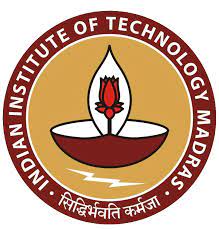
..

**BT 3040: Bioinformatics**

**Assignment 11**



Indian Institute of Technology Madras

Atharva Mandar Phatak | BE21B009

Department of Biotechnology

## **Q1**) **Obtain the consensus phylogenetic tree for the following two sets of sequences:**

#Q1|Assignment 11 | BT3040 | Atharva Mandar Phatak | BE21B009

import numpy as np

import pandas as pd

import pprint as pprint

amino\_acids = [

    "A",  # Alanine

    "R",  # Arginine

    "N",  # Asparagine

    "D",  # Aspartic Acid

    "C",  # Cysteine

    "Q",  # Glutamine

    "E",  # Glutamic Acid

    "G",  # Glycine

    "H",  # Histidine

    "I",  # Isoleucine

    "L",  # Leucine

    "K",  # Lysine

    "M",  # Methionine

    "F",  # Phenylalanine

    "P",  # Proline

    "S",  # Serine

    "T",  # Threonine

    "W",  # Tryptophan

    "Y",  # Tyrosine

    "V"   # Valine

]

prop\_dict = {

    "A": 0,  # Alanine

    "R": 0,  # Arginine

    "N": 0,  # Asparagine

    "D": 0,  # Aspartic Acid

    "C": 0,  # Cysteine

    "Q": 0,  # Glutamine

    "E": 0,  # Glutamic Acid

    "G": 0,  # Glycine

    "H": 0,  # Histidine

    "I": 0,  # Isoleucine

    "L": 0,  # Leucine

    "K": 0,  # Lysine

    "M": 0,  # Methionine

    "F": 0,  # Phenylalanine

    "P": 0,  # Proline

    "S": 0,  # Serine

    "T": 0,  # Threonine

    "W": 0,  # Tryptophan

    "Y": 0,  # Tyrosine

    "V": 0   # Valine

}

def calculate\_propensity(residues, structure):

    # Create a DataFrame to store residue counts

    df = pd.DataFrame({'Residue': list(residues), 'Structure': list(structure)})

#N

    N = len(df['Residue'])

    alpha\_conformation\_df = df[df['Structure'] == 'H']

#N\_alpha

    n\_alpha= len(alpha\_conformation\_df['Residue'])

#n\_alpha\_i

    n\_alpha\_i =alpha\_conformation\_df['Residue'].value\_counts()

#N\_alpha\_i

    N\_alpha\_i =df['Residue'].value\_counts()

    for i in amino\_acids:

        if(i in n\_alpha\_i):

            percent\_residue=n\_alpha\_i[i]/N\_alpha\_i[i]

            percent\_all = n\_alpha/N

            prop\_of\_i = percent\_residue/percent\_all

            prop\_dict[i]=prop\_of\_i

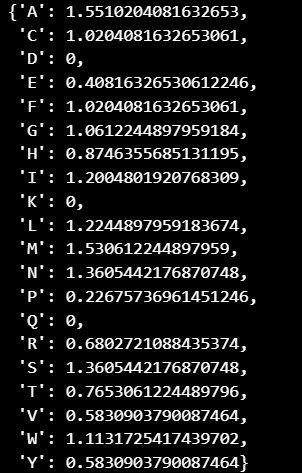
        else: continue

propensities = calculate\_propensity(seq1\_prim, seq1\_sec)

sorted\_prop\_dict = {k: v for k, v in sorted(prop\_dict.items())}

pprint.pprint(sorted\_prop\_dict)

Output :

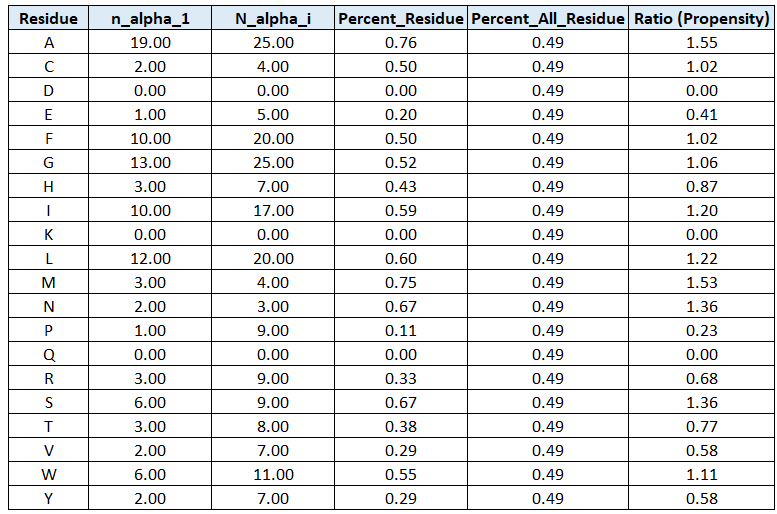


## **Q2) Find the propensity of alpha helices manually for the sequence in question 1**

Total number of AA in the given sequence (nH): 200

Total number of AA which have helical confirmation in the given sequence (N): 98

Denominator value of Propensity formula = 98/200 = 0.49



## **Q3) Using the rules for helices and strands, identify the helical and strand segments in the following sequence**

**KVFGRCELAAAMKRHGLDNYRGYSLGNWVCAAKFESNFNTQATNRNTDGSTDYGILQINSRWWCNDGRTPGSRNLCNIPCSALLSSDITASVNCAKKIVSDGNGMNAWVAWRNRCKGTDVQAWIRGCRL**



🡨--🡪 : Represents helix

EEE.. : Represents Sheet

TTT.. : Represents Turn

**Helix :**

RCELAAAMKR

WVCAAKFESN

IL

LSSDITASVNCAKKIVS

MNAWVA

DV

**Sheet**

YSLGNWV

FNTQATN

TDYGILQI

LCNI

TDV

Turns

K

S

T

R

G

T

S

DG

G

R

S

K

D

N

N

## **Q4) Verify one of the helical and strand segments, manually**

