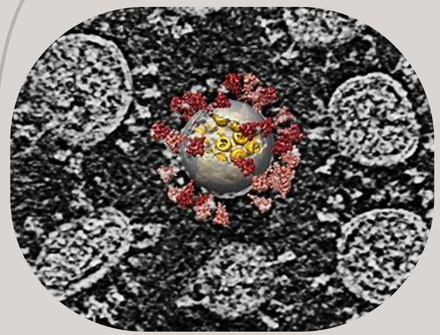
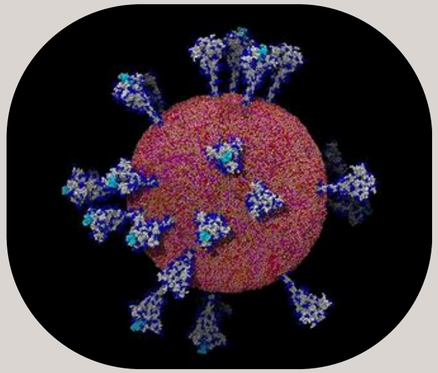


DEVELOPING MUTATION DETECTION ALGORITHMS FOR SARS-COV-2 COVID 19 GENOMIC SEQUENCES

Group A9

1 Abhishek Karthik J-CB.SC.U4AIE23010
2 Keerthivasan S V-CB.SC.U4AIE23037
3 Mothishwaran M P-CB.SC.U4AIE23041
4 Mopuru Sai Bavesh Reddy-CB.SC.U4AIE23044



INTRODUCTION

- Viruses, including SARS-CoV-2, mutate over time, affecting transmissibility, vaccine effectiveness, and public health policies.
- We use Multiple Sequence Alignment (MSA) techniques like MAFFT and PRANK to systematically compare genetic sequences.
- The study is based on genomic data from different states (California, Texas, New York, etc.).

Key objectives:

- Identify mutations in different viral strains to track their evolution.
- Compare genomic sequences between states to detect regional mutation patterns.
- How we achieve this:MSA techniques align large-scale genomic data to detect variations.
- Phylogenetic analysis helps understand the virus's evolutionary relationships.

OBJECTIVE

- Develop MSA approaches (MAFFT and PRANK) from scratch to analyze SARS-CoV-2 genomic sequences
- Generate consensus viral sequences from California, New York, and Texas samples
- Compare sequences with Wuhan reference strain to track evolutionary divergence
- Identify mutation hotspots and regional variation patterns
- Construct phylogenetic trees to visualize evolutionary relationships
- Implement efficient algorithms for sequence alignment, mutation detection, and phylogenetic analysis

PROJECT SCOPE

- Implement Multiple Sequence Alignment (MSA) algorithms from scratch using both MAFFT and PRANK methodologies.
- Align regional sequences with the Wuhan reference strain to establish a baseline for mutation tracking.
- Identify mutation hotspots across different states and analyze their biological significance.
- Construct phylogenetic trees to visualize the evolutionary distance from the original Wuhan strain.
- Compare alignments between states to assess regional differences in viral evolution.
- Analyze clade structures to determine how far each region's viral population has diverged.
- Provide insights that can contribute to genomic surveillance and epidemiological response strategies.

DATASET

Dataset Overview:

- Content: Complete SARS-CoV-2 genomic sequences (~30,000 nucleotides each)
- Geographical Focus: California, Texas, and New York
- Reference: Wuhan-Hu-1 strain (NC_045512.2) included for comparison

Dataset Features:

- Genomic Metadata: Accession ID, BioProject/BioSample IDs, Collection Date/Location
- Sequence Data: Complete nucleotide sequences in FASTA format
- Variant Classification: Pangolin lineage assignments
- Quality Metrics: Sequence completeness, coverage depth

DATASET

Data Processing:

- Filtering: Removed sequences <29,000 bp or with >1% ambiguous bases
- Standardization: Converted all sequences to uppercase and standard IUPAC notation
- Cleaning: Corrected common sequencing artifacts and trimmed low-quality regions
- Organization: Structured into state-specific FASTA files with consistent formatting

