

- ▼ Kaushik Raman

HW1

▼ 1.Sequence Analysis

```
sequence = ("ATGTACAACATAGTGATGGAGACGGAGCCATTGAAGCCGCCGGGCCCCGCAGCAAACCTTCGGGGGGCGGCGG(
```

A. Calculate number of nucleotides in the given sequence.

```
length = len(sequence) #finds the length of object
length
```

99

B. Calculate number of codon in the given sequence (if it is all translated)?

```
codons = len(sequence)/3
codons
```

33.0

C. Check whether the given sequence starts with 'ATG'.

```
#finds and checks if the object sequence starts with "ATG"
if(sequence.startswith("ATG",0,3)):
    print("The sequence starts with ATG")
else:
    print("The sequence doesn't start with ATG")
```

The sequence starts with ATG

D. Check whether the sequence ends in frame. (the length of the sequence is divisible by 3)

```
#checks whether the remainder of the seq length is 0 to check divisibility by 3
if (len(sequence)%3==0):
    print("The sequence ends in frame.")
```

```
else:
    print("The sequence doesn't end in frame.")
```

The sequence ends in frame.

E. What percentage of sequence is GC ?

```
#counts the G,C in the seq and divides them by the total length
GC_content = (sequence.count('G') + sequence.count('C'))/length*100
GC_content

63.63636363636363
```

F.Print only the first and last codon.

```
first_codon = sequence[0:3]
last_codon = sequence[-3:]
```

```
first_codon
```

```
'ATG'
```

```
last_codon
```

```
'UAG'
```

G. Using the dictionary provided below, what amino acid does the second codon represent?

```
geneticcode = {
'ATA': 'I', 'ATC': 'I', 'ATT': 'I', 'ATG': 'M',
'ACA': 'T', 'ACC': 'T', 'ACG': 'T', 'ACT': 'T',
'AAC': 'N', 'AAT': 'N', 'AAA': 'K', 'AAG': 'K',
'AGC': 'S', 'AGT': 'S', 'AGA': 'R', 'AGG': 'R',
'CTA': 'L', 'CTC': 'L', 'CTG': 'L', 'CTT': 'L',
'CCA': 'P', 'CCC': 'P', 'CCG': 'P', 'CCT': 'P',
'CAC': 'H', 'CAT': 'H', 'CAA': 'Q', 'CAG': 'Q',
'CGA': 'R', 'CGC': 'R', 'CGG': 'R', 'CGT': 'R',
'GTA': 'V', 'GTC': 'V', 'GTG': 'V', 'GTT': 'V',
'GCA': 'A', 'GCC': 'A', 'GCG': 'A', 'GCT': 'A',
'GAC': 'D', 'GAT': 'D', 'GAA': 'E', 'GAG': 'E',
'GGA': 'G', 'GGC': 'G', 'GGG': 'G', 'GGT': 'G',
'TCA': 'S', 'TCC': 'S', 'TCG': 'S', 'TCT': 'S',
'TTC': 'F', 'TTT': 'F', 'TTA': 'L', 'TTG': 'L',
'TAC': 'Y', 'TAT': 'Y', 'TAA': '_', 'TAG': '_',
'TGC': 'C', 'TGT': 'C', 'TGA': '_', 'TGG': 'W' }
```

```
second_codon = sequence[3:6]
geneticcode[second_codon]

'Y'
```

The second codon represents the amino acid Y

▼ 2. Fahrenheit to Celsius conversion

A. Convert the temperature 23 degrees in Fahrenheit to Celsius.

```
#function that converts F to C
def f2c(f):
    assert isinstance(f,int), "input must be an integer"
    print("The temperature in Farenheit is", f)
    c = ((f-32)*5)/9
    print("The temperature in Celsius is", c)

    return(c)

f2c(23)

    The temperature in Farenheit is 23
    The temperature in Celsius is -5.0
    -5.0
```

▼ 3. Gene Sets

Below are two lists of genes in Rice. One is a list of genes that responds to Nitrogen and the other is Carbon.

Nitrogen = Os1, Os2, Os3, Os5

Carbon = Os1, Os3, Os4, Os5

A. Create 2 sets of genes, one for Nitrogen and one for Carbon.

```
Nitrogen = set(["Os1", "Os2", "Os3", "Os5"])
Carbon = set(["Os1", "Os3", "Os4", "Os5"])
```

B. Using python functions, determing which genes are present in both sets.

Nitrogen & Carbon #finding the interesection between sets

```
{'Os1', 'Os3', 'Os5'}
```

C. Using python functions, determing which gene(s) are present only in Nitrogen set.

Nitrogen - Carbon #finding the difference between sets

```
{'Os2'}
```

D. Create a new set called Allgenes that contains genes from both sets.

Allgenes = Nitrogen | Carbon #creating a new set that has elements from both sets
Allgenes

```
{'Os1', 'Os2', 'Os3', 'Os4', 'Os5'}
```

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 7:09 PM

