

Mini-Project: DNA Sequence Analysis & Mutation Detection

Deadline: 25 January 2026

Project Objective

This mini-project aims to help students apply the theoretical and practical knowledge acquired during TP (Travaux Pratiques) to a real-world bioinformatics application used in biological research laboratories. Students will design and develop a software tool that simulates real bioinformatics analysis workflows.

Project Context

Students will build a clean, intuitive, and user-friendly graphical interface capable of analyzing DNA sequences provided in FASTA format. The application must support both single-sequence analysis and two-sequence comparison for mutation detection and biological interpretation.

Input Format

The application must accept DNA sequences in FASTA format. Both direct text input and file upload should be supported. The application must validate the format and content before analysis.

PART 1 — Single DNA Sequence Analysis

- 1 Sequence validation: only A, T, C, G allowed.
- 2 Basic statistics: sequence length, nucleotide counts, GC% and AT%.
- 3 Reverse-complement sequence generation.
- 4 Codon analysis: splitting into triplets, detection of start codon (ATG) and stop codons (TAA, TAG, TGA).
- 5 ORF detection and counting.
- 6 Protein extraction from detected ORFs.
- 7 DNA to protein translation.

PART 2 — Two DNA Sequence Comparison

- 1 Sequence length comparison and validation.
- 2 Detection of substitutions, insertions, and deletions.
- 3 Classification of mutations: transition, transversion, insertion, deletion.
- 4 Mutation count and mutation rate calculation.
- 5 ORF comparison before and after mutation.
- 6 Protein sequence comparison before and after mutation.

- 7 Protein impact analysis: silent, missense, nonsense, frameshift mutations.

PART 3 — Visualization

- 1 Sequence alignment view using symbols (| for match, * for mutation).
- 2 Clear mutation highlighting.
- 3 Protein sequence comparison visualization.

3D Protein Visualization (Advanced Feature)

The application should display a 3D protein structure generated from the translated amino acid sequence. Students may integrate external databases and viewers such as AlphaFold DB, RCSB Protein Data Bank, NGL Viewer, Mol*, or any other suitable molecular visualization tool. The visualization should allow interactive exploration of protein structure and mutation effects.

UI / UX Requirements

- 1 Clean, minimal, and intuitive interface.
- 2 Separate tabs or sections for single analysis and comparison.
- 3 Clear error messages and validation feedback.
- 4 Readable fonts and consistent biological color coding.

Deliverables

- 1 Working application with graphical user interface.
- 2 Well-structured and commented source code.
- 3 Example FASTA test files.
- 4 Short technical report explaining algorithms and design choices.

Educational Outcomes

By completing this mini-project, students will strengthen their understanding of bioinformatics algorithms, gain experience in real biological data analysis, and learn how computational tools are applied in biological and genetic research laboratories.