# Tree-Based Phylogenetic Analysis: A Tool for Understanding Pathogen Dynamics in Indonesia

Grace Evelyn Simon - 13523087[1,2]
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
[1]13523087@std.stei.itb.ac.id, [2]graceevelynsimon@gmail.com

***Abstract*—Phylogenetic analysis has become an indispensable tool for understanding pathogen dynamics, especially in the context of emerging infectious diseases like COVID-19. This study utilizes genome sequences from various pathogens, including SARS, MERS, Ebola, HIV, and Malariae, to investigate their evolutionary relationships and genomic similarities through trinucleotide composition and pairwise sequence alignment. By employing distance-based methods, a phylogenetic tree is constructed to visualize these relationships. This approach provides insights into the evolutionary trajectories and potential zoonotic origins of pathogens, emphasizing its importance for public health strategies in Indonesia. The study demonstrates how phylogenetic tools can be applied to assess genetic diversity and trace pathogen evolution, contributing to preparedness and response against emerging infectious diseases.**

***Keywords*—Genome, Pathogen Dynamic, Phylogenetic Tree, Tree.**

## I. INTRODUCTION

Indonesia, with its vast biodiversity and rapidly growing population, faces significant public health challenges related to infectious diseases. The emergence and re-emergence of pathogens, such as SARS-CoV-2, dengue, avian influenza, and tuberculosis, highlight the critical need for tools that can unravel their evolutionary patterns and transmission dynamics. Tree-based phylogenetic analysis has emerged as an indispensable method for studying these pathogens, enabling researchers to understand their origins, track mutations, and predict future outbreaks.

Phylogenetic trees, which depict evolutionary relationships among various organisms or genomes, are constructed using genetic data. These trees provide a visual and analytical framework for comparing the genomes of pathogens, identifying common ancestors, and tracking genetic changes over time. By applying tree-based methods, scientists can monitor the evolution of antimicrobial resistance and identify genetic markers associated with increased pathogenicity or transmissibility.

In Indonesian context, phylogenetic analysis has far-reaching implications. As a biodiversity hotspot, Indonesia serves as a reservoir for many zoonotic pathogens due to its dense human-animal interactions. Moreover, its unique geography, spanning thousands of islands, poses logistical challenges for traditional epidemiological surveillance. Tree-based phylogenetics offers a scalable and data-driven approach to overcome these challenges by utilizing genomic data to trace outbreaks and predict pathogen behavior. For example, during the COVID-19 pandemic, phylogenetic studies were instrumental in tracing the origins of SARS-CoV-2 and understanding its mutations as it spread globally. Similarly, such analyses have been applied to avian influenza viruses in Indonesia to track their genetic evolution and assess pandemic potential. These insights are critical for informing public health policies, designing vaccines, and implementing targeted interventions.

This paper explores the applications of tree-based phylogenetic analysis in Indonesia, focusing on its role in monitoring pathogen dynamics, addressing public health challenges, and improving preparedness for future pandemics. By leveraging this powerful tool, Indonesia can enhance its ability to respond to infectious diseases and contribute to the global understanding of pathogen evolution.

## II. FUNDAMENTAL THEOREM

### A. *Graph*

Graphs are commonly used to represent discrete objects and the relationships between those objects. A graph $G$ is defined as $G = (V, E)$ where:

$V$ = A non-empty set of vertices (nodes) denoted as $V = \{v_1, v_2, \dots, v_n\}$ with the set $V$ must not be empty, meaning the graph cannot exist without any vertices.

$E =$ A set of edges that connect pairs of vertices denoted as $E = \{e_1, e_2, \dots, e_m\}$ with the set $E$ can be empty, meaning a graph can exist without any edges.

A pair of vertices in a graph can be connected by two different edges, and such a pair of edges is called multiple edges. Additionally, there are edges that start and end at the same vertex, which are referred to as loops. Based on the presence of multiple edges or loops, graphs can be categorized as follows:

1. Simple Graph: A graph that does not have multiple edges or loops.
2. Unsimple Graph: A graph that contains either multiple edges or loops. An unsimple graph that includes loops is referred to as a pseudo graph.

