

HW 11

Take the class and function definitions from the Lecture 20 notebook (you can use the answer key versions from Canvas) and put them in a .py file. Then, do the following:

- 1) Re-write `file_reader()` to be able to handle FASTA files that have sequences that span multiple lines, instead of having just one line of sequence for each entry.
- 2) Write a third function called `reverse_printer()` that takes as an argument a list of sequence objects and prints out on one line the sequence id, and on the second line, the reverse complement of each sequence.
- 3) Modify `main()` to remove any print statements. Instead, `main()` should pass the list of sequences returned by `file_reader` to `reverse_printer()`.

Save the complete class and function definitions in a file called HW_11.py and turn it in by Sunday 11/28 at midnight.

You can see any example of what your terminal output should look like here:

```
Please give the name of your FASTA file: my_data3.fasta
seq1
CTTGGCGTTTCGCGTACCCGAAAGGATGTACCCGTCGGATGGTGGGGCTGTCCGAAATGGGCGATGCATCAGATATTGATTTGGAGCAGGAAAAAGATCCCTCGGGGGTCACTCGGTGGGATGACCGTATAAGAAGTGAAGGGCCAGGACCAA
AAACATAGGTTCGGGTTTGGCAGACCCGTGAGGATCCCGAGTATAAAGCCGCTCTTAATTTCCAACTAGTTGCAGGAGGAACGCTCCTCGGCATCTTGGTAATCAGAGAAGCTACCGCTGCTTAACATAGATCCCTATGATAACCGACTTGG
AATCGCAGCGATTACATTAGGATACGATCTTCGAGTACTAATCAATAAAACAGTGCCTGCCTCCGACGAACTCAGTACACATGTAACAACAGAGTGGCTCGGAATCGCGGCGCCGTCGACGCTCCGAGATGTCTGCGTATGCATATTTCTCCGA
GTACCTTCTACCTAAACACTGGTCTGTGGAGGGAAGGTAAAGCAGAGGCGTCGCATACCATAGCAGTTGCATAGTCTACTTTTCCAACTTTCC
seq2
ACTTCCAAACCGGAACCCAGAAGATTACGGCGTGGACTGTACGATACGTTTACACTCATACCGAAAAGATATTGCCAGCAGCGGCAAGTATCAATCAGCTGTGACCCAGAGTCGGCGCTAAACACAGTGACTCTCGCACTGCTGTGCCAGT
AGTCCAAATTCATTCACGCGTGGGCAACCGCTCCCGCCGAGATATTAACATGAAATCCAGTGGGAAAGTCTCATAGCCTGAAACAGTAATATCTTGGTAAGGATCCATTAAACGCGGACATTTCCACACAGATCATGAGCTCCTATCTAAT
TACACAGAGGATTACGACCCAGATTGGCTACTCACAGATGTTGTTGGCGAACTAATTAAGATTCAATAGAGCTTATTGAAGAGTTTTTGTGCTTACGGTTGCCAAGATTATCCATACGGCCCCGGACATACCCACGTACGTATCTCAGACTCT
GTTGGAGCCAACCCGGCATTAAATACAGGAAGTCATAATTTTGGCTATGTGACTAGAGCTTCTGACAGCAAAAGGGCTGCACAGTCAAGTGAACCG
seq3
TTGTACGTGCGGCGAGCGGACTTTGTTTATGAACCAAGTCTCTAGCGATTCTGCACGTTGTTAATCTCAACGGAATGGGTAAACGAGTGTGTTAGAGGCCAACGCGATAAAGTGTTCGTGCGGGCTTAATACAGCGGAATAGTAGCAGTCTTACC
AATGAGCAAGGCTGAGGCTATGAGTTGCGGTGCACGACGCCCCGATCAGTACTCGGTGGGGGGACACGCCGTCGATGTGATGCGTAGAAAGTAAACGCCGCGCGCTACCCCTAAGCTGTTGATGGGCACGCTTCCATAGGTACTGTTTC
CTTACCCATCGACGACTCTGCTTGGACCCCTTGATAAGCGCGTCCCTTCGTATAATATAACGCCCTACAGACGGAATTTATACAGTTTGGAAATCTCGCATGTTATACACAACAGAGTCCGGTTGACTTATAGAAAAGAACGGGCCGAGTCCGCGT
GTTACATGTGGGTTAGTAAAGAAGCGTTCTAAAGTTTCATCTCAATACGGGTTGGTACTGCTACAGCTGATACATCCGTAGGCTATGCAAGCCGCT
seq4
CGTCCGACAGAAATACCTGCGGCGACTAATATGACAGATTAGTTGCCCGGAATTTATCCGCGCTCTAATGTAAGCATTGCATGTTCTGTGTAAGGTTTCGATTAACCTACTTGTGCCACCTGGATCAGGTGTGTGGTTAATGACTCTAACG
CGTTTACCTTCGGGAAACTTGGTACGTGGGTGAGAGGCCAATTGGTGTATTGTATTACTATATTTTATAGTTAGGCGGTTTAAAGCTATGACCGATATCCGCTACTACGACTATATCTTATGCTGGCATGTTAACTTTTCGTGGGTTTATG
GGTAATTCCAGCCGCTGCTGATAGTTCGCGGGGTAGGCGAGGGCAAGAGCTCGTAAAGAGGGTGAATACATGATAGGGAACACAGGATTCGCTGCACTTTGGCTCGCACACTCTCGAAAGTATTCGATTCGCGGTGATAGTGACCATTAGTC
ACTCTGACTCGTTGAACACGATCGGAATGAAAATAAACGTTGTTACACTTAGGGATTGCTGTTGCTGTGACGCTATCTACTAGTCCATTGTTG
seq5
GACTAGAAATGACAAATGGTAGACGCGGAATAGACCTTGGAACTTAGCCCTTTTATATATGTCACATGGAAGGGTATATACAGTAGTGAAGTTAAGGGAGTCGACCCGCTCTTAATAAGATGGTGTCCCCCTGAGTGAGTTGGTGAATTCGAGA
ATCGATTCCGACAGTATGAAGAGATACAGTACATGATATAGATGACCCCTTACAATTTGTCTAGCCCTCCGCGCAAGCCCTCGATTTCAGGCCATGGTTCCAAAGGGTTTCATACCGACGGAAGGCTCAATAGAACTCCCGGCTACGTTCCGTTAC
CTTCAAGCGTATTACCTTAAGGAATACCTTCTAACGCCAATACCCGTGCATATTAGCAGCTGCGGTGACACACTGGAACAGCGAAACCCGACTCGGAGAGAAGTGTGTGCGGACGCGGCTCATCACTCTCGGTGGCCACGGGTGATGTGCG
AGGTTTGAAGCGCGCGCTGTGAGAGTAACTCAGCAATTCTAGTCAGTACGCTTCCAGTCCACCCTCTGTGGCGCAAAACCCGGA
seq6
GATGCTAACGTGGCCGGGAGGCTGGTAAAATGTGGAAATGTAAACAGTCGACGAAACCGCACTCTAAGGCGTAGTTTCCATATCGTCGAAAGTCTAATGTGATCGCTTGGGGATCAACAGCAAGTGAAGGACATTTACCTCCACGAGCAT
GTGCGGACGCGGAACAGTCCACTGCCACACCCGATATGGGGTTGCGTCTTGGCTCTCCGCGGAGTGGGGGGAAGTGTGCTTCTACGTTTATAGCGTAGGGAGTCCCTATAGACTAGCATCAATTTACAACGCGGACGTCACCAACAACAACCA
GAGCCCCACAACCTTGCAATTTGTCAACGAATTGGGCGGTTCCCTTCAGCGACGCTGCTCTACTACGAAAACTCGAGAAGCACTATTGGTGTACAACCTATGCAAACTACACCGGTACACGCGCGAGCCGATATCCAGTTGACCTGGGAG
TGAAGGCTCCTCTGAGCTCGCGTGAAGTTAGTAACACCCCTGTTACCAAAAAGTTTATATGCCAGCAGGAAGTGCATCGCCAGCATCACAAA
```

