# DS2 Project Report - L3 - Group 6

By Rubab Shah, Yashfeen Zehra Zaidi, Fatima Dossa, and Zuhair Abbas

**Project Title: Genome Detective**

## 1. Introduction

In the field of bioinformatics, there are several challenges that researchers and healthcare professionals face when it comes to analyzing genetic data effectively. These challenges include the absence of user-friendly tools for comparing gene sequences, reliance on time-consuming diagnostic procedures for disease detection, and a lack of intuitive visualization techniques for genetic sequences. Addressing these challenges is crucial for advancing our understanding of genetic disorders and improving patient care.

To tackle these issues, we developed GenomeDetective, a comprehensive tool that facilitates healthcare professionals in their genetic analysis endeavors. We have used an advanced data structure, the hash map, in our code and offer three features:
a gene comparator, detection of diseases caused by mutation and lastly, a sequence logo.

### 1.1. Background to some basic Genetic biology

Nucleotides are the building blocks of DNA and RNA, consisting of a sugar molecule, a phosphate group, and a nitrogenous base. The four types of nitrogenous bases found in DNA are adenine (A), thymine (T), cytosine (C), and guanine (G). Meanwhile, codons are sequences of three nucleotides in mRNA that correspond to specific amino acids during protein synthesis. There are 64 possible codons, each encoding for either an amino acid or a stop signal. The genetic code is redundant, meaning that some amino acids can be encoded by more than one codon. Genes are specific segments of DNA which code for a particular protein.

Genetic diseases result from mutations in DNA that alter the normal function of genes, i.e changes in nucleotides and their sequences. These mutations can occur spontaneously or be inherited from parents and can affect various aspects of cellular function, leading to a wide range of disorders. Understanding the genetic basis of diseases is essential for diagnosis, treatment, and prevention strategies.

In the context of GenomeDetective, knowledge of these basic genetic concepts is crucial for interpreting gene sequences, identifying mutations associated with diseases, and understanding the significance of sequence logos in visualizing genetic data.

## 2. Our approach

### 2.1. Data Structure used

We used a hash map as it has O(1) for inserting and searching. This is important given that our data is stored in string sequences of only four letters, and we have to convert from this to groups of 3. Generally, it was used in two ways:

1. To convert between codon (group of three letters) and amino acid (special letter). So, for instance, CAG maps to G, AUG maps to M, and so on.
2. To keep all positions of amino acids (special letters) in the amino acid sequence (random sequence of letters). So, for instance, K maps to indices 2, 5, 81, etc.

### 2.2. Features:

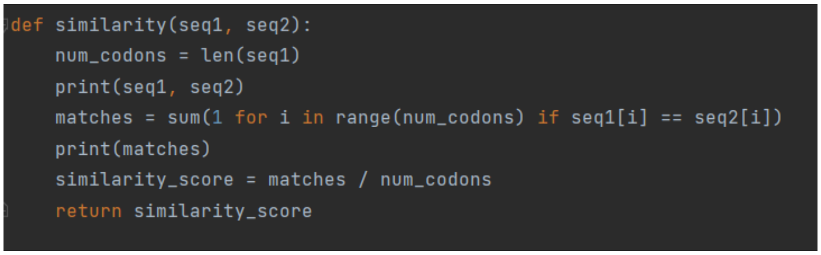
#### 2.2.1. Sequence Comparison

In this feature, the user is prompted to input two sequences into designated fields which need to contain only valid characters ('A', 'T', 'G', or 'C'). Once validated, both sequences undergo preprocessing steps to ensure consistency and remove unnecessary characters such as spaces or digits. Subsequently, the program verifies that both sequences have the same length and are divisible by 3, essential for accurate codon comparisons. The sequences are then translated into amino acid sequences, with each codon mapped to its corresponding amino acid. Following translation, the program compares each corresponding amino acid in the two sequences, calculating the percentage of matches to determine similarity.