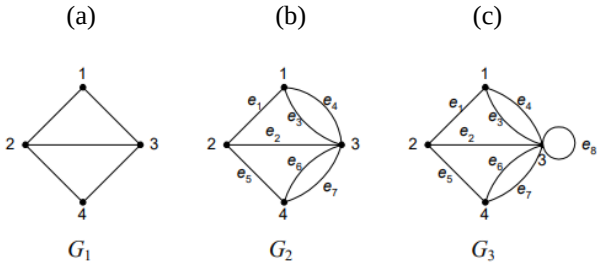


*Fig. 2.1 (a) Simple Graph (b) Unsimple Graph (c) Pseudo Graph*
*Source: [1]*

Based on the orientation of the edges, graphs are classified into two types:

1. Undirected Graph: A graph where the edges do not have any orientation or direction.
2. Directed Graph (or Digraph): A graph where each edge has a specific orientation or direction.

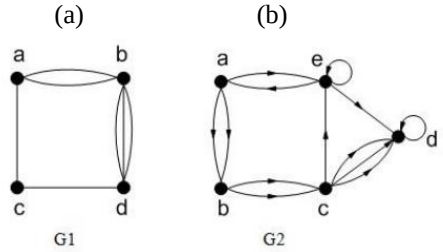


*Fig. 2.2 (a) Undirected Graph (b) Directed Graph*
*Source: [1]*

Here are some basic graph terminologies that are essential to understand:

1. Adjacency
   Two vertices in a graph are said to be adjacent if they are connected by at least one edge.

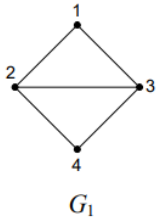


*Fig. 2.3 Adjacency Graph*
*Source: [1]*

2. Incidency
   An edge e is said to be incident to vertices $v_i$ and $v_j$ if it connects $v_i$ and $v_j$.

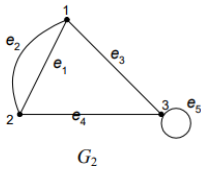


*Fig. 2.4 Incidency Graph*
*Source: [1]*

3. Isolated Vertex
   An isolated vertex is a vertex that has no edges incident to it.

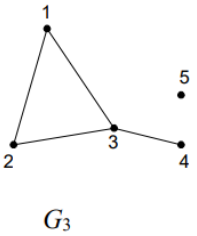


*Fig. 2.5 Isolated Vertex Graph*
*Source: [1]*

4. Null Graph
   A null graph is a graph whose edge set is empty.

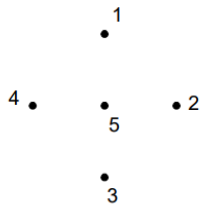


*Fig. 2.6 Null Graph*
*Source: [1]*

5. Degree
   The degree of a vertex is the number of edges that are incident to it.
6. Path
   A path of length n is a sequence of vertices and edges $v_0, e_1, v_1, e_2, v_2, \ldots, e_n, v_n$ that connects the initial vertex $v_0$ to the terminal vertex $v_n$. The path length n refers to the number of edges in the path.
7. Cycle (Circuit)
   A cycle or circuit is a path that starts and ends at the same vertex.
8. Connectivity
   Two vertices $v_1$ and $v_2$ are said to be connected if there exists a path that links $v_1$ to $v_2$. $G$ is connected graph if for every node $v_i$ and $v_j$ in $V$ there is path from $v_i$ to $v_j$. If none, $G$ is disconnected graph.

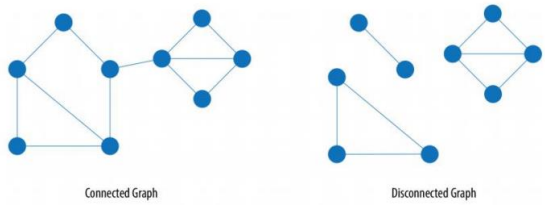


*Fig. 2.7 Connected amd Disconnected Graph*
*Source: [1]*

9. Subgraph and Complement of a Subgraph
   Let $G = (V, E)$ be a graph. A graph $G_1 = (V_1, E_1)$ is a subgraph of $G$ if $V_1 \subseteq V$ *and* $E_1 \subseteq E$. The complement of a subgraph $G_1$ relative to $G$ is $G_2 = (V_2, E_2)$ where $E_2 = E - E_1$ *and* $V_2$ is the set of vertices incident to the edges in $E_2$.

