# phylogeny\_builder.py saved under 4. Source code folder

‘’’

Execution of below will construct a phylogenetic tree based on the sequence data in the database. This tree represents the evolutionary relationships between the sequences.

Phylo module from the BioPython library is used to generate a phylogenetic tree. The logic used is to align all sequences in the database and then construct a tree using the Neighbor-Joining method.

The following executable steps are included:

**build\_phylogenetic\_tree**: This function takes the sequences, creates a multiple sequence alignment, and calculates a **distance matrix** to construct a phylogenetic tree using the **Neighbor-Joining method**.

**DistanceTreeConstructor**: This tool in **BioPython** constructs trees from distance matrices.

**Phylo.draw**: This function visualizes the tree and optionally saves it as an image.

‘’’

from Bio import Phylo

from Bio.Phylo.TreeConstruction import DistanceTreeConstructor, \_DistanceMatrix

from Bio.Align import MultipleSeqAlignment

from Bio import SeqIO

def build\_phylogenetic\_tree(sequences):

"""Build a phylogenetic tree using the Neighbor-Joining method."""

# Create a multiple sequence alignment from the sequences

alignment = MultipleSeqAlignment(sequences)

# Create a distance matrix based on the sequence alignment

constructor = DistanceTreeConstructor()

dm = \_DistanceMatrix(names=[seq.id for seq in sequences], data=alignment)

# Build the tree using Neighbor-Joining method

tree = constructor.nj(dm)

return tree

def main():

# Load the dog breeds sequences from the database

sequences = list(SeqIO.parse("data/dog\_breeds.fa", "fasta"))

# Build the phylogenetic tree

tree = build\_phylogenetic\_tree(sequences)

# Print the tree in a Newick format

print(tree)

# Optionally, save the tree as a PNG image

tree.ladderize()

Phylo.draw(tree)

if \_\_name\_\_ == "\_\_main\_\_":

main()