BT 3040: BIOINFORMATICS

Assignment 8

Atharva Mandar Phatak | BE21B009
Department of Biotechnology

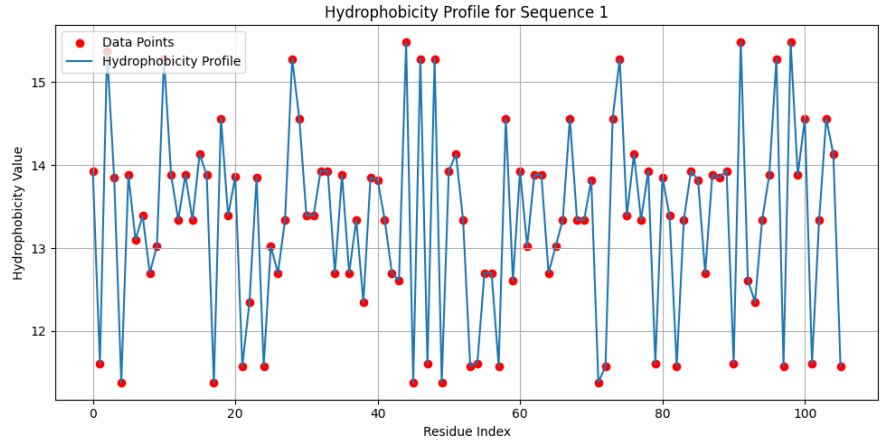
Indian Institute of Technology
Madras

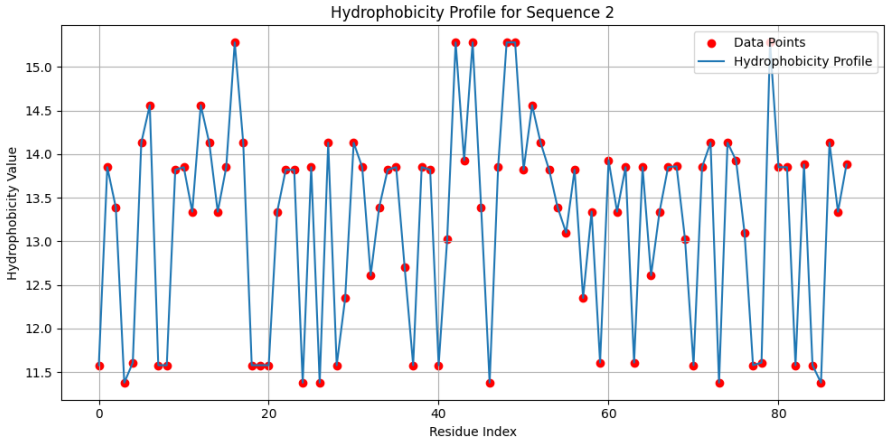
Q1) Compute the amino acid composition of the following sequences. Provide the output as a table of amino acid percentage values for each sequence and comment on the results.

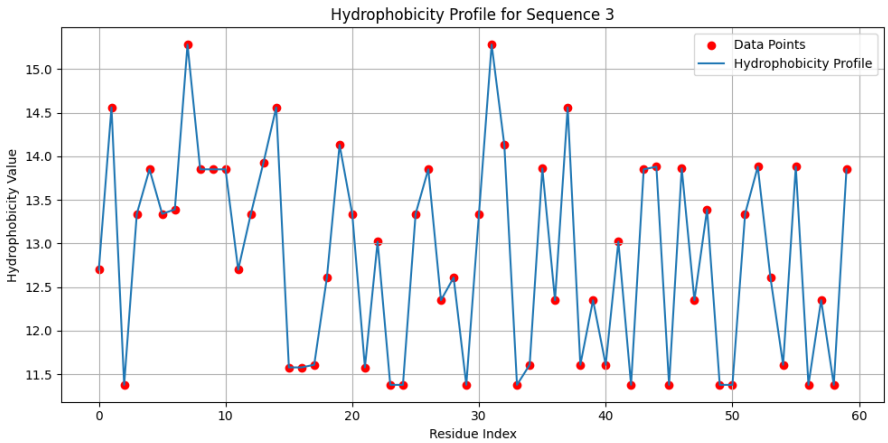
```
#BT3040 | Assignment 8 | Q1 |Atharva Mandar Phatak | BE21B009
import matplotlib.pyplot as plt

# 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)
```