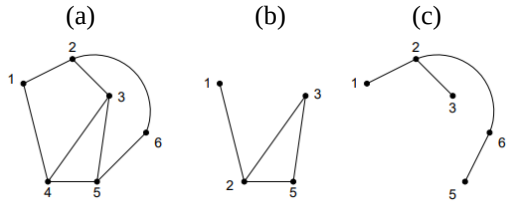


*Fig. 2.8 (a) Graph $G_1$ (b) An Upagraph (c) Complement of Upagraph (b)*
*Source: [1]*

10. Spanning Upagraph
    A subgraph $G_1 = (V_1, E_1)$ of $G = (V, E)$ is called a spanning subgraph if $V_1 = V$, meaning $G_1$ contains all the vertices of $G$.

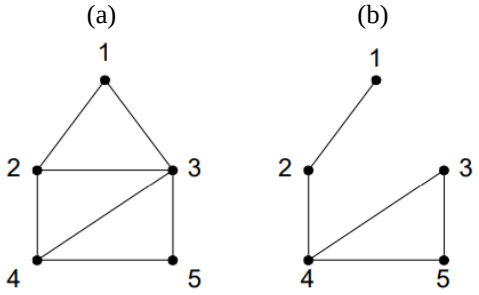


*Fig. 2.9 (a) Graph G (b) Spanning of Upagraph (a)*
*Source: [1]*

11. Cut-Set
    A cut-set of a connected graph $G$ is set of edges that, if removed, causes $G$ to become disconnected. A cut-set always divides $G$ into two components.
12. Weighted Graph
    A weighted graph is a graph where each edge is assigned a specific value or weight.

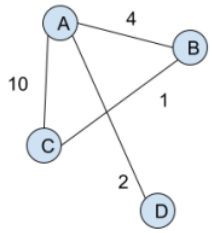


*Fig. 2.10 Weighted Graph*
*Source: [1]*

13. Complete Graph
    A complete graph is a graph in which every vertex is connected by an edge to every other vertex in the graph.

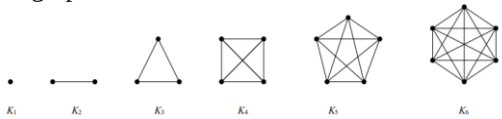

*Fig. 2.11 Complete Graph*
*Source: [1]*

14. Cycle Graph
    A cycle graph is a simple graph where every vertex has a degree of two.

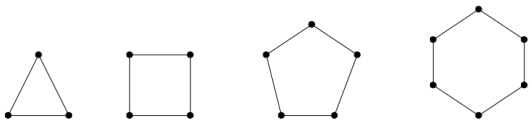

*Fig. 2.12 Cycle Graph*
*Source: [1]*

15. Regular Graph
    A graph is called a regular graph if all its vertices have the same degree.

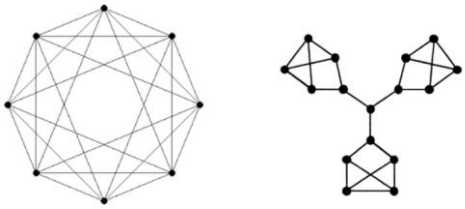

*Fig. 2.13 Regular Graph*
*Source: [1]*

16. Bipartite Graph
    A graph $G$ is called a bipartite graph if its vertex set can be divided into two disjoint subsets $V_1$ and $V_2$, such that every edge in $G$ connects a vertex in $V_1$ to a vertex in $V_2$. It is denoted as $G(V_1, V_2)$.

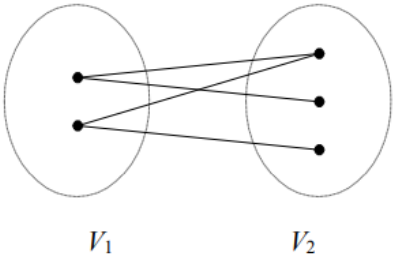


*Fig. 2.14 Bipartite Graph*
*Source: [1]*

*B. Tree*

A tree is an undirected graph that is:

1. Connected: All vertices are reachable from one another.
2. Contains no circuits (cycles): There are no closed loops within the graph.

Meanwhile, a forest is:

1. A collection of disjoint trees, or
2. A disconnected graph with no circuits. Each connected component of such a graph is a tree.

Let $G = (V, E)$ be a simple, undirected graph with n vertices and m edges. The following statements about $G$ are equivalent, and any one of them can serve as a definition of a tree:

1. $G$ is a tree.
2. Every pair of vertices in $G$ is connected by exactly one unique path.
3. $G$ is connected and has m = n - 1 edges.
4. $G$ contains no circuits and has m = n - 1 edges.
5. $G$ contains no circuits, and adding any one edge to $G$ creates exactly one circuit.
6. $G$ is connected, and every edge is a bridge (removing any edge would disconnect the graph).

