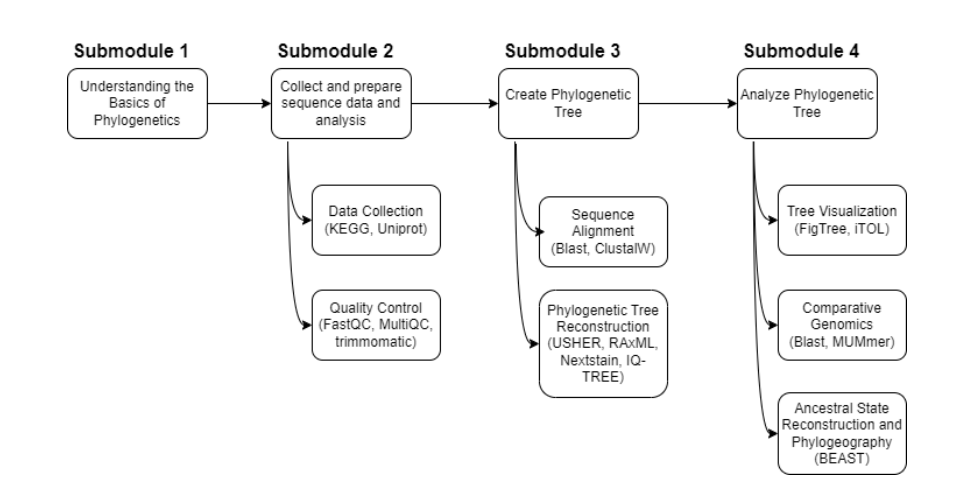
NOSI Phylogenetic Tree Analysis Workflow



# Submodule 1: Understanding the Basics of Phylogenetic

## What is a Phylogenetic Tree?

A phylogenetic tree is a visual representation that illustrates the evolutionary connections and ancestral relationships among different biological species or organisms. It depicts how closely related these species are based on similarities and differences in their physical traits or genetic makeup. For example, a phylogenetic tree of various primates (like humans, chimpanzees, gorillas, and orangutans) would show how closely or distantly related these species are to one another based on their shared characteristics and common ancestors. Essentially, a phylogenetic tree is a hypothesis or educated guess about the evolutionary history and lineage of the organisms being studied.

## The Purpose of Phylogenetic Trees

Phylogenetic trees serve several crucial functions in the field of evolutionary biology and related disciplines:

1. **Tracing Evolutionary Pathways**: These trees help scientists understand how different species have evolved over time from common ancestors. They provide insights into the branching patterns and evolutionary relationships between organisms.
2. **Mapping Genetic Changes**: By analyzing genetic sequences, phylogenetic trees enable researchers to track genetic changes that have occurred throughout the evolutionary process. This helps in studying the evolution of specific traits or genes across different species.
3. **Understanding Biodiversity:** The branching patterns in phylogenetic trees shed light on the diversification of species and the development of new characteristics or traits. This knowledge contributes to our understanding of the vast biodiversity on Earth.
4. **Disease Research:** In the context of pathogens, such as viruses or bacteria, phylogenetic trees are invaluable tools for tracking the spread and evolution of diseases. They help identify the origins, transmission pathways, and potential sources of disease outbreaks.

## Data Sources for Creating Phylogenetic Trees

To construct accurate phylogenetic trees, researchers rely on various data sources:

1. **Genetic Sequences**: The primary data used in phylogenetic analysis are DNA, RNA, or protein sequences obtained from different species or strains. These sequences are compared to identify similarities and differences.
2. **Public Databases:** Genetic sequence data can be accessed from public repositories like GenBank, EMBL, and DDBJ, which maintain comprehensive and annotated genetic information for numerous organisms.
3. **Genomic Projects:** Large-scale genomic projects, such as the Human Genome Project or the 1000 Genomes Project, provide extensive datasets that can be utilized for phylogenetic studies.
4. **Sequencing Technologies:** Advances in sequencing technologies, like next-generation sequencing (NGS), have made it easier and more cost-effective to obtain high-quality genetic data for a wide range of organisms.

