

BIOI 3500, Advanced Bioinformatics Programming Assignment 1



Assignment 1 :: 50 points

Objective

This assignment will give you the opportunity to continue learning Python by solving straightforward bioinformatics problems.

Program (50 points)

The GC-content of a fragment of DNA is expressed using the formula:

$$\%GC = \frac{G+C}{A+T+G+C} \times 100$$

where *G*, *C*, *A* and *T* are the number of guanines (**G**), cytosines (**C**), etc. in the fragment. In general, DNA with high GC-content is more thermally stable than that with low GC-content.

For a given DNA fragment, the *complement* of the fragment is given by converting each nucleotide to its Watson-Crick complement (i.e., **A** → **T**, **T** → **A**, **G** → **C**, **C** → **G**). The *reverse complement* of the fragment is the complement of the fragment written in reverse. For example, the complement of fragment

5'-**GACCGGCTCGAGTGCTACGCGCCACCCTCTCTACTACGACTAATT**-3'

is

3'-**CTGGCCGAGCTCACGATGCGCGGTGGGAGAGATGATGCTGATTAA**-5'

and its reverse complement is

5'-**AATTAGTCGTAGTAGAGAGGGTGGCGCGTAGCACTCGAGCCGGTC**-3'

where the directionality of the DNA is indicated with the 5' and 3' labels.

Write a (procedural) Python (version 3) program that calculates and displays (two significant digits) the GC-content of an entered DNA fragment and the reverse complement. All non-base characters (i.e., characters other than **G**, **C**, **A** and **T**) should be ignored in the calculation. Name your program **netID_Assign1.py** where **netID** is your UNO NetID.

The output of your program should mimic—i.e., for the given input look exactly the same as—the following. Note that the colons are followed by two spaces. For clarity, user input is underlined; your program does not need to mimic the underlining.

```
$ python netID_Assign1.py
```

```
Enter a sequence: GACCGGCTCGAGTGCTACGCGCCACCCTCTCTACTACGACTAATT
```

```
%GC content: 57.78%
```

```
Complement: CTGGCCGAGCTCACGATGCGCGGTGGGAGAGATGATGCTGATTAA
```

```
Reverse complement: AATTAGTCGTAGTAGAGAGGGTGGCGCGTAGCACTCGAGCCGGTC
```

Note that the sequence should be echoed in ALL UPPERCASE with all non-base characters (i.e., characters other than **G**, **C**, **A** and **T**) converted to **X**.

Your program must be written in a consistent style and be appropriately commented (see the *Python Coding Style Guide*).