A rooted tree is a tree where one vertex is designated

as the root, and all edges are directed outward from the root, forming a directed graph. Rooted trees are widely used in computer science, particularly in hierarchical data structures. There are some terminologies in rooted trees:

1. Child and Parent
   A vertex connected directly below another vertex is called a child, while the vertex above is the parent.
2. Path
   A path is a sequence of vertices connected by edges, starting from the root to a target vertex.
3. Siblings
   Vertices that share the same parent are called siblings.
4. Subtree
   A subtree is any tree rooted at one of the vertices of the original tree.

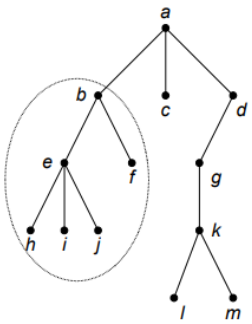


*Fig. 2.15 Subtree*
*Source: [2]*

5. Degree
   The degree of a vertex in a rooted tree refers to the number of children (or subtrees) it has.
6. Leaf
   A leaf is a vertex with no children (degree 0).
7. Internal Node
   An internal node is a vertex with at least one child.
8. Level
   The level of a vertex is the length of the path from the root to that vertex.

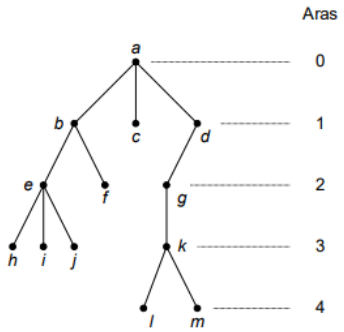


*Fig. 2.16 Tree Level*
*Source: [2]*

9. Height (Depth)
   The height (or depth) of a tree is the maximum level of any vertex in the tree.

## III. PROBLEM ANALYSIS

Infectious diseases such as COVID-19, SARS, and MERS have posed significant global health threats. Understanding their evolutionary origins and relationships is essential for tracing zoonotic sources and predicting future outbreaks. Indonesia, with its rich biodiversity, provides a unique opportunity to explore the origins of such pathogens due to the presence of natural reservoirs like bats and civets. Yet, despite Indonesia's role as a biodiversity hotspot, there is limited research comparing its endemic pathogens to global viral strains. This gap restricts our understanding of how these viruses evolve and adapt locally and their potential to contribute to future pandemics. Evolutionary relationships between pathogens are often difficult to determine without appropriate computational tools. Genomic mutations and conserved regions across different viral genomes need to be analyzed to understand how viruses adapt to hosts and environments, particularly in regions with high ecological diversity like Indonesia. Effective monitoring of pathogen dynamics requires tools that can map the relationships between local and global pathogen strains. These tools are essential for understanding viral movement, adaptation, and potential risks of zoonotic spillovers.

Phylogenetic trees serve as vital tools for understanding pathogen dynamics, particularly in biodiverse regions like Indonesia. They offer insights into the origins and spread of infectious diseases, helping researchers predict outbreak risks and identify transmission pathways. Phylogenetic trees help in:

1. Understanding Evolutionary Connections
   Phylogenetic trees visualize evolutionary relationships among pathogens based on genomic data. By aligning and comparing viral sequences, the tree identifies divergent points that are traced back to common ancestors. For example, comparing COVID-19 with SARS and MERS can reveal how these viruses evolved from shared origins.
2. Tracing Zoonotic Transmission
   By including genomes from potential animal reservoirs, phylogenetic trees help pinpoint the zoonotic origins of viruses. This analysis is particularly relevant in Indonesia, where interactions between humans and wildlife increase the risk of cross-species transmission.
3. Regional Insights and Global Context
   Including genomic sequences from Indonesian strains in a phylogenetic tree highlights the role of regional biodiversity in viral evolution. It also helps determine whether these strains are linked to global outbreaks or have unique evolutionary pathways.

## IV. IMPLEMENTATION

Phylogenetic analysis involves comparing genomic sequences to uncover evolutionary relationships among pathogens. This implementation uses genome sequences of viruses such as SARS, MERS, COVID-19, along with animal reservoir genomes like bats and civets. The primary steps include genome preparation, multiple sequence alignment (MSA), similarity calculation, and phylogenetic tree construction. These steps ensure a comprehensive understanding of evolutionary patterns and pathogen dynamics, especially in a biodiverse region like Indonesia.