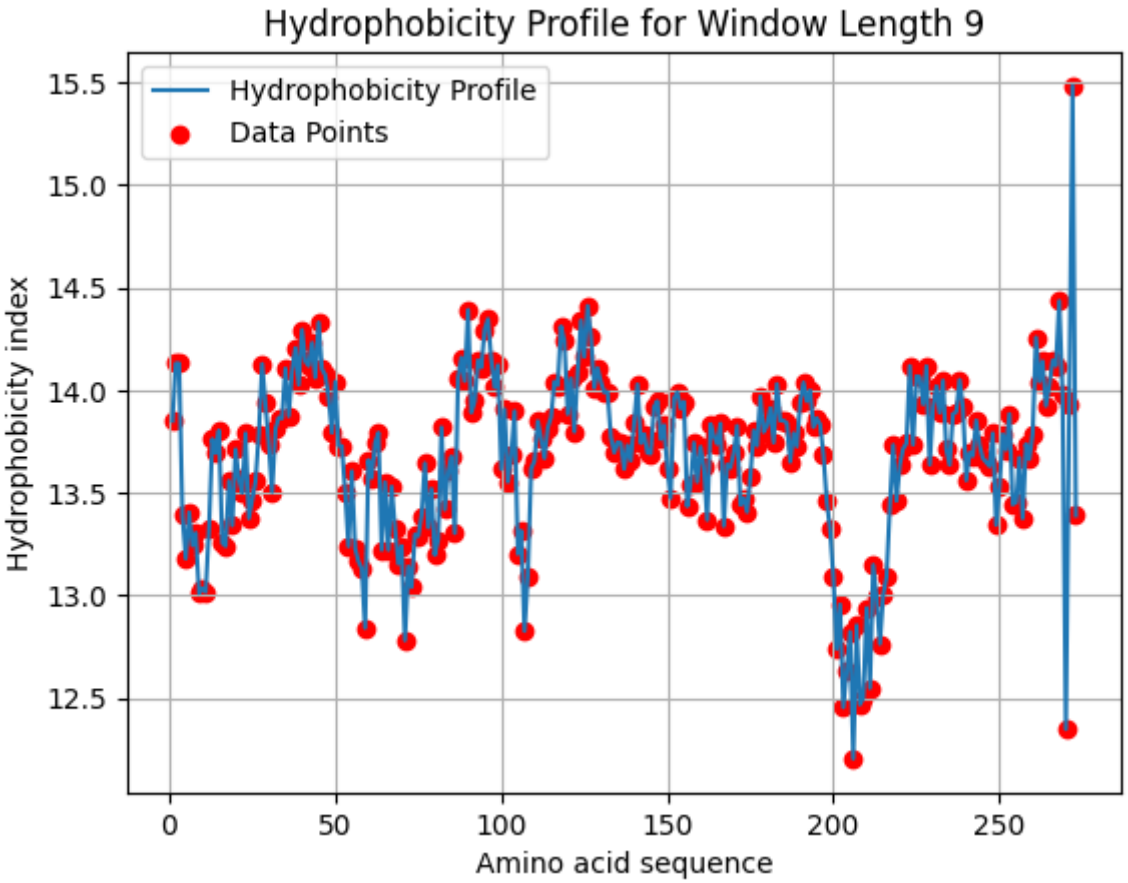
Output:

