Maxwell Chibuogwu's Bio_Systems 534 Project

Finding Open Reading Frames

This python program would generate potential open reading frames (ORFs) from a given sequence of DNA or RNA using Frame 1. After conversion of the sequence to amino acids, the program will identify potential open reading frames, with the option of choosing the start codon i.e. the codon that the potential ORFs will start from. The ORF(s) will be printed to --output default: 'out.fa' in FastA format.

- NAME

```
orf_finder.py -- Finding the Open Reading Frames (ORFs)
```

- SYNOPSIS

```
orf_finder.py [-h] [-s str] [-m int] [-o outfile] DNA/RNA sequence
```

- DESCRIPTION

```
This program `orf_finder.py` will accept a text(nucleotide) sequence (DNA or RNA)
from the command line or a file with a nucleotide sequences. It will then look for
potential Open Reading Frames and write them to a named output file in FastA
format.
```

```
The program will first convert the DNA sequence to RNA. Then, the potential open
reading frames would be determined starting from the Default `AUG` codon or any of
the specified codons given with `-s` or `--start` option. Using the `-m` or `--
minlength` option will specify the minimum length of ORFs that should be found. The
default is 25 and the minimum ORF length that the program would generate is 20.
```

```
At the end, ORFs found will be printed in FastA format to a file, and the number of
ORFs found would be shown to the user. For example, "I found 2 ORFs, written to
out.fa."
```

- EXIT STATUS

> If no ORFs are found, the program will report several options depending on the iput given. For example, if the input text or file is empty, the program will report `There is no sequence here`, if there is no start or stop codons, the program will report such.

When run with no inputs, the program would print a brief usage statement:

```
$ ./orf_finder.py
usage: orf_finder.py [-h] [-s str] [-m int] [-o outfile] DNA/RNA sequence
orf_finder.py: error: the following arguments are required: DNA/RNA sequence
```

Or a longer usage for the -h or --help flags:

```
$ ./orf_finder.py -h
usage: orf_finder.py [-h] [-s str] [-m int] [-o outfile] DNA/RNA sequence

Finding the Open Reading Frames (ORFs)

positional arguments:
  DNA/RNA sequence      Input nucleotide sequence or file

optional arguments:
  -h, --help            show this help message and exit
  -s str, --start str   CODON to start reading (default: AUG)
  -m int, --minlength int
                        The minimum length of an ORF (default: 25)
  -o outfile, --outfile outfile
                        Output file (default: out.fa)
```

The program would reject a -m or --minlength less than 20.

```
$ ./orf_finder.py -m 15 inputs/input1
usage: orf_finder.py [-h] [-s str] [-m int] [-o outfile] DNA/RNA sequence
orf_finder.py: error: --minlength "15" must be greater than 20.
```

When given a good input, for example, a file (input1) in the 'inputs' directory with a DNA sequence, the amino acid sequence will be printed to the screen and the ORFs discovered to the outfile:

```
$ ./orf_finder.py inputs/input1
From this amino acid sequence

NFCAKPESMVASLNDGCCNAPSSSLRSVHANPGRGSAACRARSAR_QASVRSVGAEVSAQDSRVDRAVRSRHPRGPGCSTGSLHQG
LPCAWQFPRRQCV?YLAVPHRHQHGEELPGVPWKTAARQRCQLRGCGV?RRRSWPQGSRVPRALVVAG_NRRMCPSHHPATAQRPT
HGLDPA_VRRAE?RRHCQCHAMSGGYRALSNLPRSGGHRQSPATFAAGN_

I found 2 ORFs, written to "out.fa".
```

And the content of the out.fa file would be:

```
$ cat out.fa
>ORF 1
MVASLNDGCCNAPSSSLRSVHANPGRGSAACRARSAR
Found ORF with 37 amino acids.
>ORF 2
MSGGYRALSNLPRSGGHRQSPATFAAGN
Found ORF with 28 amino acids.
```

A passing test suite will look like this:

```
$ make test
pytest -xv --disable-pytest-warnings test.py orf_finder.py
============================== test session starts ===============================
...
collected 9 items

test.py::test_exists PASSED                                              [ 11%]
test.py::test_usage PASSED                                               [ 22%]
test.py::test_positional_argument PASSED                                 [ 33%]
test.py::test_empty_file PASSED                                          [ 44%]
test.py::test_bad_minlength PASSED                                       [ 55%]
test.py::test_good_input PASSED                                          [ 66%]
orf_finder.py::test_start_points PASSED                                  [ 77%]
orf_finder.py::test_find_orf PASSED                                      [ 88%]
orf_finder.py::test_sort_orf PASSED                                      [100%]


============================== 9 passed in 1.24s ===============================
```