COM.e PROJECT: Investigating DNA in Python

Utsav Choudhury L24i

TARGET ASSESSMENT LEVEL: 3

1 Preface

[Jump ahead to the specification if this is boring.]

In Marjo Petäjäaho's BIO4 course, we are studying DNA and protein synthesis. Interestingly, DNA is the body's way of conveying information in code. After realizing that this project's purpose was text processing, I decided that processing theoretical DNA strands might be interesting. Of course, the project is highly theoretical, not taking mutations and specific complex processes into account; hence, it cannot be used in real-world applications. Nevertheless, it offers quite an interesting insight into how the body might theoretically process DNA.

2 Background Theory

DNA is a very complex molecule; however, the way it conveys information is quite understandable. DNA strands contain the organic bases adenine, cytosine, guanine, and thymine, commonly abbreviated as "A," "C," "G," and "T." These organic bases exhibit a characteristic: complementary pairing. A always bonds with T, and G always bonds with C.

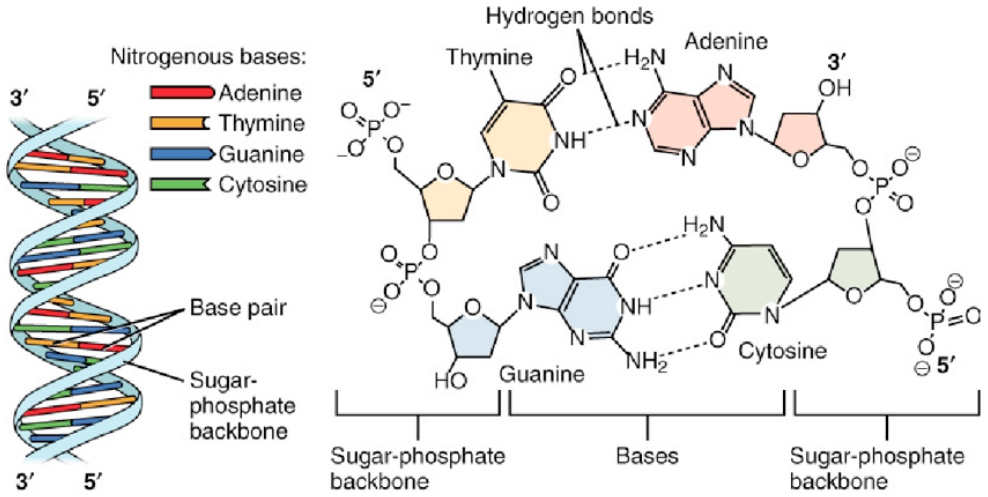

Figure 1: DNA Molecule [**LibreTexts**]

Complementary pairing can be used to our advantage. When DNA replicates, its two strands unwind. When one strand becomes split, transcription can take place. The singular strand pairs with its complementary counterpart. However, thymine is replaced by uracil (T becomes U). The newly formed strand is called mRNA, which is composed of many different triplets called codons. These codons are either code for a unique amino acid or punctuation. "codon tables" organize this information (Figure 3).

Punctuation means that codons can indicate "START" or "STOP," signaling when to start or stop the amino acid chain. Translation is the process where amino acids connect to their codon counterparts and form amino acid chains. When these amino acids link up, they form proteins, which are fundamental to all bodily processes. This whole process is simply an algorithm where characters and strings are processed, making it relevant for a CS project [**Britannica**].