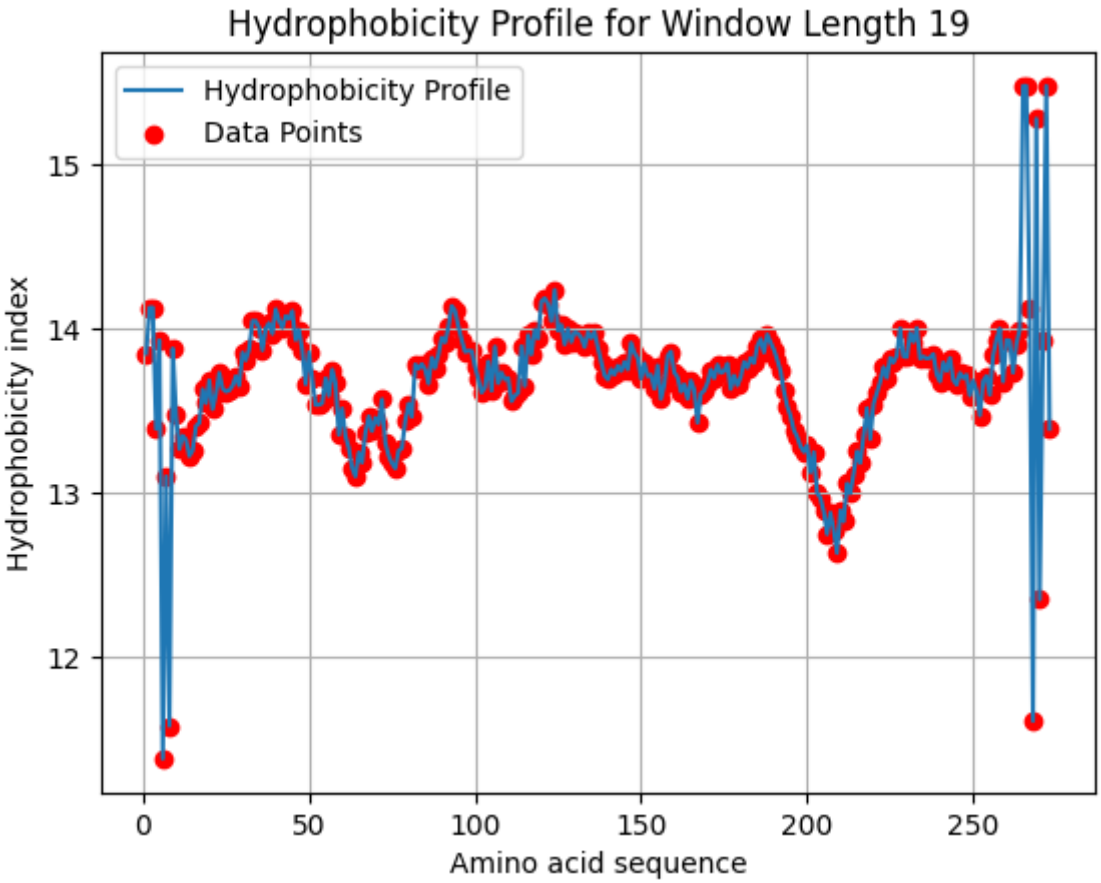
```
For seq 1:
Beta sheet (0, 5): 0.45
Beta sheet (44, 49): 3.89
Alpha helix (5, 12): 0.68
Alpha helix (13, 20): 0.61
Alpha helix (25, 32): 0.99
Alpha helix (33, 40): 0.25
Alpha helix (60, 66): 2.89
Alpha helix (92, 99): 1.19

For seq 2:
Beta sheet (40, 45): 0.60
Beta sheet (82, 87): 0.44
Alpha helix (1, 8): 0.06
Alpha helix (9, 16): 0.85
Alpha helix (32, 39): 1.27
Alpha helix (51, 58): 0.07

For seq 3:
Beta sheet (28, 33): 0.68
Beta sheet (40, 45): 0.46
Alpha helix (3, 10): 0.24
Alpha helix (11, 18): 0.25
Alpha helix (19, 26): 0.69
```

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:

