

# Central Dogma Script

Created by **Georgios I. Iatropoulos**

City: **Piraeus**

Country: **Greece**

A project for Harvard's CS50 python 2023

# Description

- The Central Dogma script is a console application crafted to handle the input of a DNA sequence in plain text format, along with its orientation, according to the principles of the Central Dogma of Biology.
- It outputs the corresponding complementary DNA sequence, the equivalent mRNA sequence, and, if applicable, the corresponding polypeptide sequence.

- In addition, the script calculates the number of phosphodiester and hydrogen bonds, as well as the GC content for the double stranded molecule of each DNA sequence.

# Usage

```
python3 project.py [data.file]
```

# Script Files

1. **project.py**: It contains the **main()** function, and the rest of the custom functions that compose the procedural part of the script.
2. **bio.py**: It contains the translation\_table dictionary and the DNA\_obj custom class, which is imported in project.py.
3. **test\_project.py**: It contains the unit tests for the functions (procedural part) and the methods (object oriented part) of the script.

# Purpose

- Emulating the Central Dogma of Biology (CDB) procedures.
- Aiding ***secondary education biology students*** in understanding and applying the CDB procedural concepts.
- Assisting ***biology teachers*** in writing their own practical exercises.

# Operation modes

1. Manual input
2. Data file input

# Manual mode output

1. On Screen
2. Output file **dogma\_manual.txt**



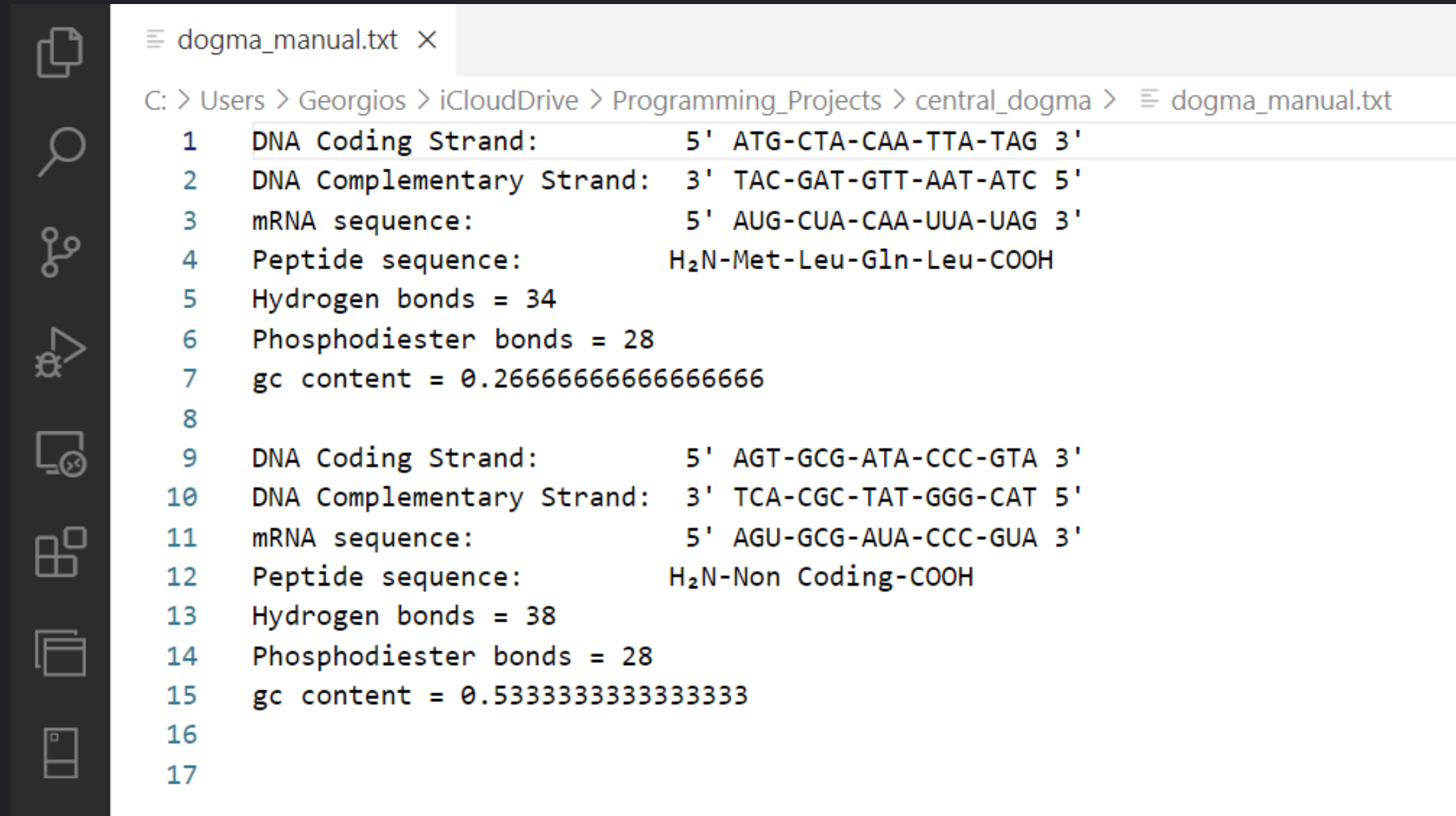
# Input is a Coding DNA Sequence

```
PS C:\Users\Georgios\iCloudDrive\Programming_Projects\central_dogma> python project.py
Existing output files may be overwritten.
Continue? (Y/N)y
If you want to load a data file for multiple coding sequences run this script as:
python3 central_dogma.py /path/name_of_file.extension
Enter your coding DNA sequence:
ATGCTACAATTATAG
Enter the direction of your sequence:
1 for 5->3
2 for 3->5
Enter your direction: 1
DNA Coding Strand:          5' ATG-CTA-CAA-TTA-TAG 3'
DNA Complementary Strand:   3' TAC-GAT-GTT-AAT-ATC 5'
mRNA sequence:              5' AUG-CUA-CAA-UUA-UAG 3'
Peptide sequence:           H2N-Met-Leu-Gln-Leu-COOH
Hydrogen bonds = 34
Phosphodiester bonds = 28
GC content = 0.266666666666666666
The output was added to the dogma_manual.txt file.
Do you want to continue? (Y/N)y
```