HW 12

Create a new .py file named HW_12.py that contains just the class definition for `Sequence()`, from the Lecture 20 Workshop notebook but none of the above functions.

Add a new method to the `Sequence` class called `gene_translation()` that:

- 1) Calls the `start()` method to identify whether a start codon is present in a DNA sequence

2) If `start()` returns True, converts a DNA sequence to RNA by calling the `to_rna()` method. If `start()` returns False, then the method prints 'Sorry, no gene in this sequence' and stops running.

3) Slices the RNA sequence to begin at the start codon 'AUG'.

4) Generates a list of codons beginning with 'AUG' and continuing every three nucleotides until the end of the sequence. So the sequence 'AUGAGGACC' would generate the list ['AUG', 'AGG', 'ACC'].

5) Slices that list to contain everything from index 0 through the first occurrence of one of the following three stop codons: 'UAG', 'UAA', or 'UGA'. **Hint:** There are lot of ways you can find the first stop codon. I would probably use a list comprehension to make a list of all indices where those stop codons appear: `[i for i in range(len(mylist)) if mylist[i] in ['UAG', 'UAA', 'UGA']]` and slice my list up to and including the first index number in the resulting list.

6) Makes use of the dictionary given in the Lecture 20 Workshop to translate each triplet into the corresponding amino acid sequence and prints the translation to the terminal as a single string.

This method should require no arguments aside from self and returns nothing (just prints the amino acid sequence to the terminal).