Methodology

Data Collection Process

- Extracted SARS-CoV-2 genomic sequences from California, Texas, and New Yor
- Implemented robust CSV parsing system that:
  - Automatically detected CSV delimiters
  - Used fuzzy matching for column identification (Sequence ID, Nucleotide Sequence, location, etc.)
  - Filtered by geographic location
  - Handled data consistency issues
- Output: Three structured CSV files with genomic sequences and metadata

Sequence Preprocessing

- Sequence cleaning:
  - Converted sequences to uppercase
  - Removed ambiguous characters (keeping only A, T, C, G)
  - Filtered out short sequences (<29,000 bp)
- FASTA format conversion:
  - Formatted with proper headers
  - Wrapped sequence lines at 80 characters
  - Saved as separate FASTA files for each state

Methodology: MAFFT Algorithm Implementation

Numerical Encoding and FFT Application

- Assigned numerical values to nucleotides: A=1, C=2, G=3, T=4
- Applied Fast Fourier Transform to convert numerical sequences to frequency domain
- Calculated magnitude spectrum of FFT results
- Generated plots of FFT magnitude spectra for visualization

Distance Matrix Computation

- Calculated Euclidean distance between FFT magnitude spectra for each sequence pair
- Assembled distances into a symmetric matrix
- Smaller distances indicated greater sequence similarity

Guide Tree Construction

- Applied UPGMA algorithm for hierarchical clustering based on pairwise distances
- Iteratively merged closest pairs of clusters, recalculating distances at each step
- Created internal nodes representing ancestral relationships between merged clusters

Methodology: MAFFT Algorithm Implementation

Progressive Alignment

- Started with most closely related sequences based on guide tree
- Performed pairwise alignments using dynamic programming
- Created profiles representing aligned groups
- Aligned profiles following guide tree order
- Applied appropriate gap penalties

Iterative Refinement

- Systematically removed one sequence at a time
- Realigned each removed sequence to remaining profile
- Evaluated alignment score improvement
- Repeated until no further improvement or maximum iterations reached

Methodology: PRANK Algorithm Implementation

Pairwise Distance Calculation

- Calculated Hamming distance between sequence pairs (counting positions with different nucleotides)
- Normalized by dividing by shorter sequence length
- Constructed symmetric distance matrix

UPGMA Tree Construction

- Started with each sequence as its own cluster
- Organized pairwise distances in priority queue
- Iteratively merged clusters with smallest distance
- Recalculated distances after each merger
- Maintained tree structure recording merge history

Methodology: PRANK Algorithm Implementation

Pairwise Alignment with Indel Distinction

- Implemented Needleman-Wunsch variant with separate gap penalties
- Maintained traceback matrix recording cell score origins
- Annotated gaps as insertions or deletions
- Reconstructed aligned sequences following traceback path

Progressive Phylogeny-Aware Alignment

- Represented partially aligned sequences as profiles
- Reconstructed ancestral sequences at internal nodes
- Aligned profiles following guide tree topology
- Maintained consistency in gap treatments

Methodology: Global Alignment Implementation

Initialization

- Created scoring matrix (m+1) × (n+1)
- Initialized first row and column with cumulative gap penalties
- Used scoring scheme: +2 for matches, -1 for mismatches, -2 for gaps

Matrix Filling

- Calculated score for each cell as maximum of:
  - Diagonal: Previous diagonal cell + match/mismatch score
  - Horizontal: Previous horizontal cell + gap penalty
  - Vertical: Previous vertical cell + gap penalty
- Propagated optimal alignment choices through matrix

Methodology: Global Alignment Implementation

Traceback

- Started at bottom-right cell
- Determined which move led to each cell's score
- Constructed alignment:
  - Diagonal: Aligned corresponding characters
  - Horizontal: Gap in first sequence
  - Vertical: Gap in second sequence
- Handled leading gaps as needed