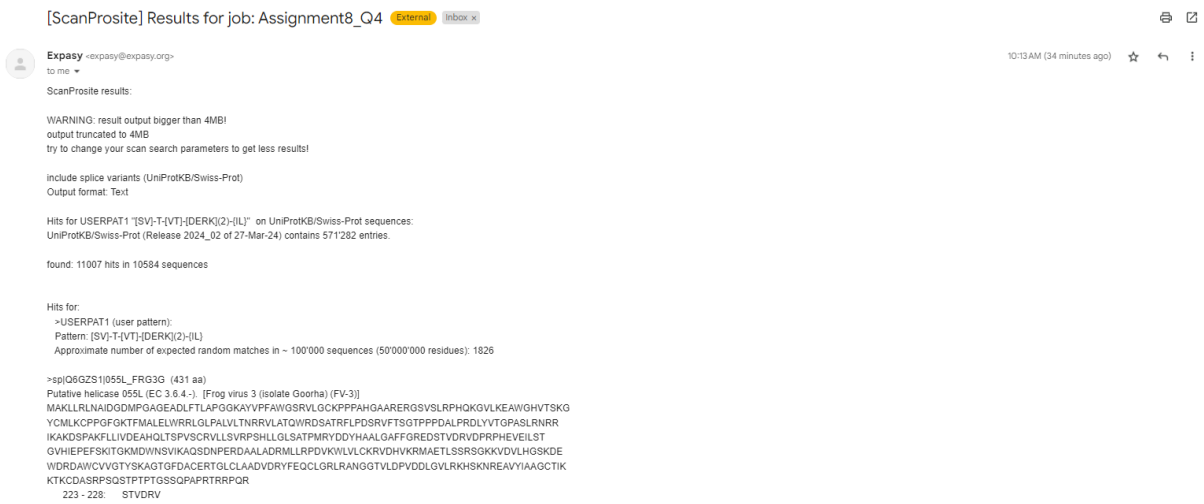

Hydrophobicity Profile for Window Length 9
Hydrophobicity Profile
Data Points
Hydrophobicity index
Amino acid sequence
15.5
15.0
14.5
14.0
13.5
13.0
12.5
0
50
100
150
200
250



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.

a) Patten 1:



[ScanProsite] Results for job: Assignment8_Q4 External Inbox ×

Expasy <expasy@expasy.org> 10:13 AM (34 minutes ago)
to me

ScanProsite results:

WARNING: result output bigger than 4MB!
output truncated to 4MB
try to change your scan search parameters to get less results!

include splice variants (UniProtKB/Swiss-Prot)
Output format: Text

Hits for USERPAT1 "[SV]-T-[VT]-[DERK](2)-{IL}" on UniProtKB/Swiss-Prot sequences:
UniProtKB/Swiss-Prot (Release 2024_02 of 27-Mar-24) contains 571'282 entries.