# Input is a Non Coding DNA Sequence - Inverted Orientation

```
Enter your coding DNA sequence:
ATGCCCATAGCGTGA
Enter the direction of your sequence:
1 for 5->3
2 for 3->5
Enter your direction: 2
DNA Coding Strand:          5' AGT-GCG-ATA-CCC-GTA 3'
DNA Complementary Strand:   3' TCA-CGC-TAT-GGG-CAT 5'
mRNA sequence:              5' AGU-GCG-AUA-CCC-GUA 3'
Peptide sequence:           H2N-Non Coding-COOH
Hydrogen bonds = 38
Phosphodiester bonds = 28
GC content = 0.5333333333333333
The output was added to the dogma_manual.txt file.
Do you want to continue? (Y/N)n
PS C:\Users\Georgios\iCloudDrive\Programming_Projects\central_dogma>
```

# dogma\_manual.txt output file sample



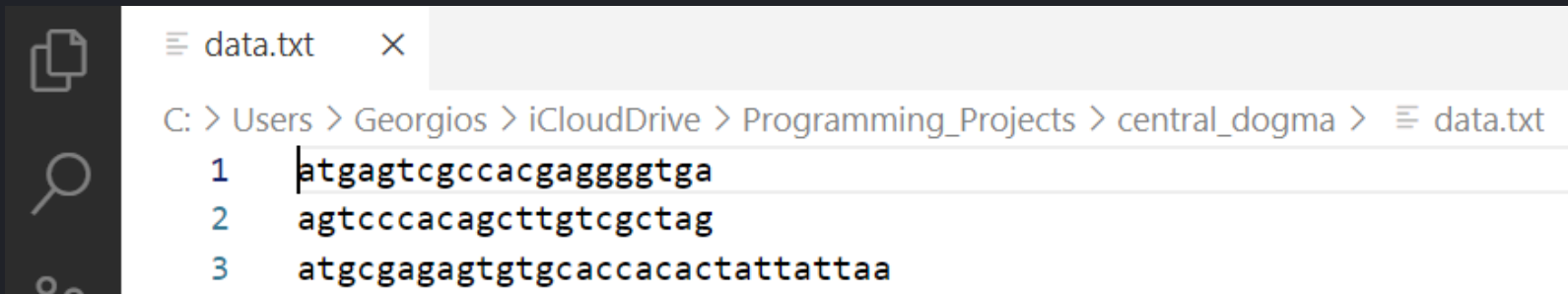
```
≡ dogma_manual.txt X
C: > Users > Georgios > iCloudDrive > Programming_Projects > central_dogma > ≡ dogma_manual.txt
1  DNA Coding Strand:          5' ATG-CTA-CAA-TTA-TAG 3'
2  DNA Complementary Strand:  3' TAC-GAT-GTT-AAT-ATC 5'
3  mRNA sequence:             5' AUG-CUA-CAA-UUA-UAG 3'
4  Peptide sequence:          H2N-Met-Leu-Gln-Leu-COOH
5  Hydrogen bonds = 34
6  Phosphodiester bonds = 28
7  gc content = 0.266666666666666666
8
9  DNA Coding Strand:          5' AGT-GCG-ATA-CCC-GTA 3'
10 DNA Complementary Strand:  3' TCA-CGC-TAT-GGG-CAT 5'
11 mRNA sequence:             5' AGU-GCG-AUA-CCC-GUA 3'
12 Peptide sequence:          H2N-Non Coding-COOH
13 Hydrogen bonds = 38
14 Phosphodiester bonds = 28
15 gc content = 0.5333333333333333
16
17
```