Identity Calculation

- Counted positions with identical characters
- Calculated percentage: (matches ÷ total alignment length) × 100

Methodology: Phylogenetic Analysis

1. Data Collection
   - Retrieved consensus nucleotide sequences from three U.S. regions: New York, Texas, and California in FASTA format.
   - Obtained the FASTA file of a known deadly virus for comparison.
2. Preprocessing and Cleaning
   - Removed whitespaces and newline characters from all sequences.
   - Standardized all sequences to uppercase for consistency.
3. Sequence Padding
   - Aligned all sequences to the maximum length found across regions using gap characters (-) to ensure proper comparison.
4. Deadly Virus Alignment
   - Read the deadly virus sequence and padded or trimmed it to match the length of the regional sequences.

Methodology: Phylogenetic Analysis

5.Multiple Sequence Alignment

- Combined all padded sequences (regional + deadly virus) into a single dataset for alignment.

6.Distance Calculation

- Used identity-based distance calculation to compute the pairwise distances between sequences.

7.Phylogenetic Tree Construction

- Built a Neighbor-Joining (NJ) tree using the distance matrix.

8.Result Visualization and Interpretation

- Identified the closest region to the deadly virus based on the smallest average distance.

Methodology: Consensus Sequence Comparison

Consensus Sequence Generation

- Analyzed each position in aligned sequences
- Applied majority rule (>50% frequency)
- Inserted gaps where no nucleotide had majority

Fast Identity Comparison

- Excluded positions with gaps in either sequence
- Counted matching nucleotide positions
- Calculated percentage of matching positions

Identity Matrix Construction

- Performed all-vs-all comparison of consensus sequences
- Organized percentages in symmetric matrix
- Formatted for clear visualization

PRANK

What is PRANK?

PRANK (Probabilistic Alignment Kit) is a phylogeny-aware multiple sequence alignment algorithm.

Unlike traditional methods, PRANK distinguishes between insertions and deletions during the alignment process, which helps in more accurately representing the evolutionary history of sequences.

This approach is particularly valuable for analyzing rapidly evolving viruses like SARS-CoV-2, where accurately identifying insertion-deletion events is crucial.

Key Features of PRANK:

1. **Phylogeny-aware: PRANK uses evolutionary information to guide the alignment process**
2. **Accurate indel handling: It distinguishes between insertions and deletions, reducing alignment errors**
3. **Evolutionary modeling: Incorporates evolutionary models to better represent sequence changes over time**
4. **Reduced alignment bias: Avoids systematic errors common in progressive alignment methods**

HOW IS PRANK ??

WHAT ARE THE STEPS INVOLVED IN THIS?

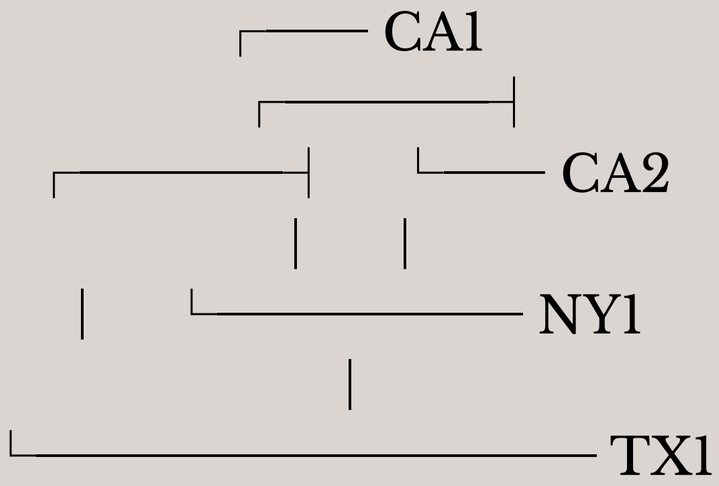
- CONSTRUCTING A GUIDE TREE USING EVOLUTIONARY DISTANCES
- DISTINGUISHING INSERTIONS FROM DELETIONS DURING ALIGNMENT
- PERFORMING PHYLOGENY-AWARE PROGRESSIVE ALIGNMENT USING
- POSTERIOR PROBABILITY CALCULATIONS FOR ALIGNMENT DECISIONS