**Note: The order of the codon matters as each codon codes for an amino acid and each amino acid sequence entails a specific protein.**

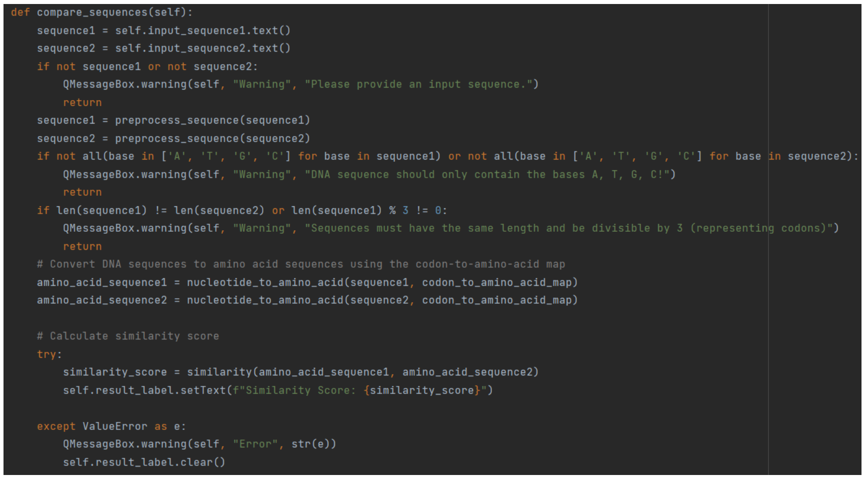
The result, presented as a percentage, indicates the degree of similarity between the two gene sequences. Throughout the process, error-handling mechanisms are in place to address any issues such as sequence length discrepancies. This systematic approach ensures robust and reliable comparison functionality, empowering users to analyze gene sequences effectively.

Similarity calculation on the number of similar codons and their order:



```python
def similarity(seq1, seq2):
    num_codons = len(seq1)
    print(seq1, seq2)
    matches = sum(1 for i in range(num_codons) if seq1[i] == seq2[i])
    print(matches)
    similarity_score = matches / num_codons
    return similarity_score
```

Validation of correct input:



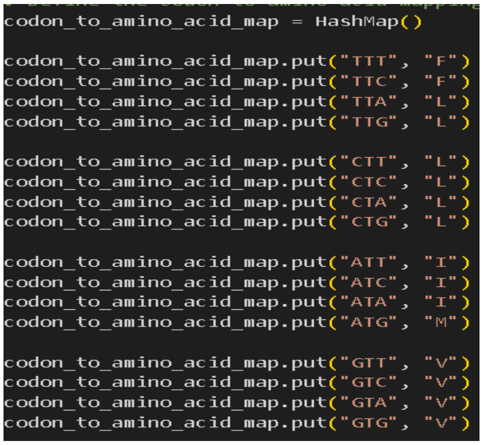
```python
def compare_sequences(self):
    sequence1 = self.input_sequence1.text()
    sequence2 = self.input_sequence2.text()
    if not sequence1 or not sequence2:
        QMessageBox.warning(self, "Warning", "Please provide an input sequence.")
        return
    sequence1 = preprocess_sequence(sequence1)
    sequence2 = preprocess_sequence(sequence2)
    if not all(base in ['A', 'T', 'G', 'C'] for base in sequence1) or not all(base in ['A', 'T', 'G', 'C'] for base in sequence2):
        QMessageBox.warning(self, "Warning", "DNA sequence should only contain the bases A, T, G, C!")
        return
    if len(sequence1) != len(sequence2) or len(sequence1) % 3 != 0:
        QMessageBox.warning(self, "Warning", "Sequences must have the same length and be divisible by 3 (representing codons)")
        return
    # Convert DNA sequences to amino acid sequences using the codon-to-amino-acid map
    amino_acid_sequence1 = nucleotide_to_amino_acid(sequence1, codon_to_amino_acid_map)
    amino_acid_sequence2 = nucleotide_to_amino_acid(sequence2, codon_to_amino_acid_map)

    # Calculate similarity score
    try:
        similarity_score = similarity(amino_acid_sequence1, amino_acid_sequence2)
        self.result_label.setText(f"Similarity Score: {similarity_score}")

    except ValueError as e:
        QMessageBox.warning(self, "Error", str(e))
        self.result_label.clear()
```

### 2.2.2. Disease Detection

The disease detection feature detects the diseases that have been caused by certain mutations or if the person is at risk of developing a particular disease. It is essential to diagnose these diseases since they can lead to a wide range of health problems including developmental delays, physical disabilities, and metabolic disorders among others. For the scope of this project, we implemented the detection of 10 such diseases: Sickle Cell Anemia, Huntington's Disease, Cystic Fibrosis, Fragile X Syndrome, Duchenne Muscular Dystrophy, Tay-Sachs Disease, Familial Hypercholesterolemia, Beta-Thalassemia, Hemochromatosis, Alkaptonuria. We first got the gene data for the particular diseases through the NCBI platform and then cleaned it by removing spaces and numbers. After that, we made functions for each disease with the help of our research as to which type of mutation it is and what position it occurs in. We tested our functions by manually changing our gene sequences into mutated ones to check if the functions were working properly.

