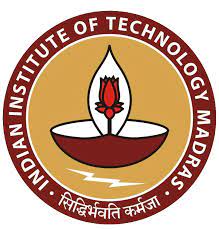
..

**BT 3040: Bioinformatics**

**Assignment 8**



Indian Institute of Technology Madras

Atharva Mandar Phatak | BE21B009

Department of Biotechnology

## **Q1**) **Compute the amino acid composition of the following sequences. Provide the output as a table of amino acid percentage values for each sequence and comment on the results.**

#BT3040 | Assignment 8 | Q1 |Atharva Mandar Phatak | BE21B009

import matplotlib.pyplot as plt

# hydrophobicity values dictionary

hydrophobicity\_values = {

    'A': 13.85, 'D': 11.61, 'C': 15.37, 'E': 11.38, 'F': 13.93,

    'G': 13.34, 'H': 13.82, 'I': 15.28, 'K': 11.58, 'L': 14.13,

    'M': 13.86, 'N': 13.02, 'P': 12.35, 'Q': 12.61, 'R': 13.10,

    'S': 13.39, 'T': 12.70, 'V': 14.56, 'W': 15.48, 'Y': 13.88

}

# Function to calculate hydrophobicity values for a sequence

def calculate\_hydrophobicity(sequence):

    return [hydrophobicity\_values[residue] for residue in sequence]

# List of sequences

sequences = [q1\_seq1,q1\_seq2,q1\_seq3]

# Create subplots for each sequence

fig, axs = plt.subplots(len(sequences), 1, figsize=(10, 5 \* len(sequences)))

# Plot each sequence

for i, seq in enumerate(sequences):

    seq\_val = calculate\_hydrophobicity(seq)

    sequence = [i for i in seq]

    values = seq\_val

    # Scatter plot

    axs[i].scatter(range(len(sequence)), values, marker='o', label='Data Points',c='red')

    # Line plot

    axs[i].plot(range(len(sequence)), values ,label="Hydrophobicity Profile")

    # Customize subplot

    axs[i].set\_title('Hydrophobicity Profile for Sequence {}'.format(i+1))

    axs[i].set\_xlabel('Residue Index')

    axs[i].set\_ylabel('Hydrophobicity Value')

    axs[i].grid(True)