# Invalid DNA sequence input

```
Windows PowerShell x + v
PS C:\Users\Georgios iCloudDrive\Programming_Projects\central_dogma> python project.py
Existing output files may be overwritten.
Continue? (Y/N)y
If you want to load a data file for multiple coding sequences run this script as:
python3 central_dogma.py /path/name_of_file.extension
Enter your coding DNA sequence:
atgfgtagatga3agatagacg
Enter the direction of your sequence:
1 for 5->3
2 for 3->5
Enter your direction: 2
Traceback (most recent call last):
  File "C:\Users\Georgios iCloudDrive\Programming_Projects\central_dogma\project.py", line 175, in <module>
    main()
  File "C:\Users\Georgios iCloudDrive\Programming_Projects\central_dogma\project.py", line 60, in main
    dna_object = set_coding_sequence(d_seq, d)
                  ^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "C:\Users\Georgios iCloudDrive\Programming_Projects\central_dogma\project.py", line 76, in set_coding_sequence
    dna_obj = DNA_obj(c_seq, direct)
               ^^^^^^^^^^^^^^^^^^^
  File "C:\Users\Georgios iCloudDrive\Programming_Projects\central_dogma\bio.py", line 73, in __init__
    self.seq = seq.strip().upper()
                ^^^^^^^
  File "C:\Users\Georgios iCloudDrive\Programming_Projects\central_dogma\bio.py", line 92, in seq
    raise ValueError("Invalid Sequence!")
ValueError: Invalid Sequence!
PS C:\Users\Georgios iCloudDrive\Programming_Projects\central_dogma>
```

**Data file input**

# data.txt input file sample

A screenshot of a text editor window. The title bar shows 'data.txt' with a close button. The breadcrumb path is 'C: > Users > Georgios > iCloudDrive > Programming\_Projects > central\_dogma > data.txt'. The editor contains three lines of DNA sequence, each preceded by a line number (1, 2, 3) in blue. Line 1 is 'atgagtcgccacgaggggtga', line 2 is 'agtccacagcttgctcgtag', and line 3 is 'atgcgagagtgtgcaccacactattattaa'.

```
data.txt  
C: > Users > Georgios > iCloudDrive > Programming_Projects > central_dogma > data.txt  
1 atgagtcgccacgaggggtga  
2 agtccacagcttgctcgtag  
3 atgcgagagtgtgcaccacactattattaa
```

# Data file mode output

1. On Screen

2. Output files:

- dogma.txt
- peptides.csv
- peptides\_tab.txt

# Invalid data filename output

```
PS C:\Users\Georgios\iCloudDrive\Programming_Projects\central_dogma> python project.py wrong_filename.txt
Existing output files may be overwritten.
Continue? (Y/N)y
Coding Sequence file not found.
Check that the path and the name of your sequence file are correct,
or run the program without arguments and input a coding DNA sequence manually.
PS C:\Users\Georgios\iCloudDrive\Programming_Projects\central_dogma>
```