### 2.2.3. Sequence Logo

The sequence logo has two functions. Firstly, it generates a histogram of amino acids (special groups of 3 in the original sequence) against their frequency in the sequence. Secondly, it looks for special sequences of amino acids (e.g: GAK, TER, etc.) in the sequence and highlights their positions in a visual format. The mapping has been done like this.

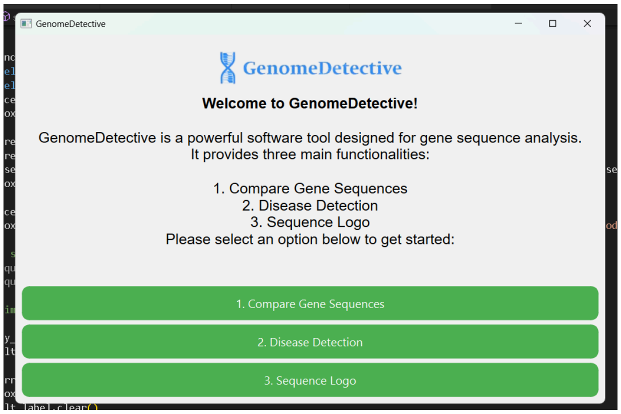


```
# Define the codon to amino acid mapping
codon_to_amino_acid_map = HashMap()

codon_to_amino_acid_map.put("TTT", "F")
codon_to_amino_acid_map.put("TTC", "F")
codon_to_amino_acid_map.put("TTA", "L")
codon_to_amino_acid_map.put("TTG", "L")

codon_to_amino_acid_map.put("CTT", "L")
codon_to_amino_acid_map.put("CTC", "L")
codon_to_amino_acid_map.put("CTA", "L")
codon_to_amino_acid_map.put("CTG", "L")

codon_to_amino_acid_map.put("ATT", "I")
codon_to_amino_acid_map.put("ATC", "I")
codon_to_amino_acid_map.put("ATA", "I")
codon_to_amino_acid_map.put("ATG", "M")

codon_to_amino_acid_map.put("GTT", "V")
codon_to_amino_acid_map.put("GTC", "V")
codon_to_amino_acid_map.put("GTA", "V")
codon_to_amino_acid_map.put("GTG", "V")
```

## 3. <u>Its application</u>

The main window looks like this:

a)



Welcome to GenomeDetective!

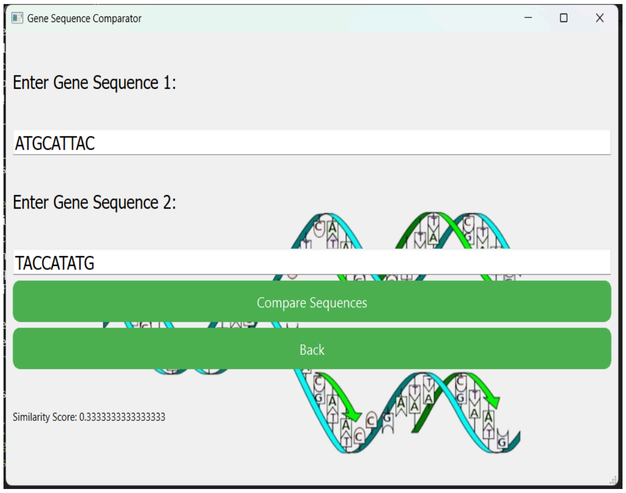
GenomeDetective is a powerful software tool designed for gene sequence analysis.
It provides three main functionalities:

1. Compare Gene Sequences
2. Disease Detection
3. Sequence Logo

Please select an option below to get started:

