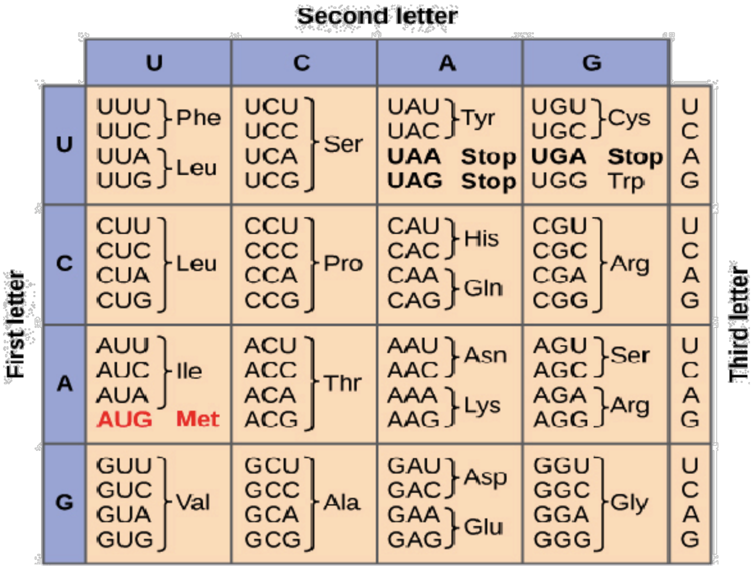
22BIO201 Intelligence of Biological Systems 1

Lab Sheet 2

1. Create a Python dictionary to store the RNA codon table explained in the class(Use the one letter representation of the amino acid). Download the DNA sequence of ‘Insulin’ from NCBI and do the process of transcription and translation to see which amino acid sequence is produced from it. Compare your result against the amino acid sequence of Insulin downloaded from NCBI.



		Second letter			
	U	C	A	G	
First letter U	UUU UUC } Phe UUA UUG } Leu	UCU UCC UCA } Ser UCG	UAU UAC } Tyr **UAA Stop** **UAG Stop**	UGU UGC } Cys **UGA Stop** UGG Trp	U C A G
C	CUU CUC CUA } Leu CUG	CCU CCC CCA } Pro CCG	CAU CAC } His CAA CAG } Gln	CGU CGC CGA } Arg CGG	U C A G
A	AUU AUC } Ile AUA AUG Met	ACU ACC ACA } Thr ACG	AAU AAC } Asn AAA AAG } Lys	AGU AGC } Ser AGA AGG } Arg	U C A G
G	GUU GUC GUA } Val GUG	GCU GCC GCA } Ala GCG	GAU GAC } Asp GAA GAG } Glu	GGU GGC GGA } Gly GGG	U C A G Third letter

2. Create a .fasta file with the following content

>O00626|HUMAN Small inducible cytokine A22.

MARLQTALLVVLVLLAVALQATEAGPYGANMEDSVCCRDYVRYRLPLRVVKHFYWTS DS<=

CPRPGVVLLTFRDKEICADPR

VPWVKMILNKLSQ

a. Read the file, extract the header information and print it.

b. Read and print the sequence from the file.

c. Append molecular weight of the sequence at the end of the file.

3. Compute the Number of Times a Pattern Appears in a Text

Description : This is the first problem in a collection of "code challenges" to accompany *Bioinformatics Algorithms: An Active-Learning Approach* by Phillip Compeau & Pavel Pevzner.

A k-mer is a string of length *k*. We define *Count*(*Text*, *Pattern*) as the number of times that a *k*-mer *Pattern* appears as a substring of *Text*.

For example,

Count(ACAACTATGCATACTATCGGGAACTATCCT,ACTAT)=3.

We note that *Count*(CGATATATCCATAG, ATA) is equal to 3 (not 2) since we should account for overlapping occurrences of *Pattern* in *Text*.

Implement PatternCount

Given: {DNA strings}} *Text* and *Pattern*.

Return: *Count*(*Text*, *Pattern*).

Pseudocode:

PatternCount(*Text*, *Pattern*)

```
count ← 0
for i ← 0 to |Text| - |Pattern|
    if Text(i, |Pattern|) = Pattern
        count ← count + 1
return count
```

Sample Dataset

```
GCGCG
GCG
```

Sample Output

2

Real Dataset

Input:

Text : Vibrio Cholerae Oric DataSet

Pattern: ATGATCAAG

Output:

3

Optional : Visit http://rosalind.info/problems/ba1a/ . Solve the problem. Use the sample dataset given in the site.

4. Find All Occurrences of a Pattern in a DNA String

Description: In this problem, we ask a simple question: how many times can one string occur as a substring of another? Recall from “Find the Most Frequent Words in a String” that different occurrences of a substring can overlap with each other. For example, ATA occurs three times in CGATATATCCATAG.Pattern Matching Problem

Find all occurrences of a pattern in a string.

Given: Strings *Pattern* and *Genome*.

Return: All starting positions in *Genome* where *Pattern* appears as a substring. Use 0-based indexing.

Sample Dataset

```
ATAT
GATATATGCATATACTT
```

Sample Output

```
1 3 9
```

Real Dataset

Vibrio Cholerae Genome DataSet

Pattern: ATGATCAAG

Output:

116556 149355 151913 152013 152394 186189 194276 200076 224527
307692 479770 610980 653338 679985 768828 878903 985368

Visit http://rosalind.info/problems/ba1d/ . Solve the problem. Use the sample dataset given in the site.

5. Find the Most Frequent Words in a String

Description: We say that *Pattern* is a **most frequent** ***k*-mer** in *Text* if it maximizes *Count*(*Text*, *Pattern*) among all k-mers. For example, "ACTAT" is a most frequent 5-mer in "ACAACTATGCATCACTATCGGGAACTATCCT", and "ATA" is a most frequent 3-mer of "CGATATATCCATAG".

Frequent Words Problem

Find the most frequent k-mers in a string.

Given: A DNA string *Text* and an integer *k*.

Return: All most frequent *k*-mers in *Text* (in any order).

Sample Dataset

```
ACGTTGCATGTCGCATGATGCATGAGAGCT
4
```

Sample Output

```
CATG GCAT
```

Real Dataset

Vibrio Cholerae Oric DataSet

K= 9

Output:

```
atgatcaag cttgatcat tcttgatca ctcttgatc
```

Optional: Visit http://rosalind.info/problems/ba1b/ . Solve the problem. Use the sample dataset given in the site.