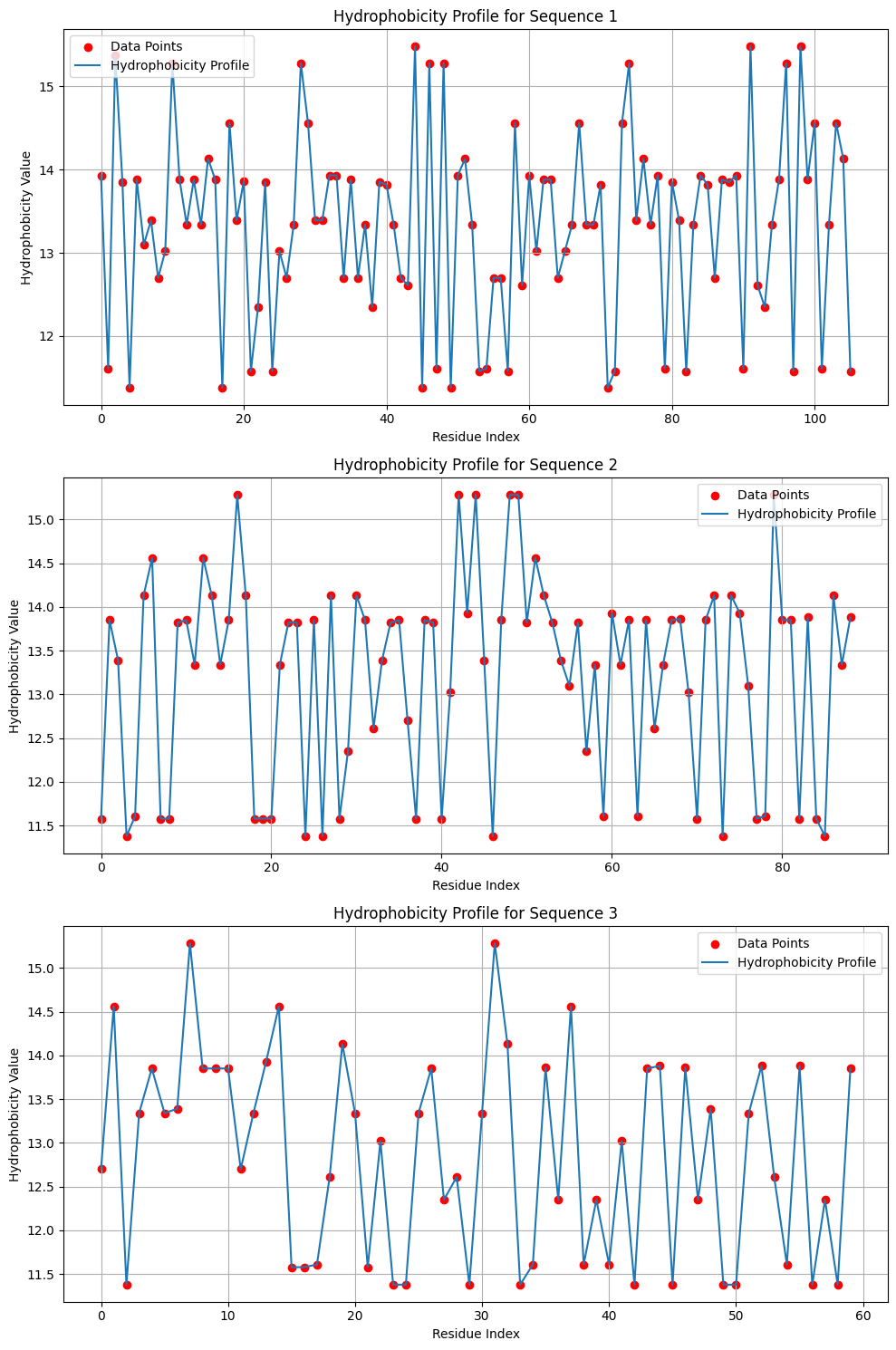
    axs[i].legend()

# Adjust layout

plt.tight\_layout()

plt.show()

Output



Seq1:

Alpha helix= [(5,12), (13,20), (25,32), (33,40), (60,66), (92,99)]

Beta sheet= [(0,5), (44,49)]

### Seq2:

Alpha helix= [(1,8), (9,16), (32,39), (51,58)]

Beta sheet= [(40,45), (82,87)]

Seq3:

Alpha helix= [(3,10), (11,18), (19,26)]

Beta sheet= [(28,33), (40,45)]

## **Q2) Calcualte the amphiphatic index for the helices and strands found in Q1. Use stretch lengths of 8 and 6 for α-helices and β-strands, respectively.**

#BT3040 | Assignment 8 | Q2 |Atharva Mandar Phatak | BE21B009

x1=[i for i in q1\_seq1]

y1=calculate\_hydrophobicity(q1\_seq1)

x2=[i for i in q1\_seq2]

y2=calculate\_hydrophobicity(q1\_seq2)

x3=[i for i in q1\_seq3]

y3=calculate\_hydrophobicity(q1\_seq3)

def amphicity(x, y, A=None, B=None):

    # Beta sheet

    if B:

        for i in B:

            s, e = i

            ad1 = [sum(y[i] for i in range(s, e + 1) if i % 2 == 0)]

            ad2 = [sum(y[i] for i in range(s, e + 1) if i % 2 != 0)]

            re = abs((ad1[0] / 3) - (ad2[0] / 3))

            print(f"Beta sheet {i}: {re:.2f}")

    # Alpha helix

    if A:

        for i in A:

            s, e = i

            ad1 = [sum(y[i] for i in range(s, e + 1) if i % 4 == 0 or i % 4 == 1)]

            ad2 = [sum(y[i] for i in range(s, e + 1) if i % 4 == 2 or i % 4 == 3)]

            re = abs((ad1[0] / 4) - (ad2[0] / 4))

            print(f"Alpha helix {i}: {re:.2f}")

