

# Technical Report: DNA Sequence Analysis & Mutation Detection Tool

---

## 1. Project Overview

This software tool was developed to simulate bioinformatics analysis workflows used in research laboratories. It provides a clean, intuitive interface for analyzing DNA sequences, detecting mutations, and visualizing protein structures.

## 2. Algorithms and Implementation

### 2.1 Sequence Validation and Statistics

- **Validation:** Implemented using regular expressions (`/^ATCG]+$/`) to ensure only standard DNA nucleotides are processed.
- **Statistics:** Direct counting of nucleotides and calculation of GC/AT percentages based on the ratio of matching characters to total sequence length.

### 2.2 Open Reading Frame (ORF) Detection

- The algorithm scans the DNA sequence in three forward reading frames.
- It identifies an ORF starting with the initiation codon (**ATG**) and terminating at the first in-frame stop codon (**TAA**, **TAG**, or **TGA**).
- A minimum threshold of 30 base pairs is used to filter out non-significant sequences.

### 2.3 Sequence Alignment (Mutation Detection)

- **Global Alignment:** An implementation of the **Needleman-Wunsch algorithm** (using dynamic programming) is used to align two sequences. This allows for the detection of substitutions, insertions, and deletions.
- **Mutation Classification:**
  - **Transitions:** Purine-to-Purine (A↔G) or Pyrimidine-to-Pyrimidine (C↔T).
  - **Transversions:** Purine-to-Pyrimidine or vice versa.
  - **Impact Prediction:** By translating the codons affected by substitutions, the tool determines if the mutation is **Silent** (no AA change), **Missense** (AA change), or **Nonsense** (premature stop). Insertions and deletions are automatically flagged as **Frameshifts**.

### 2.4 DNA to Protein Translation

- A mapping dictionary (Codon Table) is used to translate triplets of nucleotides into their corresponding amino acids.

## 3. Design Choices

### 3.1 Architecture

- **Frontend:** Built with **React** and **Tailwind CSS** for a responsive, modern scientific dashboard.

- **Backend: Node.js/Express** handles the heavy computational bioinformatics logic, separating processing from the UI.
- **Storage:** An in-memory storage system was selected for the portable version to ensure zero-dependency local execution, while the schema is ready for PostgreSQL scaling.

### 3.2 3D Visualization

- Integrated with **RCSB PDB** and **AlphaFold** viewers via iframe-based embedding. This allows users to visualize protein structures derived from their analyzed sequences without needing local molecular modeling software.

## 4. User Interface (UI/UX)

- **Biological Color Coding:** Standard colors for nucleotides (A: Green, T: Red, C: Blue, G: Yellow) are used in alignment views to aid rapid visual inspection.
- **Monospaced Typography:** Essential for bioinformatics, ensuring that sequence alignments remain perfectly vertically aligned.

## 5. Conclusion

The application successfully integrates standard bioinformatics workflows into a user-friendly package. The modular design allows for future expansion, such as adding reverse-strand ORF detection or BLAST integration.