## Types of Phylogenetic Trees

Phylogenetic trees can be classified into different types based on their structure and the information they convey:

1. **Rooted Trees:** These trees have a single ancestral root, representing the common ancestor of all the entities in the tree. The direction of the branches indicates the passage of time and evolutionary divergence.
2. **Unrooted Trees:** Unrooted trees do not show a common ancestor but illustrate the relationships among species without indicating the direction of evolutionary time.
3. **Cladograms**: Cladograms represent the branching order of evolutionary relationships but do not provide information about the branch lengths or the amount of evolutionary change.
4. **Phylograms:** Phylograms provide both the branching order and the branch lengths, indicating the amount of evolutionary change along each branch.
5. **Dendrograms:** Similar to phylograms, dendrograms can also include hierarchical clustering, making them useful in various fields like genomics and linguistics.

By studying phylogenetic trees, scientists can gain valuable insights into the evolutionary history, relationships, and diversification of different organisms, ultimately expanding our understanding of the intricate tapestry of life on Earth.

# Submodule2:

## Collect and prepare sequence data and analysis:

#### 2.1 Demonstrate Efficient Methods for Sourcing Pathogen Sequences and Preparing Data for Phylogenetic Analysis

Efficient methods for sourcing pathogen sequences and preparing data include:

* **Public Databases**: Utilize public repositories like GenBank, EMBL, and DDBJ for sourcing high-quality pathogen sequences. These databases provide comprehensive and well-annotated genetic data.
* **Sequence Retrieval Tools**: Use tools like Entrez Direct and Biopython to automate the retrieval of sequences from public databases, ensuring efficiency and reducing the risk of manual errors.
* **Data Cleaning and Preprocessing**: Implement tools such as Trimmomatic for trimming low-quality reads and removing adapters from raw sequence data. This step is crucial for ensuring the integrity of the sequences used in downstream analysis.
* **Sequence Alignment**: Use alignment tools like MAFFT or ClustalW to align sequences, correcting for gaps and mismatches to maximize homology. Proper alignment is essential for accurate phylogenetic tree construction​​.

#### 2.2 Discuss the Importance of Cloud-Based Storage Solutions in Managing Metagenomic Sequence Data

Cloud-based storage solutions are vital for managing metagenomic sequence data due to:

* **Scalability**: Cloud storage offers scalable solutions to handle the vast amounts of data generated by metagenomic studies. It allows researchers to store large datasets without worrying about local storage limitations.
* **Accessibility**: Cloud storage ensures that data can be accessed from anywhere, facilitating collaboration among researchers across different geographical locations.
* **Cost-Effectiveness**: Cloud services often operate on a pay-as-you-go model, making it cost-effective as researchers only pay for the storage and computational resources they use.
* **Data Security and Backup**: Cloud providers offer robust security measures and automatic backups, protecting valuable data from loss or unauthorized access​​.

#### 2.3 Explain How the Incorporation of Publicly Available Datasets from Reputable Metagenomic Databases Enhances the Depth of Analysis

Incorporating publicly available datasets from reputable metagenomic databases enhances analysis by:

* **Increased Data Volume**: Access to a large volume of data increases the statistical power and robustness of analyses, enabling more accurate and comprehensive studies.
* **Comparative Analysis**: Public datasets provide a wealth of comparative data, allowing researchers to identify patterns, variations, and evolutionary trends across different studies and datasets.
* **Validation and Reproducibility**: Using standardized, publicly available data ensures that results can be validated and reproduced by other researchers, enhancing the credibility of the findings.
* **Resource Sharing**: Public databases foster a collaborative environment where researchers can share resources, tools, and data, accelerating the pace of scientific discovery​​.