# seq1

A\_1 = [(5, 12), (13, 20), (25, 32), (33, 40), (60, 66), (92, 99)]

B\_1 = [(0, 5), (44, 49)]

# seq2 :

A\_2 = [(1, 8), (9, 16), (32, 39), (51, 58)]

B\_2 = [(40, 45), (82, 87)]

# seq3 :

A\_3 = [(3, 10), (11, 18), (19, 26)]

B\_3 = [(28, 33), (40, 45)]

print("For seq 1:")

amphicity(x1, y1, A\_1, B\_1)

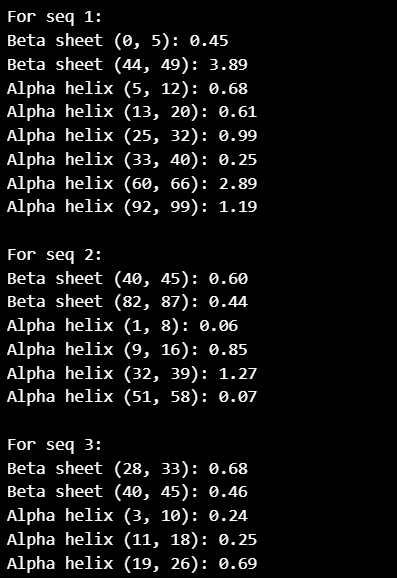
print("\nFor seq 2:")

amphicity(x2, y2, A\_2, B\_2)

print("\nFor seq 3:")

amphicity(x3, y3, A\_3, B\_3)

Output:



## **Q3)** **Plot the hydrophobicity profile for the sequence (Q2.fasta) with window lengths 9 and 19 and list the transmembrane segments**

#BT3040 | Assignment 8 | Q3 |Atharva Mandar Phatak | BE21B009

import matplotlib.pyplot as plt

def my\_h\_r\_plot(seq, w, wid\_len):

    # Create x-axis values

    x = [i for i in range(1, len(seq) + 1)]

    # Create y-axis values

    y = [hydrophobicity\_values[i] for i in seq]

    # Calculate moving average for hydrophobicity values

    yn = []

    for i in range(w):

        yn.append(y[i])

    for i in range(w, len(seq) - w):

        p = (sum(y[i - w:i]) + sum(y[i + 1:i + w + 1])) / (2 \* w)

        yn.append(p)

    for i in range(len(seq) - w, len(seq)):

        yn.append(y[i])

    # Plot the hydrophobicity profile

    plt.figure()

    plt.plot(x, yn, label="Hydrophobicity Profile")

    plt.scatter(x, yn, c="red", label="Data Points")

    plt.grid(True)

    # Set axis labels and title

    plt.xlabel('Amino acid sequence')

    plt.ylabel('Hydrophobicity index')

    plt.title(f'Hydrophobicity Profile for {wid\_len}')

    # Add legend

    plt.legend()

    # Show plot

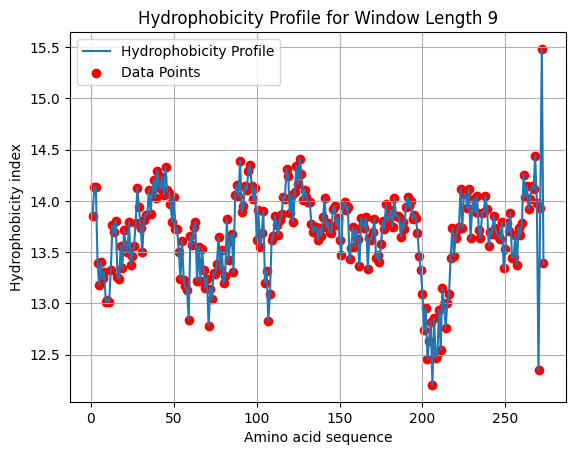
    plt.show()