Step 1: Construct a Guide Tree ($O(N^2)$)

How it Works:

- PRANK first estimates evolutionary distances between sequences
- These distances are used to build a phylogenetic guide tree
- The tree represents the evolutionary relationships between sequences

Example: For sequences from different regions (CA1, CA2, NY1, TX1):



The tree determines not just the order of alignment, but also helps track evolutionary events

Step 2: Distinguish Insertions and Deletions ($O(N^2)$)

How it Works:

- PRANK tracks the history of indels as alignment progresses
- New gaps are labeled as either insertions or deletions based on the guide tree
- This prevents the same gap from being penalized multiple times

Example:

When aligning:

Seq1: ACGTACGT
Seq2: AC--ACGT

PRANK identifies the gap as either:
An insertion in Seq1, or
A deletion in Seq2 This decision is made based on the evolutionary context from the guide tree.

Step 3: Phylogeny-Aware Progressive Alignment ($O(N^3)$)

How it Works:

- Sequences are aligned following the guide tree topology
- At each node, PRANK reconstructs ancestral sequences
- The algorithm considers the evolutionary changes that might have occurred

Example:

For three sequences:

S1: ACGTACGT
S2: ACGAACGT
S3: ACGACGT

PRANK would:

1. Infer the ancestor of S1 and S2
2. Align S3 with this ancestor, considering evolutionary events
3. Result: S1: ACGTACGT S2: ACGAACGT S3: ACG-ACGT

Step 4: Using Posterior Probabilities for Alignment Decisions ($O(N^2)$)

How it Works:

- PRANK calculates the probability of different evolutionary scenarios
- Alignment decisions are made based on these probabilities
- This approach minimizes systematic errors in gap placement

Example: When deciding whether to place a gap:

- PRANK evaluates: P(insertion) vs. P(deletion) vs. P(substitution)
- Chooses the scenario with the highest probability
- This results in more biologically accurate alignments, especially in regions with multiple indels

MAFFT

What is MAFFT?

MAFFT (Multiple Alignment using Fast Fourier Transform) is a widely-used algorithm for multiple sequence alignment. It aligns nucleotide or protein sequences to identify regions of similarity that may indicate functional, structural, or evolutionary relationships between the sequences. MAFFT uses Fast Fourier Transform (FFT) to accelerate the alignment process, making it efficient even for large datasets.

Key Features of MAFFT:

1. **Speed:** MAFFT is faster than many traditional alignment methods due to the use of FFT.
2. **Accuracy:** It provides high-quality alignments by considering both local and global sequence similarities.
3. **Versatility:** MAFFT can handle a wide range of sequence lengths and numbers of sequences.

We’re implementing a simplified version of multiple sequence alignment, inspired by the concepts used in MAFFT from scratch.

HOW IS MAFFT ??

WHAT ARE THE STEPS INVOLVED IN THIS ??

- FINDING THE SIMILARITY MATRIX USING FFT
- FROM THE SIMILARITY MATRIX FIND THE GUIDE TREE
- FROM THE GUIDE TREE FIND THE PROGRESSIVE ALINGMENT
- AFTER WE SHOULD ITERATIVELY REFINEMENT

Step 1: Compute the Similarity Matrix using FFT (O(n log n))
How it Works:

- Instead of directly comparing sequences, MAFFT uses Fast Fourier Transform (FFT) to convert sequences into frequency representations.

Example:
Consider two sequences:

S1 = "ACGTACGT"S2 = "AGGTACGT"

- Convert each sequence into a numerical signal (e.g., assigning A=1, C=2, G=3, T=4).
- Apply FFT to transform these sequences into frequency components.
- Compute similarity scores by comparing the frequency spectra.

