above.

Question 2

The goal of this exercise is to practice basic computations in Python with biological sequences. In the rest of the course we will be doing this quite a lot, and it is important to "fill comfortable" with this type of work.

We will be doing the analysis without the help of existing tools, although for the purposes of this exercise there are probably several good ready to use tools available. But the goal here is to learn how these computations are done "behind the scenes", and to develop independence in working with strings. This will allow you in the future to solve specific problems for which no tools are available, and to tailor existing tools for your needs.

Stage 1 – reading a gene

In the site you will find a file IME1.txt, containing the DNA sequence of the gene IME1 (of S.Cerevisiae, sequence downloaded from www.yeastgenome.org). We want to read this sequence from the file, translate it into a protein, and conduct several simple analyses on it.

The sequence appear in FASTA format. In this format, before the gene itself, there is a title row containing basic information on the gene. This row usually starts with the > character.

a. Read the DNA sequence without the title row, into a variable ime1_dna.
b. The length of the gene should be 1083 nucleotides. Check if this is the case (hint: no!).
c. Check if the sequence contains only the nucleotides A, T, C, G. Use the function check_dna we saw in class, and the function lower of the str class.
d. If you got a false answer, what other characters are in that string? Answer by writing a simple loop, not by mere manual inspection.
e. Another way to answer the previous section is by using a structure called **list comprehension.** This is just another convenient way to form lists in Python. Run this command:

```
[char for char in ime1_dna if letter not in "ATCG"]
```

The general syntax is:

```
[expression for variable_name in sequence if condition]
```

f. Clean the sequence from non ATCG characters. Use the function replace of class str.
g. Check again the length of the gene. It should now be 1083.
h. Check if the length of the gene is divided by 3.
i. What are the last 3 nucleotides in the sequence? Use string slicing as we saw in class.
j. Verify, using the dictionary 'standard' that we saw in class, that the last codon in a stop codon.

Stage 2 – translation and analysis

a. Try using the function dna_translate that we saw in class to translate the gene into a protein. What's the problem?
b. Solve that problem, and put the translated protein sequence into a variable called ime1_prot. Verify that the length is what you expect.
c. What is the last character in ime1_prot? Find a way to remove this character.
d. Following is the sequence of the protein of IME1 (known as Ime1):

```
MQADMHGKLHAALEDGFFLFPFEQQQQPNIYYDTTTDQEDRPCFSFGSTISPRSWHFEKS
DKIASSQLQNLVHTQPIHLINPQILFNEEFLNLENIDSQPISKETKTTKDCTMATGPERG
KKSSESTRSSSLSSLFSNDESASTFHSSFNNHDNFQKSNRNGDDIDISDTIKYETNTNAQ
KDIKIFQENFEFNEFPYTQDFYPYTTNYTYSKPTNIHESINSKNTDSYSQYQDQFPPHTD
NIHSFNNRHYSNHKSTNCNYYNNTSNNNNASDNVYEADPFIDEPQVPSYYYPLEIAFDVE
KSPPPSLQKLNSKELEFLKKLNSKLSRYAAAYSFSSSNDQDYYDKVRFQEISYKFSKTYS
```

Verify that this one is identical to what you got in section c.

Help: The sequence was copy-pasted from an internet page, and thus contains several characters which are not really part of the gene.

e. What is the percentage of basic amino acids (Lysine – K, Histidine – H, Arginine – R) in the protein? Use a simple for loop to answer that:

```
basic_aa_list = ["K", "H", "R"]
for aa in basic_aa_list:
    …
```

Stage 3 – mutations

a. **Elongation** mutation: save in a variable ime1_mut1 the IME1 gene with an addition of ATT at the end.
b. **Point** mutation: save in a variable ime1_mut2 the IME1 gene where the fifth nucleotide changed from A to T. The gene should start `ATGCT`. Do this by converting the string to a list and then back again.
c. **Insertion** mutation: save in a variable ime1_mut3 the IME1 gene where the nucleotide G was introduced before the fifth place. The gene should start `ATGCGA`. This time practice string slicing.
d. **Deletion** mutation: save in a variable ime1_mut4 the IME1 gene where the fifth nucleotide was deleted