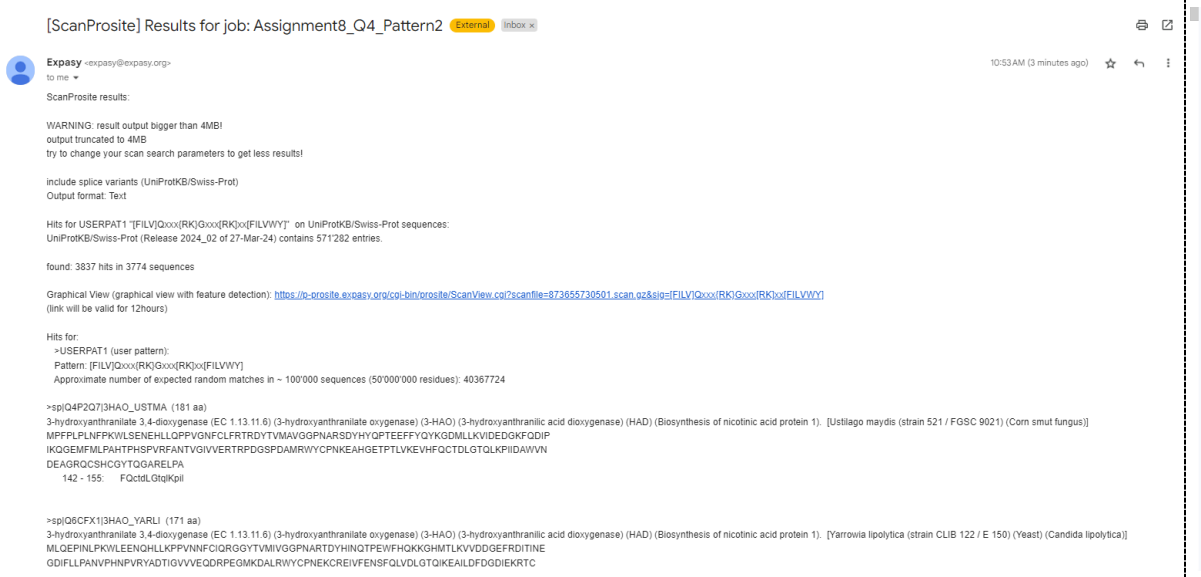
found: 11007 hits in 10584 sequences

Hits for:
>USERPAT1 (user pattern):
Pattern: [SV]-T-[VT]-[DERK](2)-{IL}
Approximate number of expected random matches in ~ 100'000 sequences (50'000'000 residues): 1826

>sp|Q6GZS1|055L_FRG3G (431 aa)
Putative helicase 055L (EC 3.6.4.-). [Frog virus 3 (isolate Goorha) (FV-3)]
MAKLLRLNAIDGDMPGAGEADLFTLAPGGKAYVPFAWGSRVLGCKPPPAHGAARERGSVSLRPHQKGVLKEAWGHVTSKG
YCMLKCPPGFGKTFMALELWRRLGLPALVLTNRRVLATQWRDSATRFLPDSRVFTSGTPPPDALPRDLYVTGPASLRNRR
IKAKDSPAKFLLIVDEAHQLTSPVSCRVLLSVRPSHLLGLSATPMRYDDYHAALGAFFGREDSTVDRVDPRPHEVEILST
GVHIEPEFSKITGKMDWNSVIKAQSDNPERDAALADRMLLRPDVKWLVLCKRVDHVKRMAETLSSRSGKKVDVLHGSKDE
WDRDAWCVVGTYSKAGTGFDACERTGLCLAADVDRYFEQCLGRLRANGGTVLDPVDDLGVLRKHSKNREAVYIAAGCTIK
KTKCDASRPSQSTPTPTGSSQPAPRTRRPQR
223 - 228: STVDRV

11007 hits in 10584 sequences

b) Pattern 2



[ScanProsite] Results for job: Assignment8_Q4_Pattern2 External Inbox ×

Expasy <expasy@expasy.org> 10:53 AM (3 minutes ago)
to me

ScanProsite results:

WARNING: result output bigger than 4MB!
output truncated to 4MB
try to change your scan search parameters to get less results!

include splice variants (UniProtKB/Swiss-Prot)
Output format: Text

Hits for USERPAT1 "[FILV]Qxxx{RK}Gxxx[RK]xx[FILVWY]" on UniProtKB/Swiss-Prot sequences:
UniProtKB/Swiss-Prot (Release 2024_02 of 27-Mar-24) contains 571'282 entries.

found: 3837 hits in 3774 sequences

Graphical View (graphical view with feature detection): https://p-prosite.expasy.org/cgi-bin/prosite/ScanView.cgi?scanfile=873655730501.scan.gz&sig=[FILV]Qxxx{RK}Gxxx[RK]xx[FILVWY]
(link will be valid for 12hours)

Hits for:
>USERPAT1 (user pattern):
Pattern: [FILV]Qxxx{RK}Gxxx[RK]xx[FILVWY]
Approximate number of expected random matches in ~ 100'000 sequences (50'000'000 residues): 40367724