Result:
\

	s1	**s2**	**s3**
s1	0.8	0.9	0.7
s2			
s3			

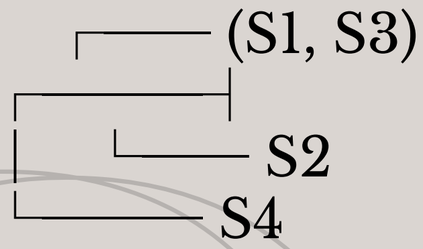
Step 2: Construct the Guide Tree ($O(N^2)$)

How it Works:

- The similarity matrix is used to build a guide tree
- Sequences with the highest similarity are clustered first.

Example :

Guide Tree Output:



The tree determines the order in which sequences will be aligned.

Step 3: Progressive Alignment ($O(N^2)$)
How it Works:

- Sequences are aligned step by step, following the order in the guide tree.

Step 4: Iterative Refinement ($O(N^3)$)
How it Works:

- The initial alignment may not be optimal.
- MAFFT removes one sequence at a time, realigns it, and checks if the alignment score improves.
- This adjusts gaps dynamically to improve accuracy.

example :

```
ACGT-
ACGG-
AGGT-
TCGT-
```

- Remove S4 and re-align:

  ACGT-
  ACGG-
  AGGT-

- Reinsert S4 with a gap for better alignment:
- diff

- ACGT-
- ACGG-
- AGGT-
- -TCGT

Results

- **Mafft:** We used MAFFT to generate a multiple sequence alignment, which is a crucial first step in comparing the genetic sequences.
- **Prank:** Another multiple sequence alignment was created using PRANK, allowing us to compare results from different alignment algorithms.
- **Building Phylogenetic Trees from Aligned Sequences:** From these alignments, we constructed phylogenetic trees to visualize the evolutionary relationships between the virus samples.
- **Phylogenetic Tree and Comparison to the Reference Sequence:** This tree visually shows how the regional samples relate to the reference sequence, indicating their evolutionary distance.
- **Global Alignment of Consensus Sequences**: By aligning the representative consensus sequences from each region, we can see the overall genetic similarity and key differences between them.

Challenges

- Challenges:
- Processing large genomic datasets (30,000+ nucleotides per sequence) required significant computational resources
- Distinguishing true mutations from sequencing errors required careful filtering and validation
- Algorithm optimization was necessary to handle the scale of data without excessive runtime
- Balancing sensitivity (detecting all mutations) with specificity (avoiding false positives)
- Regional datasets varied in quality and completeness, requiring robust preprocessing
- PRANK algorithm's phylogeny-aware approach was computationally intensive for large datasets

Literature review

Study	Methodology	Key Findings
Tracking Mutational Semantics of SARS-CoV-2 Genomes (2022)	Employed natural language processing algorithms to analyze SARS-CoV-2 genome mutations.	Demonstrated that NLP-based approaches can effectively process genomic data, revealing characteristics and evolutionary patterns of the virus.
Rapid Detection of Predominant SARS-CoV-2 Variants Using High-Resolution Melting Analysis (2023)	Developed a multiplex high-resolution melting (HRM) method for SARS-CoV-2 detection and mutation site identification.	Established a one-step multiplex HRM assay capable of detecting SARS-CoV-2 and identifying specific mutation sites, facilitating rapid variant analysis.

Conclusion

- Successfully implemented MAFFT and PRANK algorithms for SARS-CoV-2 sequence analysis
- Identified distinct evolutionary patterns across California, Texas, and New York
- Found high similarity between California and Texas (90.08%) using MAFFT
- Observed lower similarity involving New York sequences (27-31%)
- Different results from PRANK implementation highlighted importance of algorithm selection
- Constructed phylogenetic trees visualizing evolutionary relationships
- Demonstrated value of implementing MSA algorithms from scratch
- Methodologies can be extended to other viral genomic analyses

Thank You