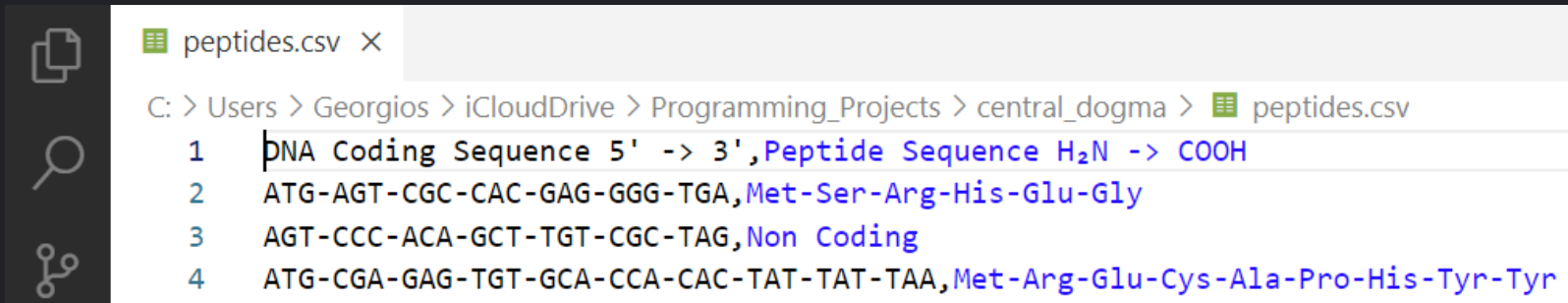
# Valid data filename output

```
PS C:\Users\Georgios\iCloudDrive\Programming_Projects\central_dogma> python project.py data.txt
Existing output files may be overwritten.
Continue? (Y/N)y
Your data file exists!
Read the dogma.txt file for Replication, Transcription and Translation data., plus various calculations for each DNA sequence.
Your translation data is written in the peptides.csv file.
For tabular format the peptides_tab.txt was created.
PS C:\Users\Georgios\iCloudDrive\Programming_Projects\central_dogma> |
```

# dogma.txt output file sample

```
≡ dogma.txt X
C: > Users > Georgios > iCloudDrive > Programming_Projects > central_dogma > ≡ dogma.txt
1 DNA Coding Strand: 5' ATG-AGT-CGC-CAC-GAG-GGG-TGA 3'
2 DNA Complementary Strand: 3' TAC-TCA-GCG-GTG-CTC-CCC-ACT 5'
3 mRNA sequence: 5' AUG-AGU-CGC-CAC-GAG-GGG-UGA 3'
4 Peptide sequence: H2N-Met-Ser-Arg-His-Glu-Gly-COOH
5 Hydrogen bonds = 55
6 Phosphodiester bonds = 40
7 gc content = 0.6190476190476191
8
9 DNA Coding Strand: 5' AGT-CCC-ACA-GCT-TGT-CGC-TAG 3'
10 DNA Complementary Strand: 3' TCA-GGG-TGT-CGA-ACA-GCG-ATC 5'
11 mRNA sequence: 5' AGU-CCC-ACA-GCU-UGU-CGC-UAG 3'
12 Peptide sequence: H2N-Non Coding-COOH
13 Hydrogen bonds = 54
14 Phosphodiester bonds = 40
15 gc content = 0.5714285714285714
16
17 DNA Coding Strand: 5' ATG-CGA-GAG-TGT-GCA-CCA-CAC-TAT-TAT-TAA 3'
18 DNA Complementary Strand: 3' TAC-GCT-CTC-ACA-CGT-GGT-GTG-ATA-ATA-ATT 5'
19 mRNA sequence: 5' AUG-CGA-GAG-UGU-GCA-CCA-CAC-UAU-UAU-UAA 3'
20 Peptide sequence: H2N-Met-Arg-Glu-Cys-Ala-Pro-His-Tyr-Tyr-COOH
21 Hydrogen bonds = 72
22 Phosphodiester bonds = 58
23 gc content = 0.4
```