### KEGG Dataset:

#### Dataset 1: KEGG for Phylogenetic Tree

##### Downloading KEGG Dataset

KEGG (Kyoto Encyclopedia of Genes and Genomes) provides a wealth of data for understanding high-level functions and utilities of biological systems. To download KEGG data:

1. **Access KEGG Dataset:** KEGG provides a website for data retrieval.
   1. <https://www.genome.jp/kegg/seq/>
   2. Downloaded file from FASTA sequence files section on the website.
2. **More resources on KEGG**: https://www.genome.jp/kegg/

#### Dataset 2: UniProt for Phylogenetic Tree

##### Downloading UniProt Dataset

UniProt is a comprehensive resource for protein sequence and functional information. To download UniProt data:

1. **Access UniProt Website**: Visit the UniProt website and search for the desired protein sequences.
2. **Retrieve Data**: Downloaded the Isoform sequences fasta file from the website.
   1. **File name**: uniprot\_sprot\_varsplic.fasta

#### 2.4 Implement Quality Control Checks Using Tools Like MultiQC and FastQC to Ensure Data Integrity

**Using FASTQC for Quality Control.**

* Followed installation and setup from:
  + <https://raw.githubusercontent.com/s-andrews/FastQC/master/INSTALL.txt>
* First need to install JAVA as FastQC uses JAVA
* Input file used to run fastqc was downloaded from:
  + <https://ftp.uniprot.org/pub/databases/uniprot/current_release/knowledgebase/complete/uniprot_sprot_varsplic.fasta.gz>
* First need to convert .fasta file to fastq file
* Code:

from Bio import SeqIO  
  
  
def fasta\_to\_fastq(fasta\_file, fastq\_file, quality=40):  
 with open(fastq\_file, "w") as output\_handle:  
 for record in SeqIO.parse(fasta\_file, "fasta"):  
 record.letter\_annotations["phred\_quality"] = [quality] \* len(record.seq)  
 SeqIO.write(record, output\_handle, "fastq")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 fasta\_file = "path\_to\_fasta\_file"  
 fastq\_file = "path\_to\_output\_fastq\_file"  
 fasta\_to\_fastq(fasta\_file=fasta\_file, fastq\_file=fastq\_file)

* RUN FASTQC command line:
  + fastqc -t 4 uniprot\_sprot\_varsplic.fastq

# Submodule 3

## Construct Phylogenetic Tree

#### 3.1 Perform Accurate Sequence Alignment of Metagenomic Data using ClustalW

Sequence alignment is a critical step in phylogenetic analysis, as it arranges the sequences in a manner that highlights their similarities and differences, allowing for accurate tree construction.

**Using ClustalW for Sequence Alignment:**

1. **Download ClustalW**: Obtain the ClustalW tool from its official website: [ClustalW Download](http://www.clustal.org/clustal2/#Download)
2. **Install ClustalW**: Follow the installation instructions specific to your operating system. For example, on Windows, it is typically installed at:

C:\Program Files (x86)\ClustalW2\clustalw2.exe

1. **Run ClustalW using Python and Biopython**:

import subprocess  
  
import matplotlib.pyplot as plt  
import networkx as nx  
from Bio import AlignIO  
from Bio.Phylo.TreeConstruction import DistanceTreeConstructor, DistanceCalculator  
  
# Define the paths  
fasta\_file = "data/cov/sequences.fasta"  
clustalw\_exe = "C:\\Program Files (x86)\\ClustalW2\\clustalw2.exe"  
seq\_algn\_file = "data/cov/sequences.aln"  
# Run ClustalW for multiple sequence alignment using subprocess  
try:  
 subprocess.run([clustalw\_exe, "-INFILE=" + fasta\_file, "-OUTFILE=" + seq\_algn\_file, "-OUTPUT=FASTA"], check=True)  
except subprocess.CalledProcessError as e:  
 print("Error running ClustalW:", e)  
 exit(1)

#### 3.2 Manage Computational Intensity through Cloud Computing

Due to the large size of metagenomic datasets, sequence alignment can be computationally intensive. Utilizing cloud computing resources can significantly enhance the efficiency and speed of these tasks.

**Benefits of Cloud Computing for Sequence Alignment:**

* **Scalability**: Easily scale up resources based on the demand of the computation.
* **Cost-Effectiveness**: Pay-as-you-go models allow for cost savings by only using resources when needed.
* **Accessibility**: Access computational resources and data from anywhere, facilitating collaboration among researchers.