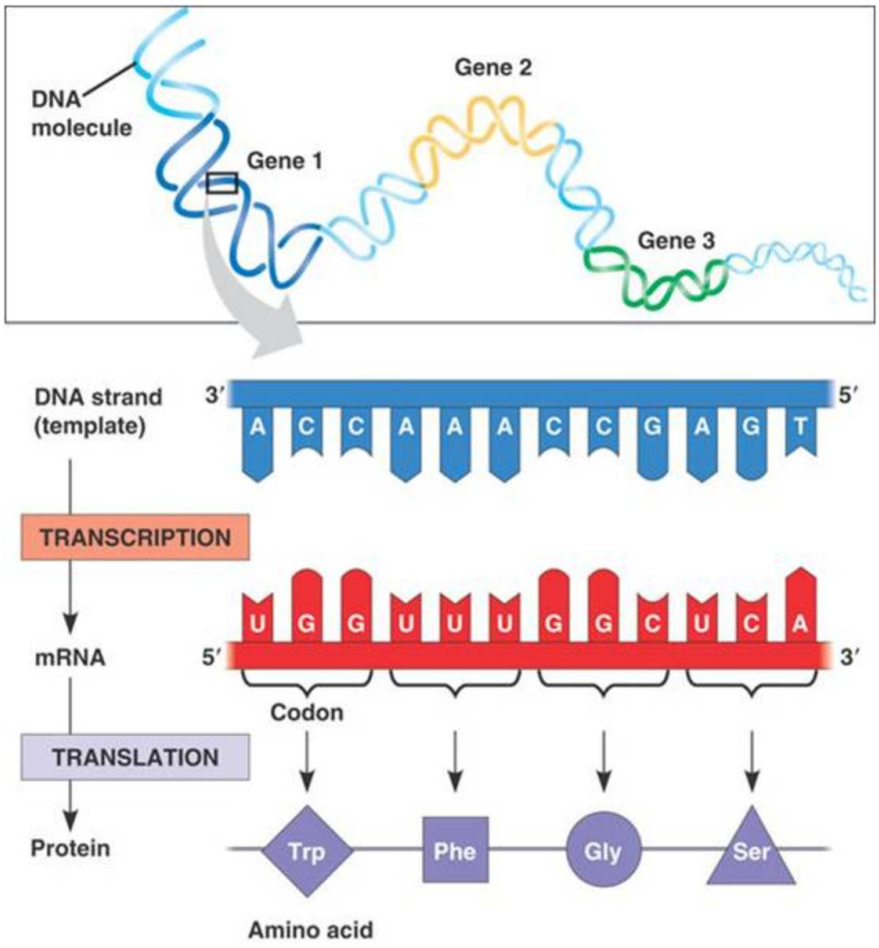

Figure 2: Transcription and Translation [**Owlcation**]

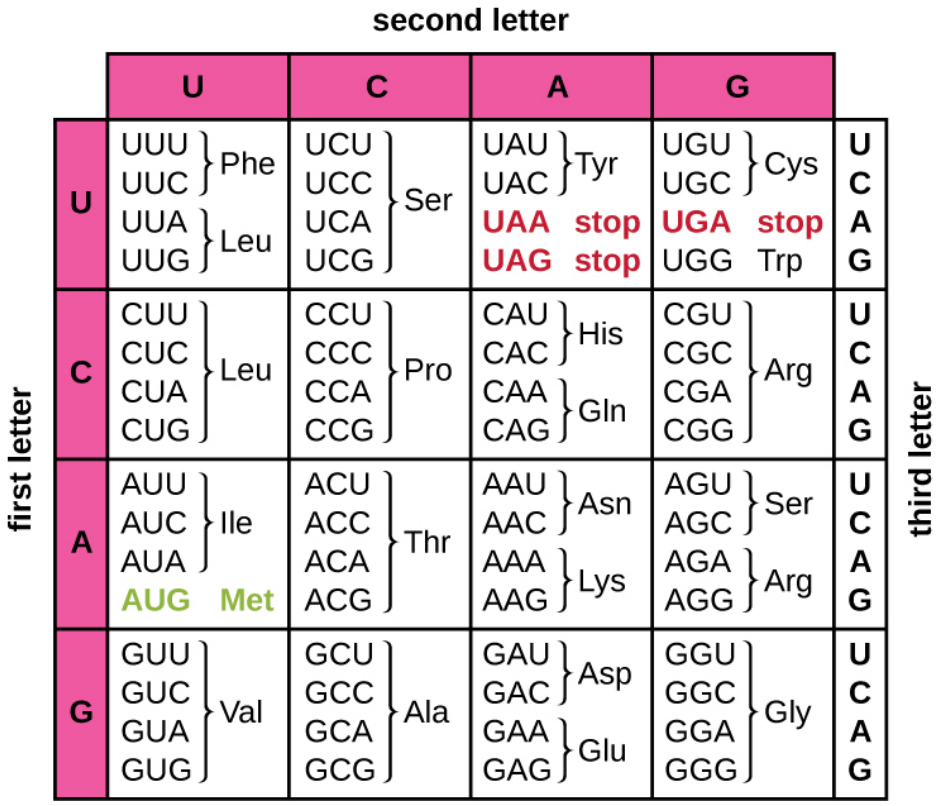

Figure 3: Codon Table [**Vedantu**]