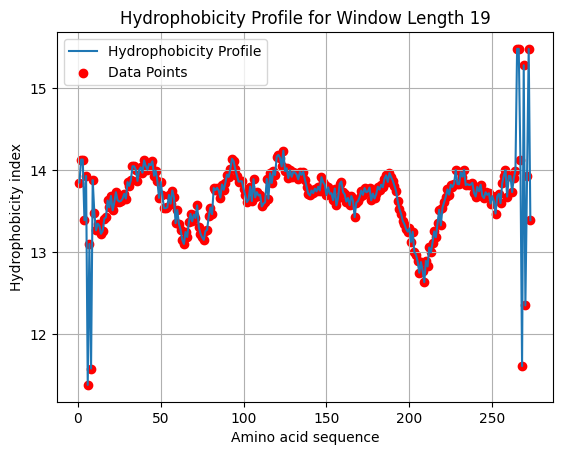
# Plot hydrophobicity profile with window length 4 and 9

my\_h\_r\_plot(q2\_seq1, 4, 'Window Length 9')

my\_h\_r\_plot(q2\_seq1, 9, 'Window Length 19')

Output:

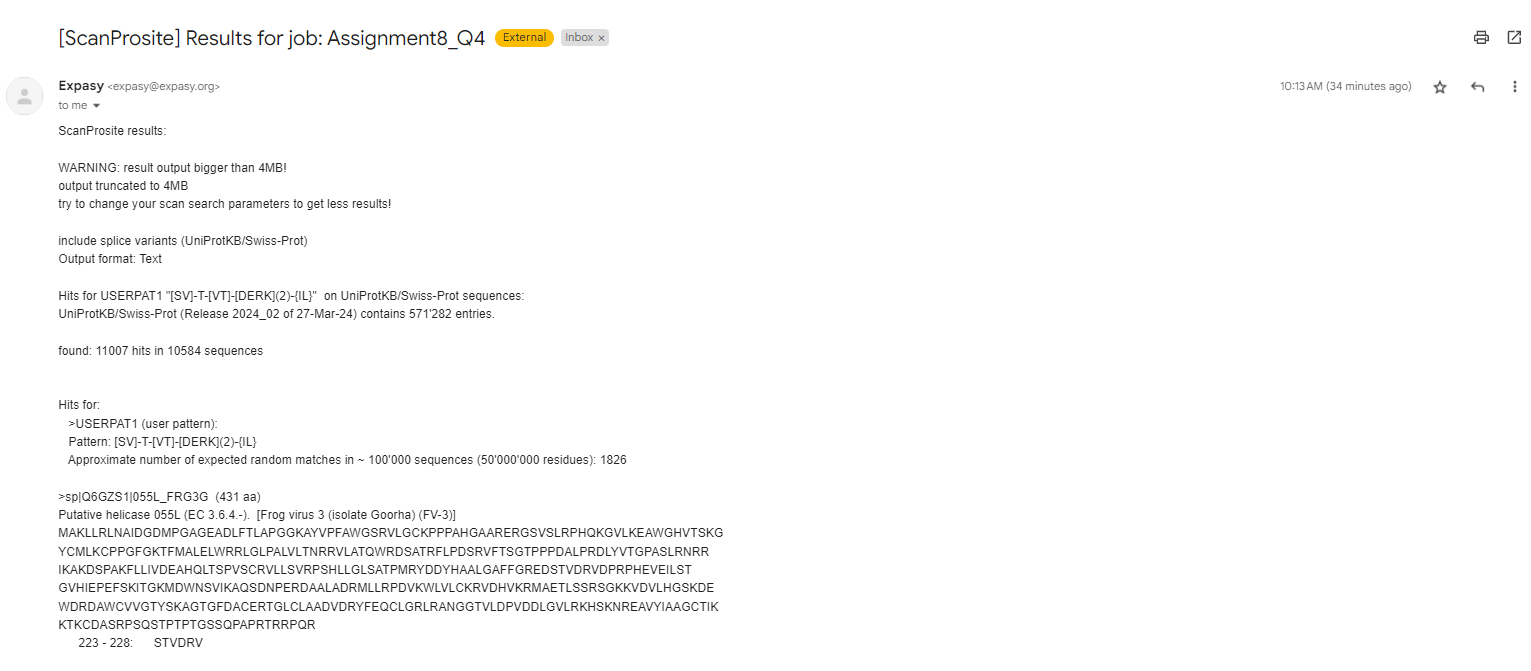




Transmembrane Segments: [9-49], [80-100], [107-193], [216-265]

