# Virus Ancestry Detection

# Team Members

Hamsa Saber 1210359

Alia Tarek 4220121

Salsabil Mostafa 1210171

Youssef Affify 1200883

# PROBLEM STATEMENT

**Challenges:**

- Genome sequences are long and complex, making manual analysis impossible.
- Comparing multiple sequences is computationally expensive

**Need for Efficiency:**

- Bioinformatics requires optimized algorithms for accuracy and speed.
- Divide-and-conquer algorithms simplify the comparison process.

# PROJECT OVERVIEW

**Goal**:

Identify the family of an unknown virus using DNA sequencing.

- Compare the unknown sequence with a database of known virus genomes.
- Determine the closest match and identify its family.

**Relevance:**

- Helps track virus evolution and origins.
- Supports research and response to pandemics.

# Workflow Overview

**Input:**

- An unknown DNA sequence from a .fasta file.
- A database of known DNA sequences from viruses.

**Processing Steps:**

1. Used a divide-and-conquer strategy to break down DNA sequences into smaller chunks for manageable comparison.
2. Leveraged Biopython's alignment tools, which use dynamic programming, to compare each chunk against database sequences.
3. Scored alignments based on matches, mismatches, and gaps to determine the closest viral match.

**Output:**

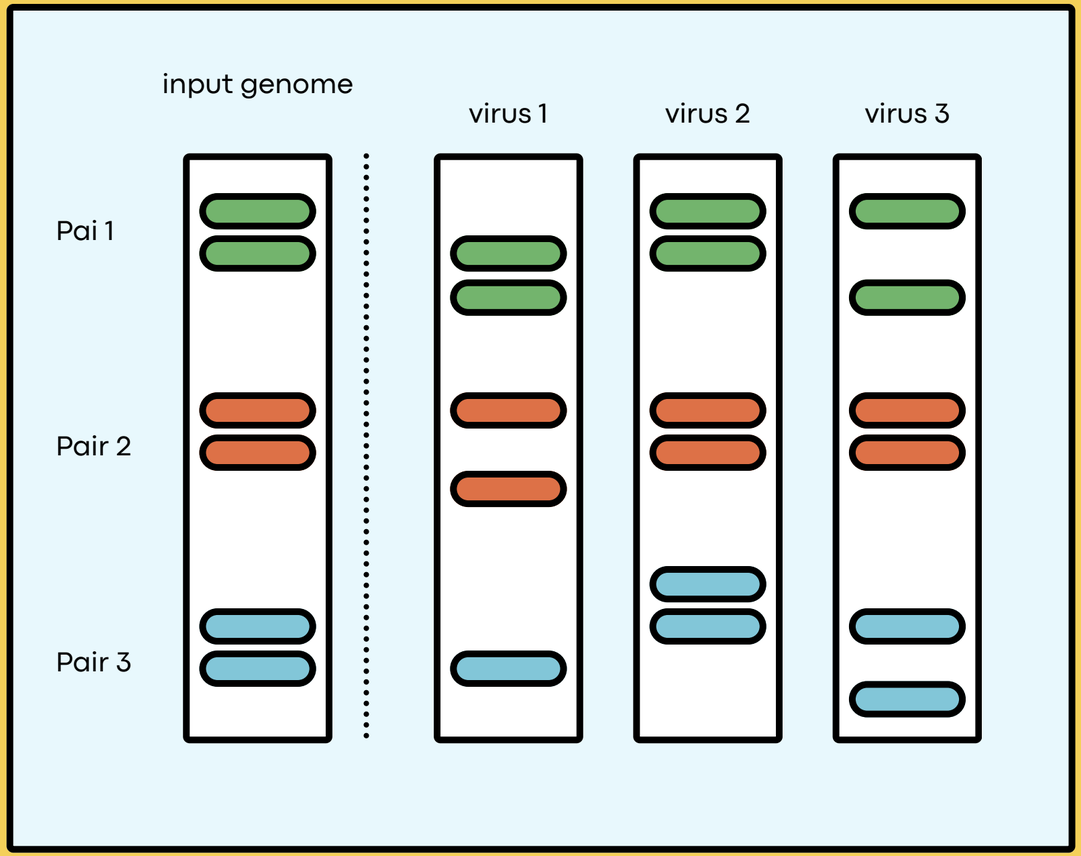
- Identify the closest match and its virus family.

# DIVIDE AND CONQUER ALGORITHM

## Divide:

- Split both the unknown DNA and database sequences into smaller chunks.
- Example: Split into halves recursively until the length reaches a minimum threshold.

# DIVIDE AND CONQUER ALGORITHM




## Conquer

- Compare chunks using the scoring system:
- Match: +2 points.
- Mismatch: -2 points.
- Gap penalties: Open gap = -5
- Extend gap = -1.

## Combine

- Merge the results of all chunk comparisons.
- Sum scores to get the overall similarity score for each sequence.