## b) Sequence Comparison:

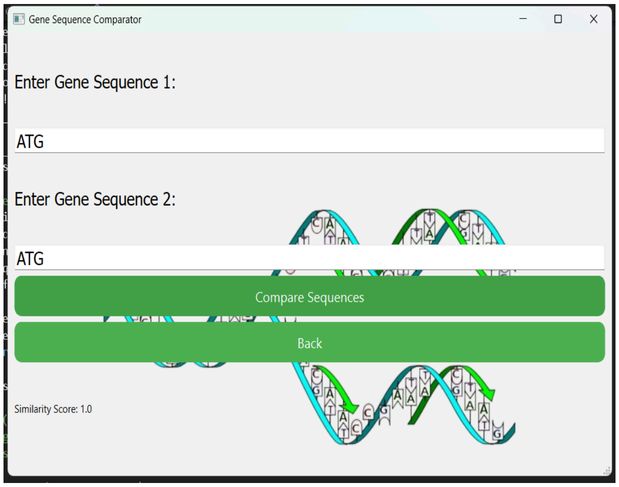
This feature is particularly beneficial in a laboratory environment, as it can be used for quality control purposes, ensuring that gene sequences obtained through sequencing techniques are accurate and reliable as well as for educational purposes.



Enter Gene Sequence 1: ATGCATTAC

Enter Gene Sequence 2: TACCATATG

Similarity Score: 0.3333333333333333



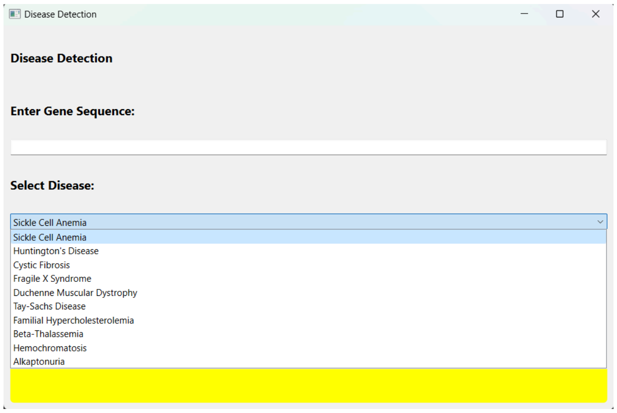
Enter Gene Sequence 1: ATG

Enter Gene Sequence 2: ATG

Similarity Score: 1.0

## C) Disease Detection:

Our app has been designed to detect 10 diseases which are seen in this snapshot below with 2 examples as well - BetaThalassemia and Familial Hypercholesterolemia:



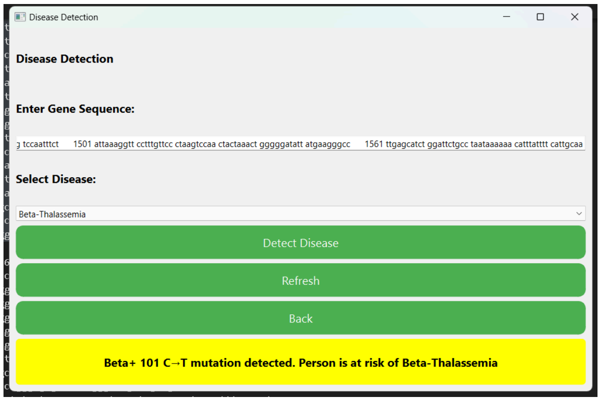
Disease Detection

**Disease Detection**

**Enter Gene Sequence:**

**Select Disease:** Sickle Cell Anemia

## -Bethathalassemia:



Disease Detection

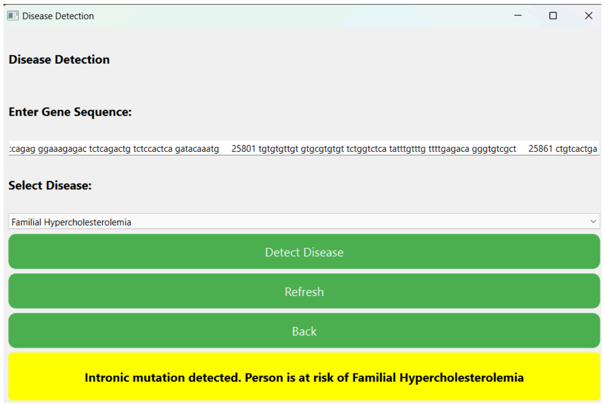
**Disease Detection**

**Enter Gene Sequence:** g tccaatttct 1501 attaaaggtt cctttgttcc ctaagtccaa ctactaaact gggggatatt atgaagggcc 1561 ttgagcatct ggattctgcc taataaaaaa catttatttt cattgcaa

