BIOI 3500, *Advanced Bioinformatics Programming*
Assignment 3

OBJECTIVES

This programming exercise will give you the opportunity to continue to learn Python and give you experience working with FASTA format, a common bioinformatics file format.

PROGRAM DESCRIPTION

Write a (procedural) Python (version 3) program, **netID_Assign3.py**, where **netID** is your UNO NetID, which converts a multisequence human reference sequence GenBank FASTA file into **tab-delimited** format. For a given FASTA record, output should be a single tab-delimited line that includes the GI number, the accession.version number, the gene name and symbol, and (optionally; see below) the sequence. Assuming that the sequence was to be included, the following record

```
>gi|46358415|ref|NM_005867.2|  Homo  sapiens  Down  syndrome  critical  region  gene  4
CAGTCACGGGCCAGCCTTTCCCTTAAGGATGTCTGAGAGCTGGTCTCCTCACTCTTTTCTTCCGCCTATTAAACTTTCCG
CTTCTTAACTCACCCATGTGTGTCCATGTCGTTAATCATCTTGACGAGAGATGATGAACCCCGGATATTTACCCCAGACA
GTGATGCCGCTTCACCAGCATTGCACTCTACTTCCCCGCTTCCTGATCCTGCCTCAGCTTCTCCTCTCCACAGAGAAGAA
AAAATTCTGCCTAAAGTCTGCAACATCGTTTCCTGCCTGAGTTTCAGCCTGCCAGCTTCTCCTACGGATTCTGGACTTGC
CAGCCCCACAATCATAACCAGAGAGGGGCAGCAATTTTGGGCAAAATGTCTGATTTGGAAATACCAACTTTACCTCCATG
GGCTCCACAAGAAATCAGATGGGAGAAGGGACAAGCAGATAAGCGCAAGCCCATCAACCTGAAGGCATAAACCACATCCA
GCCACCTCCTTCTGATCAGCAGCAAAGCTGACGTTTTGATCTCCATCTGTCTGATTCTTGTGTCTACTTCTCAGTTTACA
ACTCCAGTGGGAAAGAAAGAGCTTTATTTACAGACCCATAAAAATCCCATCAGTGTCGTCCCCTGCTGAGAGGCCATGTG
AGACCATATGGAAAAACAACAGCCATAATGGCAGCATGGCAGTGGAAGGGTTTGTCTTGTGCCCAGGCCTTGCGGTCATG
CAAGTTTCTTGTGGATCCTGTTGGGACCAGCCACTCACCAAGGCTGAGTAGGTCCACAAATAATGGGGACTTTCTACCAG
ACTCACAGAGAACTGCTGGGTTTTTGGGAAGGGTGTGCGTGTCTTTGGGGCATGGAAGTTGGGGTTATAGTGGAGACCCA
GAGGATGAGAAAACTTCTCTGCCTCAGCAGAAGAGTGGCAGCTGAGAGAGAGGCAAGAAACTCGCACCCACTGTGGACTG
GGGCAGAGAGATTTTGAGGAGAATGAAATCCAGAAACTCTGTGTGGTATTAGTTTGTATCCAGAGGGTGACCCTCTTCTC
AAGGAAATGGGTGTCATCAATTTTCTACACTATTAAAGATATAAAGTTCTTGGCATTAAAAA
```

would be converted to

```
46358415     NM_005867.2      Homo sapiens Down syndrome critical region gene 4   CAGT
CACGGGCCAGCCTTTCCCTTAAGGATGTCTGAGAGCTGGTCTCCTCACTCTTTTCTTCCGCCTATTAAACTTTCCGCTTCTTAACT
CACCCATGTGTGTCCATGTCGTTAATCATCTTGACGAGAGATGATGAACCCCGGATATTTACCCCAGACAGTGATGCCGCTTCACC
AGCATTGCACTCTACTTCCCCGCTTCCTGATCCTGCCTCAGCTTCTCCTCTCCACAGAGAAGAAAAAATTCTGCCTAAAGTCTGCA
ACATCGTTTCCTGCCTGAGTTTCAGCCTGCCAGCTTCTCCT. . .
```

where some of the sequence (**. . .**) has been omitted for brevity in this assignment definition. *Your final result should contain the entire sequence*.

Note that:

1. All newline characters, except the last one, have been removed from the sequence so that output is on a single line; and
2. There is not a leading space in the description (**Down syndrome**. . .)
   - In the FASTA record there is a leading space before the description—i.e., between the last **|** and the description—which must be removed.) Furthermore, the first line of the resulting output must contain the first record of the input file and records in the output file must be consecutive; i.e., they must not be separated by a blank line or lines.

3. Reading from standard input and writing to standard output must be done using the method described in the Google Colab notebook.
4. Your program must be written in a consistent style and be appropriately documented. (See the Python Coding Style Guide)
5. So that your program works with files of arbitrary size, do not read in the entire file at once (i.e., using **.readlines()**). Instead read in the file one line at a time (i.e., using **.readline()**).
6. Use the Python **getopt** module to process command-line arguments. Use the following option letters:

Options

i Specifies the name of the input file. If this option is not given, output should be read from standard input (i.e., the keyboard).

o Specifies the name of the output file. If this option is not given, output should be sent to standard output (i.e., the screen).

s Specifies whether the sequence should be included in the output: if this option is used, the sequence *should* be included; if it is absent, the sequence *should not* be included.

For example, the command

```
python3 netID_Assign3.py -i human.rna.fna -o human.rna.tab -s
```

would

- Read FASTA records from the file **human.rna.fna**
- Send output to the file **human.rna.tab**
- Include the sequence in the output

On the other hand, the command

```
python3 netID_Assign3.py -i human.rna.fna
```

would

- Read FASTA records from the file **rna.fna**
- Send output to the screen
- Not include the sequence in the output

HINTS

Write your program in stages making sure that each stage works before you start work on the next. For example, first write the program to process a single-sequence file, then modify it to accept command-line arguments, and finally modify it to process a multisequence file.

TESTING YOUR PROGRAM

A single-sequence FASTA file (**FASTARecord.fna**), a multisequence FASTA file (**human.rna.fna**), and an example output file (**human.rna.tab**) can be copied from Canvas (under Files/Assignments/Assignment03).

The Unix **diff** command compares two files line by line and displays those lines that are different. You can use this command to test the output of your program to make sure that it is working correctly. For example, if your output file for **human.rna.fna** is called **output.tab** you could compare it with mine (**human.rna.tab**) as follows:

```
[bioimav@bioi3500]$ diff human.rna.tab output.tab
```

As illustrated in this example, if there are no differences between the two files no output will be displayed and you will immediately get the Unix prompt back. This is one of the ways I will check your program. You will only get full credit if there are no differences between your output file and mine.