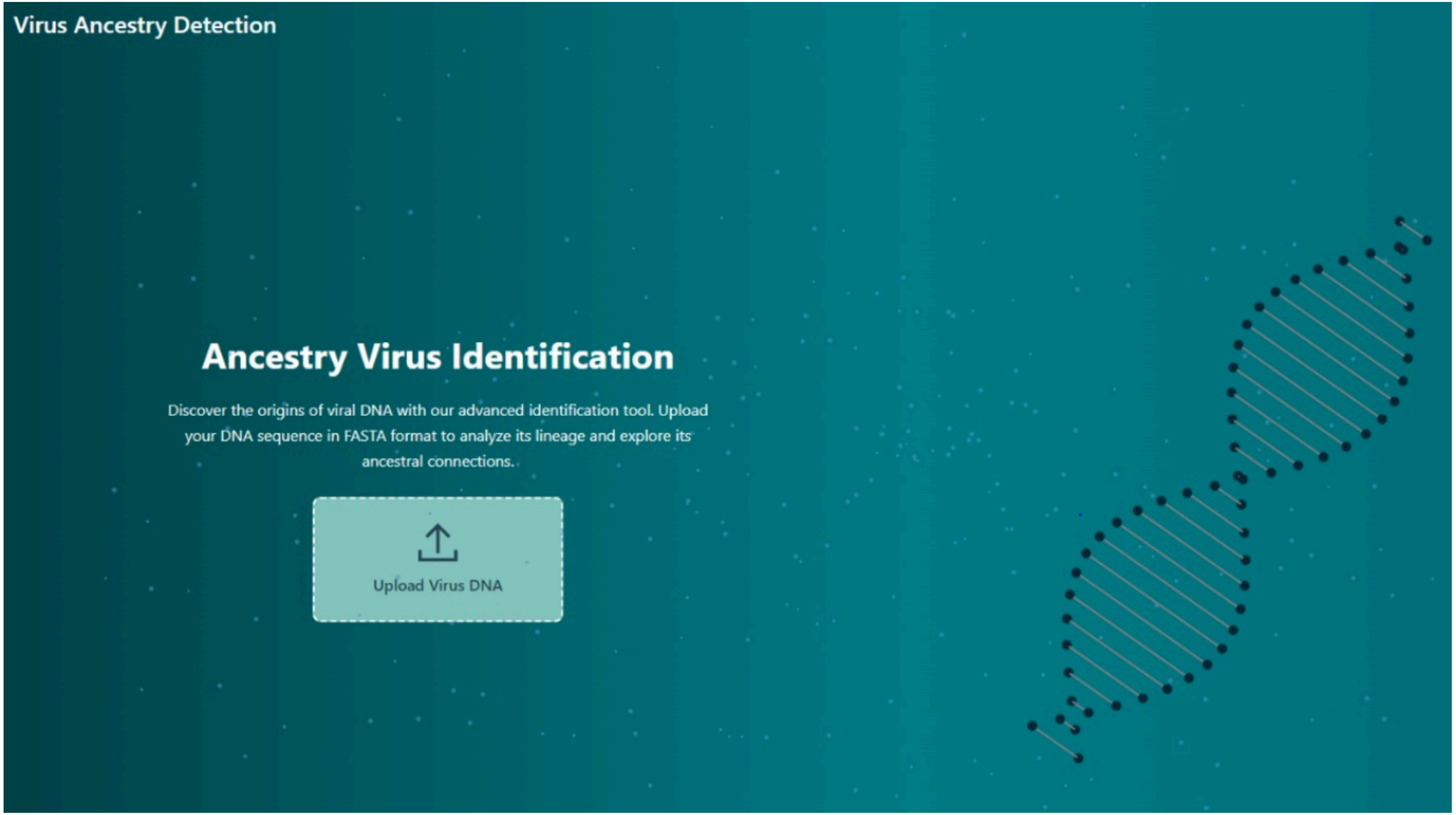
# PSUEDOCODE

1. **Loading DNA Sequences:**
   - Start by reading DNA sequences from all `.fasta` files in the database folder.
   - If a file matches the name of the unknown virus, treat it as the *target sequence*.
   - For all other sequences, store them in a database for comparison.
   - Skip very short sequences since they might not be useful for analysis.
2. **Aligning Two Sequences:**
   - Use a scoring system to align two DNA sequences:
     - Add 2 points for every matching base.
     - Subtract 2 points for mismatches.
     - Apply penalties for gaps: -5 to open a gap and -1 for extending it.
   - Normalize the score by dividing it by the length of the alignment to account for different sequence sizes.
   - Return the score along with the aligned versions of the sequences.

# PSUEDOCODE

3. **Divide and Conquer:**
   - To handle large sequences efficiently:
     - Split both sequences into two halves.
     - Align the left halves and right halves separately using the same process.
     - Combine the results to get the total score and aligned sequences.
   - If the sequences are short enough (less than 1800 bases), skip the splitting and align them directly.
4. **Finding the Closest Match:**
   - Compare the target sequence with every sequence in the database:
     - Align them using the **local alignment** method.
     - Calculate the alignment score for each match.
   - Normalize the score based on the length of the aligned sequences.
   - Calculate the average score of all alignments and set a threshold.
   - Identify the best match that has a score above the threshold.

# UI SNAPSHOTS



# UI SNAPSHOTS

# COMPLEXITY

**Aligning Sequences:**
O(n×m)
L1: length of small chunk of sequence 1
L2: length of small chunk of sequence 2

**Dividing the sequences:**
O(log(N))
N: length of longer seq

## Final complexity

O(L1*L2*log(N))

- The total work done across all levels of recursion is proportional to L1×L2 at each level.
- There are O(log(N)) levels in the recursion, since the sequences are halved at each step.

# CONCLUSION

This project showcases how efficient algorithms like divide-and-conquer can accurately identify virus families from genome sequences, aiding in research and pandemic response. It highlights the power of bioinformatics in transforming genetic data into actionable insights.

THANK YOU!!
Any Questions?