**Select Disease:** Beta-Thalassemia

**Beta+ 101 C→T mutation detected. Person is at risk of Beta-Thalassemia**

## -Familial Hypercholesterolemia:



Disease Detection

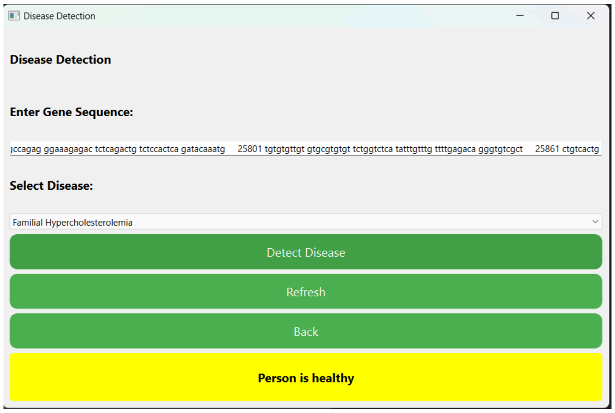
**Disease Detection**

**Enter Gene Sequence:** :cagag ggaaagagac tctcagactg tctccactca gatacaaatg 25801 tgtgtgttgt gtgcgtgtgt tctggtctca tatttgtttg ttttgagaca gggtgtcgct 25861 ctgtcactga

**Select Disease:** Familial Hypercholesterolemia

**Intronic mutation detected. Person is at risk of Familial Hypercholesterolemia**

## -Healthy person who doesn't have Familial Hypercholesterolemia mutation.



Disease Detection

**Disease Detection**

**Enter Gene Sequence:** jccagag ggaaagagac tctcagactg tctccactca gatacaaatg 25801 tgtgtgttgt gtgcgtgtgt tctggtctca tatttgtttg ttttgagaca gggtgtcgct 25861 ctgtcactg

**Select Disease:** Familial Hypercholesterolemia

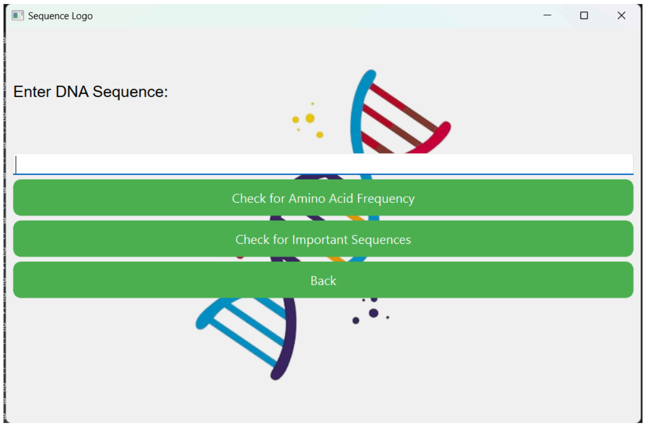
**Person is healthy**

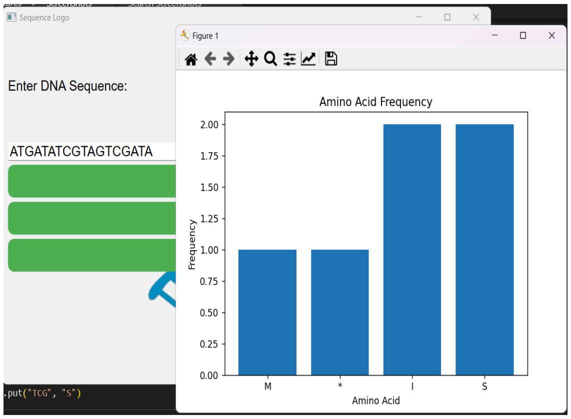
Using this tool can be very useful in the healthcare sector:

1. Early Detection of Genetic Disorders: By identifying mutations in gene sequences, healthcare professionals can diagnose genetic disorders at an early stage, even before symptoms manifest.
2. Research and Drug Development: Identifying mutations associated with specific diseases provides valuable insights into disease mechanisms and potential therapeutic targets.
3. Screening Programs: Disease detection tools can be integrated into screening programs aimed at identifying individuals at risk for certain genetic disorders.

