Programming for Data Science: Lab02 Functions

Instructor: Jakramate Bootkrajang
Partly based on material by Andrew Dalke

Outline

- Basic DNA sequence analysis
- Advanced function call

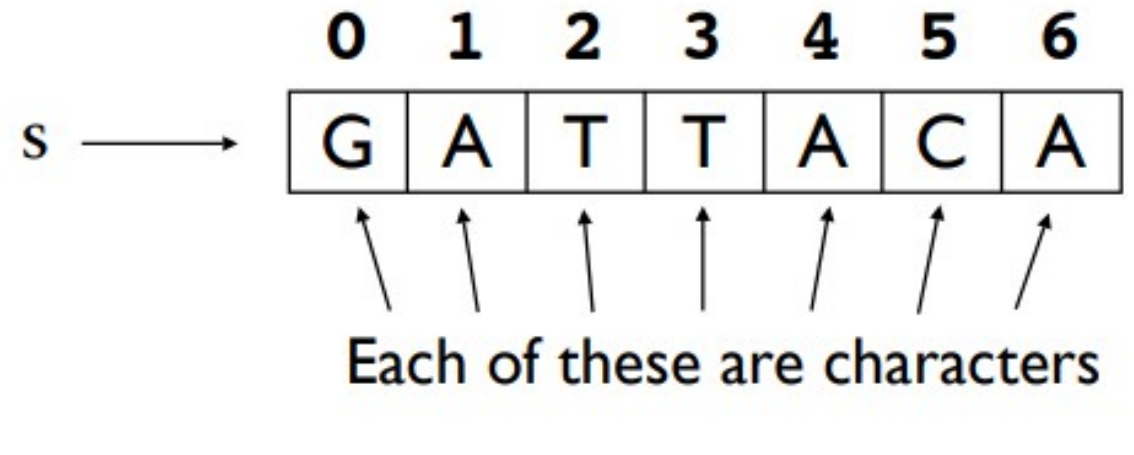
DNA Sequence

- The basic structure of DNA has four bases
  - Thymine (T)
  - Adenine (A)
  - Cytosine (C)
  - Guanine (G)
- DNA sequences are usually stored as string composed of the four bases.
  - ATCGGTAGA

Recap on string

- String is a sequence of characters

```
>>> s = "GATTACA"
```



Why are strings important ?

- DNA sequences are strings
  - ..atcgaaggaa ccacagaacc gagcgcgaag
- Database records contain strings
  - LOCUS AC005138
  - DEFINITION Homo sapiens chromosome 17, clone
- Webpage written in HTML is string

Useful string methods and functions

```
>>> len("GATTACA")
7
>>> "GAT" + "TACA"
'GATTACA'
>>> "A" * 10
'AAAAAAAAAA'
>>> "G" in "GATTACA"
True
>>> "GAT" in "GATTACA"
True
>>> "AGT" in "GATTACA"
False
>>> "GATTACA".find("ATT")
1
>>> "GATTACA".count("T")
2
>>>
```

length

concatenation

repeat

substring test

substring location

substring count

Some more methods

```
>>> "GATTACA".lower()
'gattaca'
>>> "gattaca".upper()
'GATTACA'
>>> "GATTACA".replace("G", "U")
'UATTACA'
>>> "GATTACA".replace("C", "U")
'GATTAUA'
>>> "GATTACA".replace("AT", "**")
'G**TACA'
>>> "GATTACA".startswith("G")
True
>>> "GATTACA".startswith("g")
False
>>>
```

Exercise 1

- Write a program which asks for a sequence then print its length

```
Enter a sequence: ATTAC
It is 5 bases long
```

Exercise 2

- Modify the program so it also prints the number of A,T, C, and G characters in the sequence

```
Enter a sequence: ATTAC
It is 5 bases long
adenine: 2
thymine: 2
cytosine: 1
guanine: 0
```

Exercise 3

- Modify the program to allow both lower-case and upper-case characters in the sequence

```
Enter a sequence: ATTgtc
It is 6 bases long
adenine: 1
thymine: 3
cytosine: 1
guanine: 1
```

Exercise 4

- Modify the program to print the number of unknown characters in the sequence

```
Enter a sequence: ATTU*gtc
It is 8 bases long
adenine: 1
thymine: 3
cytosine: 1
guanine: 1
unknown: 2
```

Advanced function call

- You’ve learned from previous lecture that function is one type of data object
- For this reason, it is possible to pass a function as an argument to another function

Passing function as argument

```
def g(x, y):
    return x + y

def f(z, x, y):
    return z(x,y)


f(g, 2, 3)
```

Writing a function to diff function

- Derivative of function f() at point a can be approximated by

  $$f'(x) = f(x+h) - f(x) / h$$

- We want to write a function call diff() to find the derivative of any f(x)
- How we do that ?

Solution

```
def diff(f, x, h):
    return (f(x+h)-f(x))/h


def myFunc(x):
    return x**2 + 3


print(diff(myFunc, 3, 1e-10))
```

```
6.000000496442226
```