#### 3.3 Phylogenetic Tree Reconstruction using USHER

USHER (Ultrafast Sample Placement on Existing tRee) is a tool designed to rapidly place samples on a given phylogenetic tree. It is particularly useful for large-scale phylogenetic analysis and real-time epidemiology.

**Steps to Use USHER for Phylogenetic Tree Reconstruction:**

1. **Cloning USHER Repository**:
   * Clone the USHER repository from GitHub:
     1. git clone https://github.com/yatisht/usher.git
2. **Installing Dependencies**:
   * Update the conda environment with the necessary dependencies:
     1. conda env update -f usher/workflows/envs/usher.yaml
3. **Installing Additional Packages**:
   * Install the required packages mafft and fasttree:
     1. conda install -c bioconda mafft fasttree
4. **Aligning Sequences**:
   * Use mafft to align your sequences and output them to aligned\_sequences.fasta:
     1. mafft --auto sequences.fasta > aligned\_sequences.fasta
5. **Generating VCF File**:
   * Convert the aligned sequences to a VCF file:
     1. faToVcf aligned\_sequences.fasta seq.vcf
6. **Creating Newick Tree File**:
   * Use fasttree to generate a Newick tree file:
     1. fasttree -nt aligned\_sequences.fasta > reference.nwk
7. **Running USHER**:
   * With the aligned sequences, VCF file, and Newick tree file, run USHER:
     1. usher -t reference.nwk -v seq.vcf -o seq\_output.nwk

# Submodule 4:

## Analyze Phylogenetic Trees

#### 4.1 Interpret and Visually Represent Phylogenetic Trees

Visualization tools are essential for interpreting and presenting phylogenetic trees.

**Tools for Tree Visualization:**

* **iTOL (Interactive Tree of Life)**: An online tool for the display and annotation of phylogenetic trees.
  + **Upload the seq\_output.nwk file**: [iTOL Upload](https://itol.embl.de/upload.cgi)

#### 4.2 Importance of Visual Representation

Visual representation of phylogenetic trees aids in:

* **Interpreting Results**: Makes it easier to understand evolutionary relationships.
* **Communication**: Helps in conveying findings to a broader audience, including those who may not be specialists in phylogenetics.
* **Highlighting Key Features**: Emphasizes important evolutionary events and patterns.

#### 4.3 Conduct Comparative Metagenomics along Different Branches

Comparative metagenomics involves comparing the genetic content of different samples to uncover variations.

**Steps for Comparative Metagenomics:**

1. **Install BLAST**:

conda install -c bioconda blast

1. **Create a BLAST Database**:

makeblastdb -in sequences.fasta -dbtype nucl -out seq\_database

1. **Prepare Query Sequences**:
   * Create a query\_sequences.fasta file with the sequences you want to compare.
2. **Run BLAST**:

blastn -query query\_sequences.fasta -db seq\_database -out seq\_results.txt -outfmt 6

* + The results will be in seq\_results.txt.

#### 4.4 Automate Comparative Metagenomics Analysis using Biopython

Automation can streamline comparative metagenomics analysis, making it more efficient.

**Script for Automation**:

from Bio.Blast import NCBIWWW, NCBIXML  
  
# Function to run BLAST and parse results  
def run\_blast(query\_file, db\_file, output\_file):  
 result\_handle = NCBIWWW.qblast("blastn", db\_file, query\_file)  
 with open(output\_file, "w") as out\_handle:  
 out\_handle.write(result\_handle.read())  
 result\_handle.close()  
  
# Run the BLAST  
run\_blast("data/cov/query\_sequences.fasta", "data/cov/seq\_database", "blast\_results.xml")  
  
# Parse the BLAST results  
with open("blast\_results.xml") as result\_handle:  
 blast\_records = NCBIXML.parse(result\_handle)  
 for blast\_record in blast\_records:  
 for alignment in blast\_record.alignments:  
 for hsp in alignment.hsps:  
 print(f"\*\*\*\*Alignment\*\*\*\*")  
 print(f"sequence: {alignment.title}")  
 print(f"length: {alignment.length}")  
 print(f"e value: {hsp.expect}")  
 print(f"{hsp.query[0:75]}...")  
 print(f"{hsp.match[0:75]}...")  
 print(f"{hsp.sbjct[0:75]}...")