>sp|Q4P2Q7|3HAO_USTMA (181 aa)
3-hydroxyanthranilate 3,4-dioxygenase (EC 1.13.11.6) (3-hydroxyanthranilate oxygenase) (3-HAO) (3-hydroxyanthranilic acid dioxygenase) (HAD) (Biosynthesis of nicotinic acid protein 1). [Ustilago maydis (strain 521 / FGSC 9021) (Corn smut fungus)]
MPFPLPLNFPKWLSENEHLLQPPVGNFCLFRTRDYTVMAVGGPNARSDYHYQPTEEFFYQYKGDMLLKVIDEDGKFQDIP
IKQGEMFMLPAHTPHSPVRFANTVGIVVERTRPDGSPDAMRWYCPNKEAHGETPTLVKEVHFQCTDLGTQLKPIIDAWVN
DEAGRQCSHCGYTQGARELPA
142 - 155: FQctdLGtqlKpil

>sp|Q6CFX1|3HAO_YARLI (171 aa)
3-hydroxyanthranilate 3,4-dioxygenase (EC 1.13.11.6) (3-hydroxyanthranilate oxygenase) (3-HAO) (3-hydroxyanthranilic acid dioxygenase) (HAD) (Biosynthesis of nicotinic acid protein 1). [Yarrowia lipolytica (strain CLIB 122 / E 150) (Yeast) (Candida lipolytica)]
MLQEPINLPKWLEENQHLLKPPVNNFCIQRGGYTVMIVGGPNARTDYHINQTPEWFHQKKGHMTLKVVDDGEFRDITINE
GDIFLLPANVPHNPVRYADTIGVVVEQDRPEGMKDALRWYCPNEKCREIVFENSFQLVDLGTQIKEAILDFDGDIEKRTC

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:

```
Pattern 1 matched at position 665 to 670 in:
>4A0C_2|Chains C,E|CULLIN-4B|HOMO SAPIENS (9606)
MHHHHHVDENLYFQGGGRGSAKKLVIKNFKDKPKLPENYTDETWQKLKEAVEAIQNSTSIKYNLEELYQAVENLCSYKISANLYKQLRQICEDHIKAQIHQFREDSLDSVLFLKKIDRCWQNHCRQMIMIRSIFLFLDRTYVLQNSMLPSIWDMGLELFRAHIISDQKVQNKTIDGILLLIERI

Pattern 1 matched at position 665 to 670 in:
>4A0C_2|Chains C,E|CULLIN-4B|HOMO SAPIENS (9606)
MHHHHHVDENLYFQGGGRGSAKKLVIKNFKDKPKLPENYTDETWQKLKEAVEAIQNSTSIKYNLEELYQAVENLCSYKISANLYKQLRQICEDHIKAQIHQFREDSLDSVLFLKKIDRCWQNHCRQMIMIRSIFLFLDRTYVLQNSMLPSIWDMGLELFRAHIISDQKVQNKTIDGILLLIERI

Pattern 1 matched at position 666 to 671 in:
>4A0K_1|Chain A|CULLIN-4A|HOMO SAPIENS (9606)
MHHHHHVDEENLYFQGGGRGGSKKLVIKNFRDRPRLPDNYTQDTWRKLHEAVRAVQSSTSIRYNLEELYQAVENLCSHKVSPMLYKQLRQACEDHVQAQILPFREDSLDSVLFLKKINTCWQDHCRQMIMIRSIFLFLDRTYVLQNSTLPSIWDMGLELFRTHIISDKMVQSKTIDGILLLIE

Pattern 2 matched at position 252 to 265 in:
>4FXG_2|Chains B,E|Complement C4-A alpha chain|Homo sapiens (9606)
NVNFQKAINEKLGQYASPTAKRCCQDGVTRLPMMRSCEQRAARVQQPDCREPFLSCCQFAESLRKKSRDKGQAGLQRALEILQEEDLIDEDDIPVRSFFPENWLWRVETVDRFQILTLWLPDSLTTWEIHGLSLSKTKGLCVATPVQLRVFREFHLHLRLPMSVRRFEQLELRPVLYNYLDKNL

Pattern 2 matched at position 252 to 265 in:
>4FXG_2|Chains B,E|Complement C4-A alpha chain|Homo sapiens (9606)
NVNFQKAINEKLGQYASPTAKRCCQDGVTRLPMMRSCEQRAARVQQPDCREPFLSCCQFAESLRKKSRDKGQAGLQRALEILQEEDLIDEDDIPVRSFFPENWLWRVETVDRFQILTLWLPDSLTTWEIHGLSLSKTKGLCVATPVQLRVFREFHLHLRLPMSVRRFEQLELRPVLYNYLDKNL

Pattern 2 matched at position 252 to 265 in:
>4FXK_2|Chain B|Complement C4-A Alpha chain|Homo sapiens (9606)
NVNFQKAINEKLGQYASPTAKRCCQDGVTRLPMMRSCEQRAARVQQPDCREPFLSCCQFAESLRKKSRDKGQAGLQRALEILQEEDLIDEDDIPVRSFFPENWLWRVETVDRFQILTLWLPDSLTTWEIHGLSLSKTKGLCVATPVQLRVFREFHLHLRLPMSVRRFEQLELRPVLYNYLDKNL
```