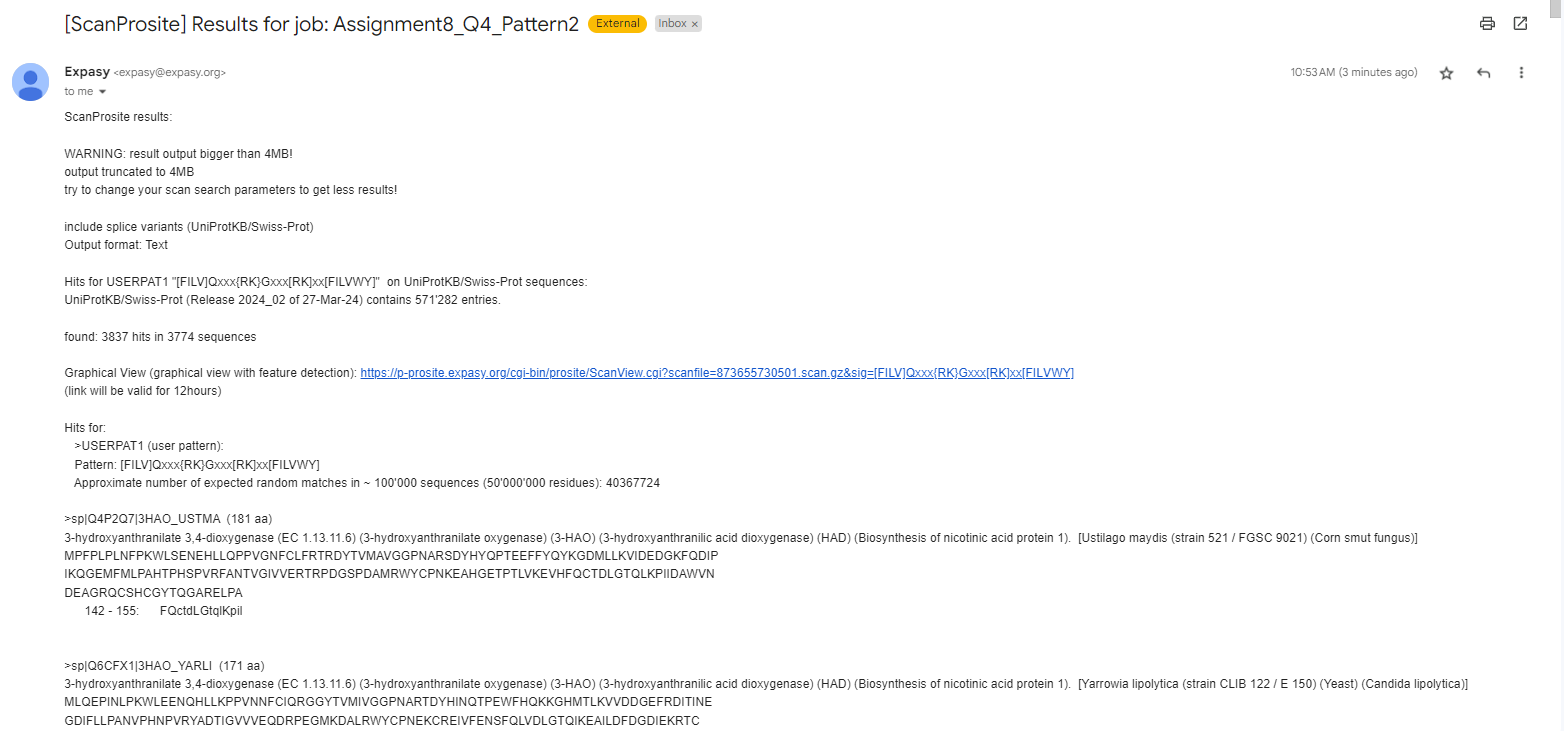
## **Q4) Use ScanProsite tool (https://prosite.expasy.org/scanprosite/ - select option 2), to search for the patterns a) [SV]-T-[VT]-[DERK](2)-{IL} and b) [FILV]Qxxx{RK}Gxxx[RK]xx[FILVWY] in UniProtKB (Include Swiss-Prot, isoforms). List the number of matches for each pattern.**

1. Patten 1:



11007 hits in 10584 sequences

1. Pattern 2



3837 hits in 3774 sequences

Complete output in email. Can forward if required.

