

Python Dictionaries: Translate DNA/RNA to AA

Write a Python program called “translate_proteins.py” that translates a given DNA/RNA sequence to amino acids using a provided codon table. The output will be written to a file either provided by the user or a default of “out.txt”.

Expected Behavior

The DNA/RNA string and codon table are both required, so be sure to set `required=True` if creating with `parser.add_argument` so that your program produces a usage statement when no arguments are provided:

```
$ ./translate_proteins.py
usage: translate_proteins.py [-h] -c FILE [-o FILE] STR
translate_proteins.py: error: the following arguments are required: STR, -c/--codons
```

Create help page on `-h` or `--help`:

```
[cholla@~/work/worked_examples/05-python-proteins]$ ./translate_proteins.py -h
usage: translate_proteins.py [-h] -c FILE [-o FILE] STR
```

Translate DNA/RNA to proteins

positional arguments:

STR	DNA/RNA sequence
-----	------------------

optional arguments:

-h, --help	show this help message and exit
-c FILE, --codons FILE	A file with codon translations (default: None)
-o FILE, --outfile FILE	Output filename (default: out.txt)

Die on a bad `--codons` argument:

```
$ ./translate_proteins.py -c foo AAA
--codons "foo" is not a file
```

If given good input, write the results to the proper output file:

```
$ ./translate_proteins.py -c codons.rna UGGCCAUGGCGCCAGAACUGAGAUCAAUAGUACCCGUAUUAACGGGUGAA
Output written to "out.txt"
$ cat out.txt
WPRPELRSIVPVLIGE
$ ./translate_proteins.py -c codons.dna gaactacaccgttctcctggt -o dna.out
Output written to "dna.out"
```

```
$ cat dna.out
ELHRSPG
```

Note that you might (well, you definitely *will*) be given the wrong codon table for a given sequence type. If you are creating a dictionary from the codon table, e.g.:

```
$ head -3 codons.rna
AAA K
AAC N
AAG K
```

Such that you have something like this:

```
>>> codons = dict(AAA='K', AAC='N', AAG='K')
```

Everything is fine as long as you ask for codons that are defined but will fail at runtime if you ask for a codon that does not exist:

```
>>> codons['AAC']
'N'
>>> codons['AAT']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'AAT'
```

If a codon does not appear in the table, use “-” instead:

```
$ ./translate_proteins.py -c codons.rna gaactacaccgttctcctggt
Output written to "out.txt"
$ cat out.txt
E-H----
```

The “Python Patterns” has an example of how to “Extract Codons from DNA” that will help you.

Test Suite

Passing tests should look like:

```
$ make test
python3 -m pytest -v test.py
===== test session starts =====
platform darwin -- Python 3.7.0, pytest-3.8.0, py-1.6.0, pluggy-0.7.1 -- /anaconda3/bin/python
cachedir: .pytest_cache
rootdir: /Users/kyclark/work/worked_examples/05-python-proteins, inifile:
plugins: remotedata-0.3.0, openfiles-0.3.0, doctestplus-0.1.3, arraydiff-0.2
collected 4 items
```

```
test.py::test_usage PASSED
test.py::test_bad_args PASSED
test.py::test_bad_codon_file PASSED
test.py::test_valid_input PASSED
```

```
===== 4 passed in 0.47 seconds =====
```