1. Genome Data Preparation

   The genome sequences for viruses like SARS, MERS, COVID-19, and others are read from .fasta files. This step extracts raw genomic data into a format suitable for alignment.

```
# sars genome
with open("/kaggle/input/genome-dataset/genome-dataset/sars.fasta", "r") as file:
    sars_genome = file.read().split("genome")[1].replace("\n", "")
# mers genome
with open("/kaggle/input/genome-dataset/genome-dataset/mers.fasta", "r") as file:
    mers_genome = file.read().split("genome")[1].replace("\n", "")
# covid-19 genome
with open ("/kaggle/input/genome-dataset/genome-dataset/cov2.fasta", "r") as file:
    cov_genome = file.read().split("genome")[1].replace("\n", "")
# bat sars-cov genome
with open("/kaggle/input/genome-dataset/genome-dataset/batsars.fasta", "r") as file:
    bat_genome = file.read().split("complete genome")[-1].replace("\n","")
# camel mers-cov genome
with open("/kaggle/input/genome-dataset/genome-dataset/camelsars.fasta", "r") as file:
    camel_genome = file.read().split("complete genome")[-1].replace("\n", "")
# civet sars-cov genome
with open("/kaggle/input/genome-dataset/genome-dataset/civetsars.fasta", "r") as file:
    civet_genome = file.read().split("complete genome")[-1].replace("\n","")
# ebola-5 genome
with open("/kaggle/input/genome-dataset/genome-dataset/ebolav.fasta", "r") as file:
    ebola5_genome = file.read().split("complete genome")[-1].replace("\n","")
# hiv-2 genome
with open("/kaggle/input/genome-dataset/genome-dataset/hiv2.fasta", "r") as file:
    hiv2_genome = file.read().split("complete genome")[-1].replace("\n", "")
# malaria genome
with open("/kaggle/input/genome-dataset/genome-dataset/malaria.fasta", "r") as file:
    malaria_genome = file.read().split("complete sequence")[-1].replace("\n", "")
```

*Fig. 4.1 Implementation of Dataset Preparation*
*Source: Author*

2. Trinucleotide Composition Analysis

   This step is needed to identify characteristic patterns within genomes that might reflect functional or evolutionary similarities. It provides insight into the nucleotide usage bias, which could be linked to gene expression or codon preference in specific pathogens.

```
# tri-nucleotide compositions
trinucleotides = ["AAA", "AAC", "AAG", "AAT", "ACA", "ACC", "ACG", "ACT",
                  "AGA", "AGC", "AGG", "ATA", "ATC", "ATG", "CAA", "CAC",
                  "CAG", "CCA","CCC","CCG","CGA","CGC","CTA","CTC","GAA"
                  "GAC","GCA","GCC","GGA","GTA","TAA","TCA"]

def trinucleotide_composition(genome):
    trinucleotide_dict = {trinucleotide: genome.count(trinucleotide) for
                          trinucleotide in trinucleotides}
    return trinucleotide_dict


labels = ["SARS", "MERS", "COVID-19", "BAT SARS-CoV","CAMEL MERS-CoV",
          "CIVET SARS-CoV","EBOLA","HIV","MALARIA"]
genomes = [sars_genome, mers_genome, cov_genome, bat_genome, camel_genome,
           civet_genome, ebola5_genome, hiv2_genome, malaria_genome]

plt.figure(figsize=(10, 6))

for i, genome in enumerate(genomes):
    composition = trinucleotide_composition(genome)
    total_composition = sum(composition.values())
    freq = [count / total_composition for count in composition.values()]
    plt.plot(trinucleotides, freq, label=labels[i])

plt.title("Trinucleotide Composition of Genomes")
plt.xlabel("Trinucleotides")
plt.ylabel("Frequencies")
plt.xticks(rotation=90)
plt.legend(loc="upper right")
plt.tight_layout()

plt.show()
```

