

## Day 2: Data structures

Let's start by importing the **data** module

```
In [13]: from data import DNA_Sequence, human_sequence, Species_list, sequences
```

### Problem 1

The `*DNA_Sequence*` variable imported from the **data** module contains DNA sequence

a) Print the **DNA\_Sequence** variable

```
In [4]: print DNA_Sequence  
GTAGCTGATGCTAGCTGTGATTATTCGTACGATTTGTCGTAGTGTCGTATGCGTAGCTGATGCGTAT
```

b) Print the length of the **DNA\_Sequence** variable

```
In [5]: len(DNA_Sequence)
```

```
Out[5]: 67
```

c) Count (with a method) the number of "A", "C", "T", and "G" nucleotides in the **DNA\_Sequence** variable and assign each one to a different variable (e.g, assign the number of "A"s to a variable named `a_count`, etc) and then print each variable

```
In [10]: a_count = DNA_Sequence.count('A')  
b_count = DNA_Sequence.count('C')  
c_count = DNA_Sequence.count('T')  
d_count = DNA_Sequence.count('G')  
print a_count  
print b_count  
print c_count  
print d_count
```

```
12  
10  
25  
20
```

d) Transcribe the **DNA\_Sequence** and assign it to a new variable. Print the new variable.

(Tip: replace the "T" nucleotides for "U" nucleotides)

```
In [15]: RNA_sequence = DNA_Sequence.replace("T", "U")  
print RNA_sequence
```

```
GUAGCUGAUGCUAGCUGUGAUUUAUUCGUACGAUUUGUCGUAGUGUCGUAUGCGUAGCUGAUGCGUAU
```

e) Split the **DNA\_Sequence** at each "GAT" motif and store the resulting list in a new variable. Determine the

number of the resulting fragments. Print the result.

(Tip: Determine the length of the resulting list of fragments to get the number of fragments)

```
In [21]: mylist = DNA_Sequence.split("GAT")
        print len(mylist)
```

5

f) Merge the first and last fragments of the list resulted from e) and store it in a new variable. Print the new variable.

```
In [25]: new_seq = mylist[0] + mylist[-1]
        print new_seq
```

GTAGCTGCGTAT

## Problem 2

The **human\_sequence** variable imported from the **data** module contains a human DNA sequence

a) Print the **human\_sequence** variable

```
In [26]: print human_sequence
```

-----CCCACGCGTCCGCGGACGCGTGGGCGTACGCGTGGGCGGACGCGTGGGAAGAAATCT'

b) Notice that both ends of the sequence contain gaps "-". Eliminate the gaps from boths ends of the sequence, and assign the resulting sequence to a new variable. Print the result.

```
In [28]: clean_seq = human_sequence.strip("-")
        print clean_seq
```

CCCACGCGTCCGCGGACGCGTGGGCGTACGCGTGGGCGGACGCGTGGGAAGAAATCTTAGACAAAAAAGT'

c) Change the capitalization of the **human\_sequence** variable and print

```
In [29]: print human_sequence.lower()
```

-----cccacgcgtccgcggacgcgtgggcgtagcgcgtgggcggaagaaatct'

## Problem 3

The **Species\_list** variable imported from the **data** module contains a list with species names.

a) Determine the number of species in **Species\_list** and print it.

```
In [30]: print len(Species_list)
```

7

```
In [41]: Species_list.sort()
print Species_list

['B_bufo', 'B_taurus', 'C_albicans', 'C_felix', 'H_sapiens', 'M_musculi']
```

```
In [42]: Species_list[2] = "D_melanogaster"
print Species_list

['B_bufo', 'B_taurus', 'D_melanogaster', 'C_felix', 'H_sapiens', 'M_mu']
```

```
In [44]: First_species = Species_list[:3]
print First_species

['B_bufo', 'B_taurus', 'D_melanogaster']
```

```
In [46]: new_species = []
new_species.append("C_kahawae")
new_species[1:3]= ["Q_suber", "L_lepida"]
print new_species

['C_kahawae', 'Q_suber', 'L_lepida']
```

[illegible]

3 of 4

```
In [49]: a = 23
b = 323
print a + b
print a - b
print a / b
print a * b
```

```
346
-300
0
7429
```

c) Notice that the division of 23 by 323 results in "0". Convert both numbers into floating point variables and repeat the division

```
In [50]: a = 23.
b = float(b)
print a / b
```

```
0.0712074303406
```

## Problem 5

The **sequences** variable contains a dictionary with taxon name as keys, and their DNA sequence of the Cytb gene as values.

a) Determine the number of taxa contained in the dictionary

```
In [53]: print len(sequences)
```

```
37
```

b) Print both the taxon name and sequence of the 3<sup>o</sup>, 5<sup>o</sup> and 7<sup>o</sup> dictionary item.

```
In [61]: third = sequences.keys()[2]
fifth = sequences.keys()[4]
seventh = sequences.keys()[6]
print third + " " + sequences[third]
print fifth + " " + sequences[fifth]
print seventh + " " + sequences[seventh]
```

```
Mo10 GGA CTGTGCCTAATTACTCAAATTGTTACAGGGTTATTTT TAGCAATACACTACAATGCAGATAT
Ib9 GGATTGTGCCTAATTACTCAAATTGTTACAGGATTATTTT TAGCAATACACTACAATGCAGATATT
Ib17 GGATTGTGCCTAATTACTCAAATTGTTACAGGATTATTTT TAGCAATACACTACAATGCAGATAT
```

```
In [ ]:
```