3 Specification

3.1 What does the program do?

The program takes in different pieces of data. First, it takes in a filename where output will be stored (one for text in a `.txt` file). Then, the program asks the user whether they want a random DNA sequence or a custom one. If the user wants a random sequence, they will be asked for a length. A random sequence will then be generated. If the user chooses custom, they will be prompted to enter a custom sequence. In both cases, the user must enter their input in a specific format, following certain criteria. The random or custom sequence will then be transcribed (see Figure 2). During transcription, start codons are added, and stop codons are deleted. After transcription, the mRNA will be translated to its amino acid counterparts (see Figure 3). It does this with a dictionary in the file `Codon_Table.py`. Finally, the amino acids will be strung together. All of these conversions are documented and written into a `.txt` file with the name that the user inputted.

3.2 Data Format

The user can select any file name for their report (except "" or the absence of value). File extensions do not matter.

The user must enter "custom" or "random", where capitalization does not matter.

If "random" is chosen, a length l_1 must be selected such that:

$$9 \leq l_1 \leq 30 \quad \text{and} \quad l_1 \equiv 0 \pmod 3.$$

If "custom" is chosen, the string must be composed only of the letters "A," "C," "G," and "T," and a length l_2 must be selected such that:

$$9 \leq l_2 \leq 30 \quad \text{and} \quad l_2 \equiv 0 \pmod 3.$$

Capitalization does not matter in any case.

4 Correctness

4.1 Ideal Test Case

In this project's GitHub repository, I have linked some files documenting input and output cases. However, it must be noted that input occurs through the Python console, not through file uploads. Obviously, the date will vary, as it changes.

4.1.1 Example Console Input

```
C:\Users\utsav.choudhury\PycharmProjects\CS_PROJECT\.venv\Scripts\python.exe
    C:\Users\utsav.choudhury\PycharmProjects\CS_PROJECT\CS_PROJECT_COPY.py
Enter a filename for your report
No file extension needed. However, please note that if a file with this name
    already exists, it will be overwritten: test
Enter 'CUSTOM' for a custom sequence or 'RANDOM' for a random sequence: custom
Enter your custom DNA sequence (A, C, G, T). Length must be 9-30 and divisible by
    3: AGTCCCTCT
Process completed. Check test.txt for information.

Process finished with exit code 0
```

4.1.2 Example Text File Output

```
REPORT

DATE: 2025-01-31

RANDOM BASE ARRANGEMENT: AGTCCCTCT

RNA sequence: UCAGGGAGA

Codons: ['AUG', 'UCA', 'GGG', 'AGA']

Amino Acids: ['Met', 'Ser', 'Gly', 'Arg']

FINAL CHAIN: Met-Ser-Gly-Arg

NOTE: NOT A FULLY ACCURATE REPRESENTATION. NOT APPLICABLE TO REAL LIFE CONTEXTS.

UTSAV CHOUDHURY 2025
```

4.2 Resource Management

The input file is opened using a `with-statement`, and will therefore be closed automatically.

References

[1] Britannica. *DNA: Discovery, Function, Facts, and Structure.* 2018. https://www.britannica.com/science/DNA

[2] Britannica. *Metabolism: The Synthesis of Macromolecules.* https://www.britannica.com/science/metabolism/The-synthesis-of-macromolecules

[3] LibreTexts. *9.1: The Structure of DNA.* 2017. https://bio.libretexts.org/Bookshelves/Introductory$_a nd_G eneral_B iology/Con$
$olecular_B iology/9.01 :_T he_S tructure_o f_D NA$

[4] Rhys Baker. *Protein Production: A Simple Summary of Transcription and Translation.* 2012. https://owlcation.com/stem/protein-production-a-step-by-step-illustrated-guide

[5] Vedantu. *How Do You Read a Codon Table?* https://www.vedantu.com/question-answer/how-do-you-read-a-codon-table-class-11-biology-cbse-60ed1177a183f842ef1654c2