After the end of the class definition, create a `main()` function that reads the file `cyto_pol.fasta` (containing the DNA sequence for the Human Cytomegalovirus polymerase gene), extracting the sequence id and the sequence, and then instantiates a Sequence object. Note that the sequence spans multiple lines of the fasta file, but the file only contains the one sequence entry. `main()` should then call `gene_translation()` on that Sequence object.

Submit your file as HW_12.py by midnight on Sunday 11/28.

Your terminal output should look like this:

Met Phe Phe Asn Pro Tyr Leu Ser Gly Gly Val Thr Gly Gly Ala Val Ala Gly Gly Arg Arg Gln Arg Ser Gln Pro Gly Ser Ala
Gln Gly Ser Gly Lys Arg Pro Pro Gln Lys Gln Phe Leu Gln Ile Val Pro Arg Gly Val Met Phe Asp Gly Gln Thr Gly Leu Ile
Lys His Lys Thr Gly Arg Leu Pro Leu Met Phe Tyr Arg Gly Ile Lys His Leu Leu Ser His Asp Met Val Trp Pro Cys Pro Trp
Arg Glu Thr Leu Val Gly Arg Val Val Gly Pro Ile Arg Phe His Thr Tyr Asp Gln Thr Asp Ala Val Leu Phe Phe Asp Ser Pro
Glu Asn Val Ser Pro Arg Tyr Arg Gln His Leu Val Pro Ser Gly Asn Val Leu Arg Phe Phe Gly Ala Thr Glu His Gly Tyr Ser
Ile Cys Val Asn Val Phe Gly Gln Arg Ser Tyr Phe Tyr Cys Glu Tyr Ser Asp Thr Asp Arg Leu Arg Glu Val Ile Ala Ser Val
Gly Glu Leu Val Pro Glu Pro Arg Thr Pro Tyr Ala Val Ser Val Thr Pro Ala Thr Lys Thr Ser Ile Tyr Gly Tyr Gly Thr Arg
Pro Val Pro Asp Leu Gln Cys Val Ser Ile Ser Asn Trp Thr Met Ala Arg Lys Ile Gly Glu Tyr Leu Leu Glu Gln Gly Phe Pro
Val Tyr Glu Val Arg Val Asp Pro Leu Thr Arg Leu Val Ile Asp Arg Arg Ile Thr Thr Phe Gly Trp Cys Ser Val Asn Arg Tyr
Asp Trp Arg Gln Gln Gly Arg Ala Ser Thr Cys Asp Ile Glu Val Asp Cys Asp Val Ser Asp Leu Val Ala Val Pro Asp Asp Ser
Ser Trp Pro Arg Tyr Arg Cys Leu Ser Phe Asp Ile Glu Cys Met Ser Gly Glu Gly Gly Phe Pro Cys Ala Glu Lys Ser Asp Asp
Ile Val Ile Gln Ile Ser Cys Val Cys Tyr Glu Thr Gly Gly Asn Thr Ala Val Asp Gln Gly Ile Pro Asn Gly Asn Asp Gly Arg
Gly Cys Thr Ser Glu Gly Val Ile Phe Gly His Ser Gly Leu His Leu Phe Thr Ile Gly Thr Cys Gly Gln Val Gly Pro Asp Val
Asp Val Tyr Glu Phe Pro Ser Glu Tyr Glu Leu Leu Gly Phe Met Leu Phe Phe Gln Arg Tyr Ala Pro Ala Phe Val Thr Gly
Tyr Asn Ile Asn Ser Phe Asp Leu Lys Tyr Ile Leu Thr Arg Leu Glu Tyr Leu Tyr Lys Val Asp Ser Gln Arg Phe Cys Lys Leu
Pro Thr Ala Gln Gly Gly Arg Phe Phe Leu His Ser Pro Ala Val Gly Phe Lys Arg Gln Tyr Ala Ala Ala Phe Pro Ser Ala Ser
His Asn Asn Pro Ala Ser Thr Ala Ala Thr Lys Val Tyr Ile Ala Gly Ser Val Val Ile Asp Met Tyr Pro Val Cys Met Ala Lys
Thr Asn Ser Pro Asn Tyr Met Asn Thr Met Ala Glu Leu Tyr Leu Arg Lys Asp Asp Leu Ser Tyr Lys Asp Ile Pro
Arg Cys Phe Val Ala Asn Ala Glu Gly Arg Ala Gln Val Gly Arg Tyr Cys Leu Gln Asp Ala Val Leu Val Arg Asp Leu Phe Asn