*Fig. 4.2 Implementation of Trinucleotide Composition*
*Source: Author*

3. Pairwise Sequence Silimarity

   Pairwise comparisons are made between the COVID-19 genome and others. A similarity score is computed using the alignment object, which quantifies how closely related two genomes are.

```
# define aligner
aligner = Align.PairwiseAligner()
aligner.mode = 'global'
aligner.match_score = 1
aligner.mismatch_score = -1
aligner.open_gap_score = -0.5
aligner.extend_gap_score = -0.1

# prepare genomes as sequences
genomes = {
    "SARS": Seq(sars_genome),
    "MERS": Seq(mers_genome),
    "BAT SARS-CoV": Seq(bat_genome),
    "CAMEL MERS-CoV": Seq(camel_genome),
    "CIVET SARS-CoV": Seq(civet_genome),
    "EBOLA": Seq(ebola5_genome),
    "HIV": Seq(hiv2_genome),
    "MALARIA": Seq(malaria_genome),
}

# compute similarity scores
print("Similarity scores between")
cov_seq = Seq(cov_genome)  # COVID-19 genome sequence
for genome_name, genome in genomes.items():
    score = aligner.score(cov_seq, genome)
    similarity_percentage = 100 * (score / len(cov_seq))
    print(f"COVID-19 & {genome_name} genome sequences:\t {score:.2f} ({similarity_percentage:.2f}%)")
```

*Fig. 4.3 Implementation of Pairwise Sequence Similarity*
*Source: Author*

4. Construct Distance Matrix

   Based on pairwise scores, a symmetrical distance matrix is constructed. Each cell contains the score between two genomes. This matrix forms the foundation for tree construction.

```python
# initialize pairwise aligner
aligner = PairwiseAligner()
aligner.mode = 'global'
aligner.match_score = 1
aligner.mismatch_score = -1
aligner.open_gap_score = -0.5
aligner.extend_gap_score = -0.1

# define genomes
genomes = {
    "COVID-19": cov_genome,
    "SARS": sars_genome,
    "MERS": mers_genome,
    "Bat-SARS-CoV": bat_genome,
    "Camel-MERS-CoV": camel_genome,
    "Civet-SARS-CoV": civet_genome,
    "Ebola": ebola5_genome,
    "HIV": hiv2_genome,
    "Malaria": malaria_genome,
}

# calculate pairwise distances
labels = list(genomes.keys())
distances = []

for i, label1 in enumerate(labels):
    row = []
    for j, label2 in enumerate(labels[:i]):
        score = aligner.score(genomes[label1], genomes[label2])
        max_score = max(len(genomes[label1]), len(genomes[label2]))
        distance = 1 - (score / max_score)
        row.append(distance)
    distances.append(row)
```

*Fig. 4.4 Implementation of Pairwise Distance Calculation*
*Source: Author*

```python
# construct distance matrix in Newick format
def construct_newick_matrix(labels, distances):
    headers = labels[:]
    matrix = [[distances[i][j] if i > j else distances[j][i] if i < j else 0 for j in range(len(labels))] for i in range(len(labels))]

    while len(headers) > 1:
        min_val = float("inf")
        x, y = -1, -1
        for i in range(len(matrix)):
            for j in range(len(matrix[i])):
                if i != j and matrix[i][j] < min_val:
                    min_val = matrix[i][j]
                    x, y = i, j

        # merge clusters
        headers[x] = f"({headers[x]},{headers[y]})"
        for i in range(len(matrix)):
            if i != x and i != y:
                matrix[i][x] = matrix[x][i] = (matrix[x][i] + matrix[y][i]) / 2 if i < x else (matrix[i][x] + matrix[i][y]) / 2

        # remove the merged cluster (row and column)
        matrix = [[matrix[i][j] for j in range(len(matrix)) if j != y] for i in range(len(matrix)) if i != y]
        headers.pop(y)

    return headers[0] + ";"
```

*Fig. 4.5 Implementation of Distance Matrix Construction*
*Source: Author*

```python
# generate the distance matrix
labels = list(genomes.keys())
distances = []

for i, label1 in enumerate(labels):
    row = []
    for j, label2 in enumerate(labels[:i]):
        score = aligner.score(genomes[label1], genomes[label2])
        max_score = max(len(genomes[label1]), len(genomes[label2]))
        distance = 1 - (score / max_score)  # Normalize distances
        row.append(distance)
    distances.append(row)

# create a Newick string
newick_tree = construct_newick_matrix(labels, distances)
```

*Fig. 4.6 Implementation of Distance Matrix Generation*
*Source: Author*

5. Phylogenic Tree Construction
   Phylogenic tree is constructed using Phylo library from Biopython.

```python
# render the phylogenetic tree
from Bio import Phylo
from io import StringIO

tree = Phylo.read(StringIO(newick_tree), "newick")
Phylo.draw(tree)
```

*Fig. 4.7 Implementation of Phylogenic Tree Render*
*Source: Author*

## V. TESTING AND ANALYSIS

The dataset used in this analysis consists of genome sequences from various pathogens, including COVID-19, SARS, MERS, and other related viruses such as Bat SARS-CoV, Camel MERS-CoV, Civet SARS-CoV, Ebola, HIV, and Malaria. These genome sequences are obtained from a curated dataset available on Kaggle, stored in FASTA format. Each genome file contains the nucleotide sequence for the respective pathogen. The genome data files are processed using Python and libraries such as Biopython for handling sequence data, aligning genomes, and calculating genetic distances. The genomes are split into their relevant sections and cleaned to remove unnecessary formatting, enabling accurate computational analysis.