## **Q5) Write a program to identify the patterns (refer Q4) in the sequence database (Q4.fasta). List the matches along with the sequence header and location of the matches in the sequence.**

import regex as re

pattern1 = "[SV]T[VT][DERK]{2}[^IL]"

pattern2 = "[FILV]Q...[^RK]G...[RK]..[FILVWY]"

with open("Q4.fasta") as file1:

    sequences = [line.strip() for line in file1]

for i in range(len(sequences)):

    matches1 = re.finditer(pattern1, sequences[i])

    matches2 = re.finditer(pattern2, sequences[i])

    for match in matches1:

        start\_pos = match.start()

        end\_pos = match.end()

        print(f"Pattern 1 matched at position {start\_pos + 1} to {end\_pos} in:")

        print(sequences[i-1])

        print(sequences[i])

        print()

    for match in matches2:

        start\_pos = match.start()

        end\_pos = match.end()

        print(f"Pattern 2 matched at position {start\_pos + 1} to {end\_pos} in:")

        print(sequences[i-1])

        print(sequences[i])

        print()

Output:

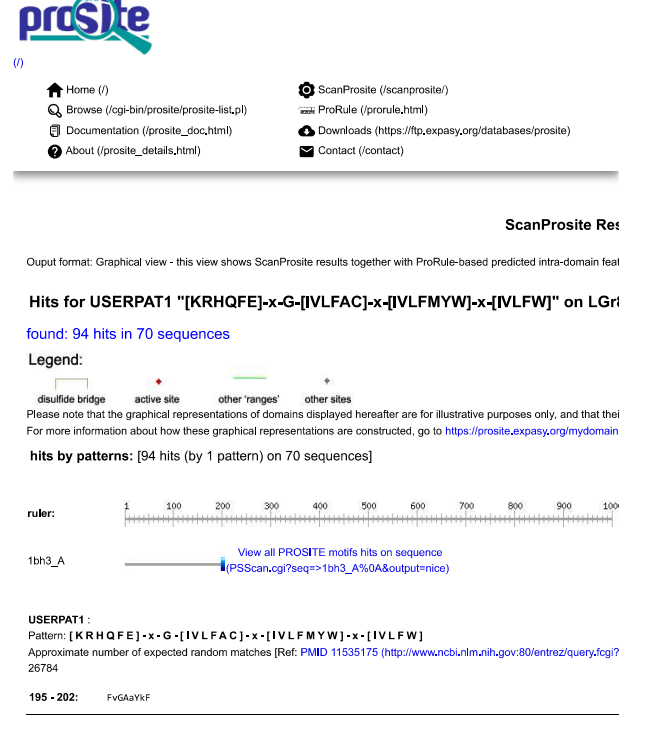
Complete output in ‘Q5\_MatchingSequences.txt’ file

## **Q6) Identify the beta barrel membrane proteins with the following pattern: [K,R,H,Q,F,E]-x-G-[I,V,L,F,A,C]-x-[ I,V,L,F,M,Y,W]-x-[ I,V,L,F,W] Use:** [**http://www.bioinformatics.org/sms2/protein\_pattern.html**](http://www.bioinformatics.org/sms2/protein_pattern.html) **and** [**http://prosite.expasy.org/scanprosite/**](http://prosite.expasy.org/scanprosite/) **Hint: Modify the patterns according to the input format of the server.**

### SMS Output



### Prosite Output



Outputs attached as PDFs, ‘Q6\_SMS\_Output.pdf’ and ‘Q6\_prosite\_match.pdf’