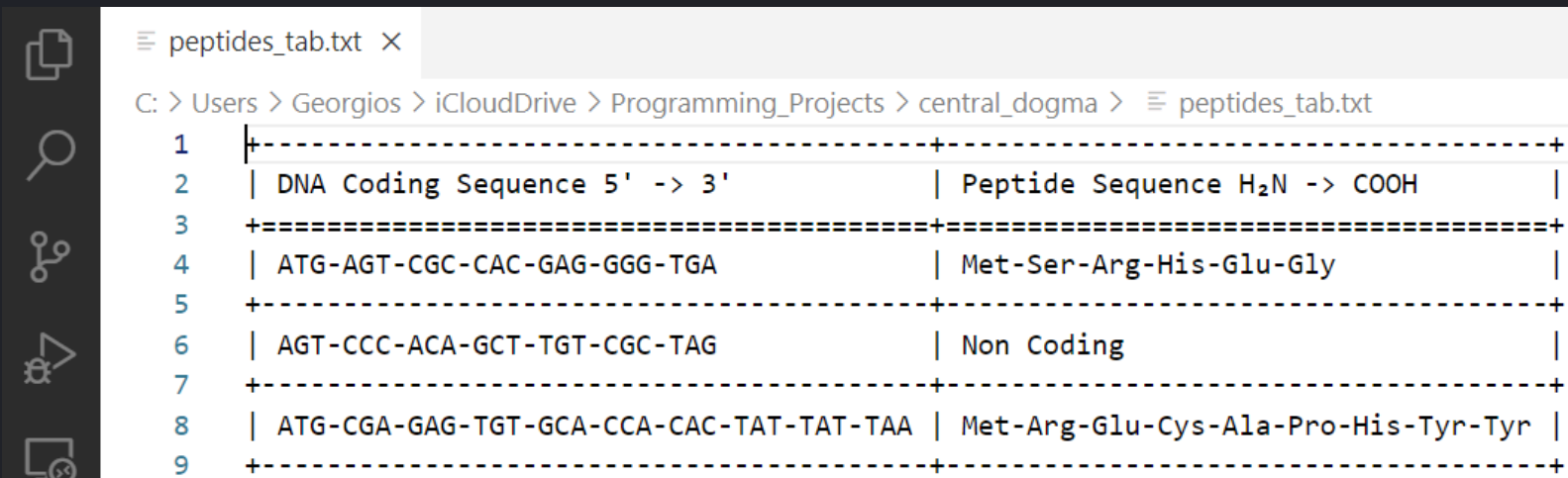
# peptides.csv output file sample



The screenshot shows a code editor window with a tab labeled 'peptides.csv'. The file path is 'C: > Users > Georgios > iCloudDrive > Programming\_Projects > central\_dogma > peptides.csv'. The file contains four lines of text, each representing a row in a CSV file. The first line is a header with two columns: 'DNA Coding Sequence 5' -> 3'' and 'Peptide Sequence H<sub>2</sub>N -> COOH'. The subsequent three lines contain DNA sequences followed by their corresponding peptide sequences, separated by commas. The peptide sequences are color-coded: Met (blue), Ser (green), Arg (red), His (purple), Glu (orange), Gly (brown), Cys (pink), Ala (light blue), Pro (light green), Tyr (light purple), and Phe (light orange).

```
1 DNA Coding Sequence 5' -> 3',Peptide Sequence H2N -> COOH
2 ATG-AGT-CGC-CAC-GAG-GGG-TGA,Met-Ser-Arg-His-Glu-Gly
3 AGT-CCC-ACA-GCT-TGT-CGC-TAG,Non Coding
4 ATG-CGA-GAG-TGT-GCA-CCA-CAC-TAT-TAT-TAA,Met-Arg-Glu-Cys-Ala-Pro-His-Tyr-Tyr
```

# peptides\_tab.txt output file sample



peptides\_tab.txt

C: > Users > Georgios > iCloudDrive > Programming\_Projects > central\_dogma > peptides\_tab.txt

1	+-----+-----+	
2	DNA Coding Sequence 5' -> 3'	Peptide Sequence H <sub>2</sub> N -> COOH
3	+=====+	
4	ATG-AGT-CGC-CAC-GAG-GGG-TGA	Met-Ser-Arg-His-Glu-Gly
5	+-----+-----+	
6	AGT-CCC-ACA-GCT-TGT-CGC-TAG	Non Coding
7	+-----+-----+	
8	ATG-CGA-GAG-TGT-GCA-CCA-CAC-TAT-TAT-TAA	Met-Arg-Glu-Cys-Ala-Pro-His-Tyr-Tyr
9	+-----+-----+	

# Central Dogma of Biology Summary

## Table (Central Dogma Overview)

**Table 2: DNA Coding Sequence Replication, Transcription and Translation (Central Dogma)**

Procedure	Macromolecule	Starting Edge	Starting Codon	Codon 1	Codon 2	Termination Codon	Ending Edge
Replication	DNA coding	5'	ATG	GAG	CTC	TAA	3'
	DNA complementary	3'	TAC	CTC	GAG	ATT	5'
Transcription	mRNA	5'	AUG	GAG	CUC	UAA	3'
Translation	Polypeptide	H <sub>2</sub> N	Met	Glu	Leu	-	COOH

# Future plans

- Adding non standard genetic tables.
- Introducing a new feature for translating a DNA coding sequence with 2 different genetic tables. Then the script will align the 2 corresponding peptides, counting their differences in amino acids.

**The End :)**