#### 4.5 Discuss Insights from Ancestral State Reconstruction

Ancestral state reconstruction provides insights into:

* **Evolutionary Dynamics**: Understanding how certain traits or genetic sequences have evolved over time.
* **Diversity**: Gaining a deeper understanding of the diversity within and between metagenomic samples.
* **Evolutionary Pressures**: Identifying the evolutionary pressures that have shaped the genetic makeup of organisms.

#### 4.6 Utilize Bayesian Inference Methods with BEAST for Ancestral State Reconstruction

Bayesian inference methods are powerful for reconstructing ancestral states and understanding evolutionary dynamics.

**Using BEAST for Ancestral State Reconstruction:**

1. **Install BEAST**:

conda install beast -c bioconda

1. **Check Installation**:

beast -beagle\_info

1. **Launch BEAUti (Graphical User Interface for BEAST)**:

find $CONDA\_PREFIX -name "beauti"

* + Use the path provided to launch BEAUti.

1. **Prepare BEAST Input File**:
   * Import your aligned sequence file (e.g., aligned\_sequences.fasta).
   * Configure the settings and generate the BEAST XML file.
2. **Run BEAST**:

beast seq\_config.xml

SARS-CoV-2 Data Analysis

# Obtaining SARS-CoV-2 Data

Downloaded SARS-CoV-2 sequence data from the Nextstrain project: <https://docs.nextstrain.org/projects/nextclade/en/stable/user/datasets.html>

The specific file used is: data/nextstrain/sars-cov-2/wuhan-hu-1/proteins/sequences.fasta

# Data Preparation

Convert the .fasta file to .fastq format using the following Python code:

from Bio import SeqIO  
  
  
def fasta\_to\_fastq(fasta\_file, fastq\_file, quality=40):  
 with open(fastq\_file, "w") as output\_handle:  
 for record in SeqIO.parse(fasta\_file, "fasta"):  
 record.letter\_annotations["phred\_quality"] = [quality] \* len(record.seq)  
 SeqIO.write(record, output\_handle, "fastq")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 fasta\_file = "data/cov/sequences.fasta"  
 fastq\_file = "data/cov/sequences\_converted.fastq"  
 fasta\_to\_fastq(fasta\_file=fasta\_file, fastq\_file=fastq\_file)

# Quality Control

Run FastQC on the converted file:

**fastqc -t 4 data/cov/sequences\_converted.fastq**

This step checks the quality of our sequence data.

# Sequence Alignment

We use ClustalW for sequence alignment. Download it from: <http://www.clustal.org/clustal2/#Download>

Install ClustalW on your system. On Windows, it's typically installed at: "C:\Program Files (x86)\ClustalW2\clustalw2.exe"

You can run ClustalW from the command line or use it in Python with Biopython:

import subprocess  
  
import matplotlib.pyplot as plt  
import networkx as nx  
from Bio import AlignIO  
from Bio.Phylo.TreeConstruction import DistanceTreeConstructor, DistanceCalculator  
  
# Define the paths  
fasta\_file = "data/cov/sequences.fasta"  
clustalw\_exe = "C:\\Program Files (x86)\\ClustalW2\\clustalw2.exe"  
seq\_algn\_file = "data/cov/sequences.aln"  
# Run ClustalW for multiple sequence alignment using subprocess  
try:  
 subprocess.run([clustalw\_exe, "-INFILE=" + fasta\_file, "-OUTFILE=" + seq\_algn\_file, "-OUTPUT=FASTA"], check=True)  
except subprocess.CalledProcessError as e:  
 print("Error running ClustalW:", e)  
 exit(1)

This process aligns the SARS-CoV-2 sequences, preparing them for phylogenetic tree construction.