Trinucleotide composition of all genomes is made to compare the frequency of different trinucleotide sequences (three-letter nucleotide combinations) across the genomes. It can be used to understand potential evolutionary relationships by examining how the trinucleotide usage differs or aligns across these organisms. It reveals significant similarities between COVID-19, SARS, and Bat SARS-CoV, suggesting close evolutionary relationships. This supports the hypothesis that SARS-CoV-2 (COVID-19) shares a common ancestor with bat coronaviruses. In contrast, distinct patterns in pathogens like Ebola, HIV, and Malaria highlight their unique genomic architecture.

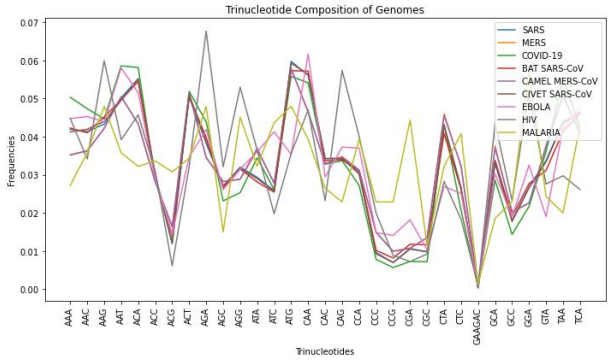


*Fig. 5.1 Trinucleotide Composition of Genomes*
*Source: Author*

Then, pairwise comparisons are made between the COVID-19 genome and others. The similarity scores between COVID-19 and other pathogens highlight key evolutionary relationships and differences. COVID-19 shows the highest similarity with SARS (69.84%) and Bat SARS-CoV (69.24%). Civet SARS-CoV also exhibits high similarity (68.95%), reflecting the role of civets in SARS-CoV transmission. In contrast, MERS and Camel MERS-CoV display lower similarities (50.57% and 50.61%, respectively), indicating more distant relationships within the coronavirus family. Pathogens like Ebola (34.22%), HIV (19.94%), and Malaria (0.03%) show significantly lower similarity, underscoring their distinct genomic architectures and evolutionary divergence.

```
Similarity scores between
COVID-19 & SARS genome sequences:          20885.00 (69.84%)
COVID-19 & MERS genome sequences:          15122.60 (50.57%)
COVID-19 & BAT SARS-CoV genome sequences:          20706.00 (69.24%)
COVID-19 & CAMEL MERS-CoV genome sequences:        15134.60 (50.61%)
COVID-19 & CIVET SARS-CoV genome sequences:        20616.90 (68.95%)
COVID-19 & EBOLA genome sequences:         10233.40 (34.22%)
COVID-19 & HIV genome sequences:           5962.60 (19.94%)
COVID-19 & MALARIA genome sequences:       8.80 (0.03%)
```

*Fig. 5.2 Seqeuence Similarity Between COVID-19 and Others*
*Source: Author*

The phylogenetic tree provides a visual representation of the evolutionary relationships among the listed pathogens, based on genomic similarity. The clustering of COVID-19, SARS, Civet-SARS-CoV, and Bat-SARS-CoV at the top indicates their close evolutionary relationships, reinforcing the idea that SARS-CoV-2 likely shares a common ancestor with bat-origin coronaviruses and civet-hosted viruses. MERS and Camel-MERS-CoV form another distinct cluster, suggesting a shared evolutionary path, but their greater distance from COVID-19 and SARS clusters reflects their divergence within the coronavirus family. Further down the tree, pathogens like Ebola, HIV, and Malaria form separate branches, indicating significant genomic dissimilarities compared to coronaviruses. Malaria, at the base of the tree, is the most genetically distinct pathogen, as it is caused by a eukaryotic parasite rather than a virus.

This tree demonstrates the utility of phylogenetics in understanding pathogen evolution, tracing origins, and identifying genetic relationships, which can guide research and public health strategies.