Thr Ile Asn Phe His Tyr Glu Ala Gly Ala Ile Ala Arg Leu Ala Lys Ile Pro Leu Arg Arg Val Ile Phe Asp Gly Gln Gln Ile
Arg Ile Tyr Thr Ser Leu Asp Glu Cys Ala Cys Arg Asp Phe Ile Leu Pro Asn His Tyr Ser Lys Gly Thr Val Pro Glu
Thr Asn Ser Val Ala Val Ser Pro Asn Ala Ala Ile Ile Ser Thr Ala Ala Val Pro Gly Asp Ala Gly Ser Val Ala Ala Met Phe
Gln Met Ser Pro Pro Leu Gln Ser Ala Pro Ser Ser Gln Asp Gly Val Ser Pro Gly Ser Gly Ser Asn Ser Ser Ser Val Gly
Val Phe Ser Val Gly Ser Gly Ser Ser Gly Gly Val Gly Val Ser Asn Asp Asn His Gly Ala Gly Gly Thr Ala Ala Val Ser Tyr
Gln Gly Ala Thr Val Phe Glu Pro Glu Val Gly Tyr Tyr Asn Asp Pro Val Ala Val Phe Asp Phe Ala Ser Leu Tyr Pro Ser Ile
Ile Met Ala His Asn Leu Cys Tyr Ser Thr Leu Leu Val Pro Gly Gly Glu Tyr Pro Val Asp Pro Ala Asp Val Tyr Ser Val Thr
Leu Glu Asn Gly Val Thr His Arg Phe Val Arg Ala Ser Val Arg Val Ser Val Leu Ser Glu Leu Leu Asn Lys Trp Val Ser Gln
Arg Arg Ala Val Arg Glu Cys Met Arg Glu Cys Gln Asp Pro Val Arg Arg Met Leu Leu Asp Lys Glu Gln Met Ala Leu Lys Val
Thr Cys Asn Ala Phe Tyr Gly Phe Thr Gly Val Val Asn Gly Met Met Pro Cys Leu Pro Ile Ala Ala Ser Ile Thr Arg Ile Gly
Arg Asp Met Leu Glu Arg Thr Ala Arg Phe Ile Lys Asp Asn Phe Ser Glu Pro Cys Phe Leu His Asn Phe Phe Asn Gln Glu Asp
Tyr Val Val Gly Thr Arg Glu Gly Asp Ser Glu Glu Ser Ser Ala Leu Pro Glu Gly Leu Glu Thr Ser Ser Gly Gly Ser Asn Glu
Arg Arg Val Glu Ala Arg Val Ile Tyr Gly Asp Thr Asp Ser Val Phe Val Arg Phe Arg Gly Leu Thr Pro Gln Ala Leu Val Ala
Arg Gly Pro Ser Leu Ala His Tyr Val Thr Ala Cys Leu Phe Val Glu Pro Val Lys Leu Glu Phe Glu Lys Val Phe Val Ser Leu
Met Met Ile Cys Lys Lys Arg Tyr Ile Gly Lys Val Glu Gly Ala Ser Gly Leu Ser Met Lys Gly Val Asp Leu Val Arg Lys Thr
Ala Cys Glu Phe Val Lys Gly Val Thr Arg Asp Val Leu Ser Leu Leu Phe Glu Asp Arg Glu Val Ser Glu Ala Ala Val Arg Leu
Ser Arg Leu Ser Leu Asp Glu Val Lys Lys Tyr Gly Val Pro Arg Gly Phe Trp Arg Ile Leu Arg Arg Leu Val Gln Ala Arg Asp
Asp Leu Tyr Leu His Arg Val Arg Val Glu Asp Leu Val Leu Ser Ser Val Leu Ser Lys Asp Ile Ser Leu Tyr Arg Gln Ser Asn
Leu Pro His Ile Ala Val Ile Lys Arg Leu Ala Ala Arg Ser Glu Glu Leu Pro Ser Val Gly Asp Arg Val Phe Tyr Val Leu Thr
Ala Pro Gly Val Arg Thr Ala Pro Gln Gly Ser Ser Asp Asn Gly Asp Ser Val Thr Ala Gly Val Val Ser Arg Ser Asp Ala Ile
Asp Gly Thr Asp Asp Asp Ala Asp Gly Gly Gly Val Glu Glu Ser Asn Arg Arg Gly Gly Glu Pro Ala Lys Lys Arg Ala Arg Lys
Pro Pro Ser Ala Val Cys Asn Tyr Glu Val Ala Glu Asp Pro Ser Tyr Val Arg Glu His Gly Val Pro Ile His Ala Asp Lys Tyr
Phe Glu Gln Val Leu Lys Ala Val Thr Asn Val Leu Ser Pro Val Phe Pro Gly Gly Glu Thr Ala Arg Lys Asp Lys Phe Leu His
Met Val Leu Pro Arg Arg Leu His Leu Glu Pro Ala Phe Leu Pro Tyr Ser Val Lys Ala His Glu Cys Cys STOP