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 and http://prosite.expasy.org/scanprosite/ Hint: Modify the patterns according to the input format of the server.

1) SMS Output

Protein Pattern Find results

Results for 284 residue sequence "3b07_A" starting "VTLYKTTATA"

no matches found for this sequence.

Results for 270 residue sequence "3b07_B" starting "AEIIKRTQDI"

no matches found for this sequence.

Results for 60 residue sequence "8b4i_C" starting "WFYHKYSTTT"

no matches found for this sequence.

Results for 38 residue sequence "8b4i_E" starting "ALSREELQAA"

no matches found for this sequence.

Results for 51 residue sequence "8b4i_I" starting "ALSEESKERI"

no matches found for this sequence.

Results for 211 residue sequence "7cbl_A" starting "CAWIPAKPLV"

no matches found for this sequence.

Results for 303 residue sequence "7cbl_a" starting "ERIRDLTSVQ"

no matches found for this sequence.

2) Prosite Output

(/)

Home (/) | ScanProsite (/scanprosite/)
Browse (/cgi-bin/prosite/prosite-list.pl) | ProRule (/prorule.html)
Documentation (/prosite_doc.html) | Downloads (https://ftp.expasy.org/databases/prosite)
About (/prosite_details.html) | Contact (/contact)

ScanProsite Res

Ouput format: Graphical view - this view shows ScanProsite results together with ProRule-based predicted intra-domain feat

Hits for USERPAT1 "[KRHQFE]-x-G-[IVLFAC]-x-[IVLFMYW]-x-[IVLFW]" on LGr

found: 94 hits in 70 sequences

Legend:

disulfide bridge | active site | other 'ranges' | other sites

Please note that the graphical representations of domains displayed hereafter are for illustrative purposes only, and that thei
For more information about how these graphical representations are constructed, go to https://prosite.expasy.org/mydomain

hits by patterns: [94 hits (by 1 pattern) on 70 sequences]

ruler: 1 100 200 300 400 500 600 700 800 900 100

1bh3_A — View all PROSITE motifs hits on sequence (PSScan.cgi?seq=>1bh3_A%0A&output=nice)

USERPAT1 :
Pattern: **[KRHQFE]-x-G-[IVLFAC]-x-[IVLFMYW]-x-[IVLFW]**
Approximate number of expected random matches [Ref: PMID 11535175 (http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?
26784

195 - 202: FvGAaYkF

Outputs attached as PDFs, 'Q6_SMS_Output.pdf' and 'Q6_prosite_match.pdf'