## d) Sequence Logo

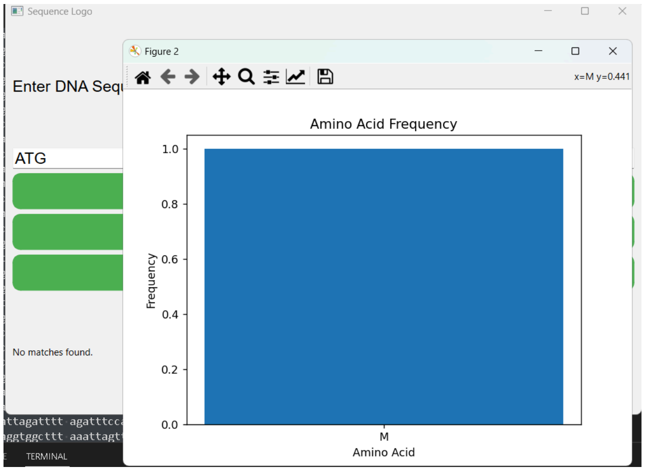
These snapshots show how the tool identifies codons in a sequence which code for a specific protein and plots a histogram on the frequency of the amino acids generated.





ATGATATCGTAGTCGATA

```
.put("TCG", "S")
```



ATG

No matches found.

```
ttagatttt agatttcc
ggtggcttt aaattagt
```

This feature has a lot of benefits:

1. Visualization of Sequence Conservation: Sequence logos visually represent the conservation of nucleotides or amino acids at each position within a sequence alignment. This allows researchers to quickly identify conserved regions, which are often functionally important, such as active sites in proteins or regulatory elements in DNA.
2. Prediction of Functional Elements: Sequence logos can aid in the prediction of functional elements within DNA or protein sequences. For example, they can help identify transcription factor binding sites, splicing signals, or protein domains by highlighting conserved sequence patterns associated with these elements.
3. Design of Experiments and Mutagenesis Studies: Sequence logos can guide the design of experiments, such as mutagenesis studies, by highlighting conserved residues or regions that are likely to be functionally important.

## 4. Challenges

We had originally decided on using a skiplist due to imagining the string sequence as a linked list form first, but that did not work out because this string cannot be considered as sorted, which is imperative in the skip list. So we decided to shift to a hash map technique. Moreover, it was very difficult to get genuine data from databases for the string sequence because, for instance, one file containing such a string amounted to more than 3GB of plain text. So we had to use smaller sections of such sequences, which were still thousands of letters long.

## 5. Work Division

Yashfeen: Did research about 5 diseases and their mutations, Found a website (NCBI) and learned it to extract gene sequence for our disease detection function, Wrote code for 4 diseases ( 2 on my own, and 2 with Fatima), and Integrated the final code. 25%
Fatima: Worked on the research of 5 diseases and compiled them in a document. Implemented 4 diseases and made the final presentation. 25%
Zuhair: Worked on the algorithm for integrating hash map in the functions, also created the sequence logo function. 25%

Rubab: Designed the UI, integrated different features of the code, and implemented the disease detection for 2 diseases in the disease detection window. 25%

## 6. Conclusion

Overall, we enjoyed working on this project as it provided us with a lot of ground to discover and delve deep into the world of bioinformatics. Despite the challenges, we were able to fulfill our goals for this project successfully as it empowers users with robust tools to explore genetic data effectively, catering to both novices and experts in the field. While our project right now is very small scale, we can see its potential in the healthcare sector as it has room for further improvements and expansions to address emerging needs in genetic analysis.

## 7. References

https://pubmed.ncbi.nlm.nih.gov/8518184/

https://www.ncbi.nlm.nih.gov/nuccore/NC_000019.10?report=genbank&from=11089463&to=11133820

https://rarediseases.org/rare-diseases/thalassemia-major/#:~:text=Beta%20thalassemia%20is%20caused%20by,mutations%20in%20both%20HBB%20genes.