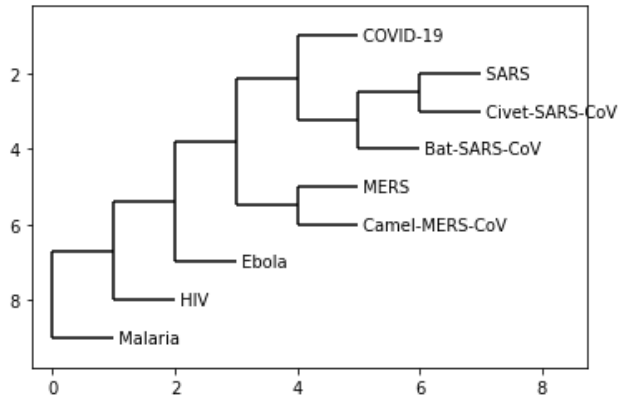


*Fig. 5.3 Phylogenic Tree*
*Source: Author*

## VI. Conclusion

The analysis of genomic data, including trinucleotide composition, similarity scoring, and phylogenetic tree construction, has provided critical insights into the evolutionary relationships and genetic characteristics of pathogens. COVID-19 (SARS-CoV-2) shows a high degree of genomic similarity with SARS and bat-origin coronaviruses, supporting the hypothesis of a zoonotic origin. This close relationship is further highlighted in the phylogenetic tree, where these pathogens cluster together, reflecting shared evolutionary pathways. In contrast, MERS, though part of the coronavirus family, demonstrates significant divergence, forming its own cluster with Camel-MERS-CoV and indicating a distinct evolutionary lineage. Pathogens such as Ebola, HIV, and Malaria are genetically distant from coronaviruses, as evidenced by their low similarity scores and distant positions in the phylogenetic tree.

The phylogenetic tree emerges as a powerful tool to visualize these evolutionary relationships, aiding in the identification of common ancestors and divergence points. This understanding is crucial for tracing the origins of outbreaks, predicting future zoonotic spillovers, and guiding vaccine and treatment research. For Indonesia, a country with rich biodiversity and frequent human-animal interactions, these insights are particularly valuable. They can inform public health strategies, enhance surveillance programs, track emerging diseases, and assess the potential for cross-species transmission.

In conclusion, this study demonstrates how computational genomic analysis can uncover evolutionary relationships, improve understanding of pathogen dynamics, and provide insights for combating infectious diseases. By integrating genomic similarity analysis with phylogenetic methods, we can better address the complexities of infectious disease outbreaks and their impact on global health.

## VII. Acknowledgment

The completion of this paper is a testament to God’s grace, providing strength, perseverance, and clarity throughout the process. The writer also extends heartfelt gratitude to Dr. Ir. Rinaldi Munir for his invaluable guidance and dedication to teaching during the Discrete Mathematics course. His insightful mentorship, extensive learning materials, and contributions to the field have greatly enriched the writer’s understanding and played a pivotal role in the successful completion of this work.

## References

[1] R. Munir, "Graf (Bag.1)," [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/20-Graf-Bagian1-2024.pdf . Accessed: Jan. 4, 2025.

[2] R. Munir, "Pohon (Bag.1)," [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2024-2025/23-Pohon-Bag1-2024.pdf . Accessed: Jan. 4, 2025.

[3] A. Singh, K. Gupta, and A. Sharma, "Deep phylogenetic-based clustering analysis uncovers new and emerging SARS-CoV-2 variants harboring signature mutations in the receptor-binding domain," *PLOS ONE*, vol. 17, no. 5, e0268389, 2022. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0268389 . Accessed: Jan. 8, 2025.

[4] D. M. Hillis, C. Moritz, and B. K. Mable, "Molecular Systematics," 2nd ed. Sunderland, MA: Sinauer Associates, 1996, pp. 515–543.

[5] M. F. Boni, P. Lemey, X. Jiang, T. T. Lam, B. W. Perry, T. A. Castoe, A. Rambaut, and D. L. Robertson, "Evolutionary origins of the SARS-CoV-2 sarbecovirus lineage responsible for the COVID-19 pandemic," *Nature Microbiology*, vol. 5, pp. 1408–1417, 2020. [Online]. Available: https://www.nature.com/articles/s41564-020-0771-4. Accessed: Jan. 7, 2025.

STATEMENT

Hereby, I declare that this paper I have written is my own work, not a reproduction or translation of someone else's paper, and not plagiarized.

Bandung, 8 January 2025

Grace Evelyn Simon, 13523087