Practical 11

## Date: 10-04-2022 Name: Aswin. V

## Day: Sunday Roll Number: BS19B003

**Q1 –**

def Q1(A, B):  
 AA = ['A', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'K', 'L',  
 'M', 'N', 'P', 'Q', 'R', 'S', 'T', 'V', 'W', 'Y']  
 c, h = [0]\*20, [0]\*20  
 p = [0]\*20  
  
 for i in range(len(A)):  
 c[AA.index(A[i])] += 1  
 if B[i] == 'H':  
 h[AA.index(A[i])] += 1  
  
 print('Propensity:\n')  
 for i in range(20):  
 if c[i] != 0:  
 p[i] = (h[i]/c[i])/(B.count('H')/len(A))  
 print(f"{AA[i]}\t:\t{p[i]}")  
  
  
A = 'LGASGIAAFAFGSTAILIILFNMAAEVHFDPLQFFRQFFWLGLYPPKAQYGMGIPPLHDGGWWLMAGLFMTLSLGSWWIRVYSRARALGLGTHIAWNFAAAIFFVLCIGCIHPTLVGSWSEGVPFGIWPHIDWLTAFSIRYGNFYYCPWHGFSIGFAYGCGLLFAAHGATILAVARFGGDREIEQITDRGTAVERAALFW'  
B = 'XHHHHHHHHHHHHHHHHHHHHHHHHHXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHXXHHHHHHHHHHHHHHHHHXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXHHHHHHHHHHHHHHHHHHHHHHHHHHXXXXXXXXXXXXXXXXXXXXXXXXXXX'  
Q1(A, B)

Output:

Propensity:

A : 1.5510204081632653

C : 1.0204081632653061

D : 0.0

E : 0.40816326530612246

F : 1.0204081632653061

G : 1.0612244897959184

H : 0.8746355685131195

I : 1.2004801920768309

K : 0.0

L : 1.2244897959183674

M : 1.530612244897959

N : 1.3605442176870748

P : 0.22675736961451246

Q : 0.0

R : 0.6802721088435374

S : 1.3605442176870748

T : 0.7653061224489796

V : 0.5830903790087464

W : 1.1131725417439702

Y : 0.5830903790087464

**Q2 –**

Table

Description automatically generated with low confidence

**Q3 –**

def Q2\_sub(a, b):  
 m = ''  
 for i in range(len(a)):  
 for j in range(len(b)):  
 d = 1  
 while i+d <= len(a) and j+d <= len(b) and a[i:i+d] == b[j:j+d]:  
 if len(m) <= len(a[i:i+d]):  
 m = a[i:i+d]  
 d += 1  
 return m  
  
def Q2(A):  
 helix = {'A': 'Ha', 'C': 'ia', 'D': 'ia', 'E': 'Ha', 'F': 'ha',  
 'G': 'Ba', 'H': 'ha', 'I': 'Ia', 'K': 'Ia', 'L': 'Ha',  
 'M': 'ha', 'N': 'ba', 'P': 'Ba', 'Q': 'ha', 'R': 'ia',  
 'S': 'ia', 'T': 'ia', 'V': 'ha', 'W': 'ha', 'Y': 'ba'}  
  
 strand = {'A': 'Ib', 'C': 'hb', 'D': 'ib', 'E': 'Bb', 'F': 'hb',  
 'G': 'ib', 'H': 'bb', 'I': 'Hb', 'K': 'bb', 'L': 'hb',  
 'M': 'Hb', 'N': 'bb', 'P': 'bb', 'Q': 'hb', 'R': 'ib',  
 'S': 'bb', 'T': 'hb', 'V': 'Hb', 'W': 'hb', 'Y': 'hb'}  
  
 ph = {'A': 1.45, 'C': 0.77, 'D': 0.98, 'E': 1.53, 'F': 1.12,  
 'G': 0.53, 'H': 1.24, 'I': 1.00, 'K': 1.07, 'L': 1.34,  
 'M': 1.20, 'N': 0.73, 'P': 0.59, 'Q': 1.17, 'R': 0.79,  
 'S': 0.79, 'T': 0.82, 'V': 1.14, 'W': 1.14, 'Y': 0.61}  
  
 pb = {'A': 0.97, 'C': 1.30, 'D': 0.80, 'E': 0.26, 'F': 1.28,  
 'G': 0.81, 'H': 0.71, 'I': 1.60, 'K': 0.74, 'L': 1.22,  
 'M': 1.67, 'N': 0.65, 'P': 0.62, 'Q': 1.23, 'R': 0.90,  
 'S': 0.72, 'T': 1.20, 'V': 1.65, 'W': 1.19, 'Y': 1.29}  
  
 cf = {'Ha': 1, 'ha': 1, 'Ia': 0.5, 'ia': 0, 'ba': -1, 'Ba': -1,  
 'Hb': 1, 'hb': 1, 'Ib': 0.5, 'ib': 0, 'bb': -1, 'Bb': -1}  
  
 hs = []  
 ss = []  
  
 print('\nAlpha Helices:')  
 i = 0  
 while i < len(A) - 6:  
 value = 0  
 for j in range(6):  
 value += cf[helix[A[i:i+6][j]]]  
 if value >= 4:  
 done = 1  
 k = 0  
 while done == 1:  
 next\_seg = A[i+k+2:i+k+6]  
 p = 0  
 for l in range(4):  
 p += ph[next\_seg[l]]  
 if p < 4.00:  
 done = 0  
 else:  
 k += 1  
 if k == 0:  
 print(A[i:i + k + 6])  
 hs.append(A[i:i + k + 6])  
 i = i + k + 6  
 else:  
 print(A[i:i + k + 5])  
 hs.append(A[i:i + k + 5])  
 i = i + k + 5  
 else:  
 i += 1  
   
 print('\nBeta Strands')  
 i1 = 0  
 while i1 < len(A) - 5:  
 value = 0  
 for j in range(5):  
 value += cf[strand[A[i1:i1+5][j]]]  
 if value >= 3:  
 done = 1  
 k = 0  
 while done == 1 and (i1+k+5) <= len(A):  
 next\_seg = A[i1+k+2:i1+k+5]  
 prop = 0  
 for l in range(3):  
 prop += pb[next\_seg[l]]  
 if prop < 3.00:  
 done = 0  
 else:  
 k += 1  
 if k == 0:  
 print(A[i1:i1 + k + 5])  
 ss.append(A[i1:i1 + k + 5])  
 i1 = i1 + k + 5  
 else:  
 print(A[i1:i1 + k + 4])  
 ss.append(A[i1:i1 + k + 4])  
 i1 = i1 + k + 4  
 else:  
 i1 += 1  
  
 hf = []  
 sf = []  
 print('\nCommon segments ')  
 for i in range(len(hs)):  
 h = hs[i]  
 for j in range(len(ss)):  
 s = ss[j]  
 c = Q2\_sub(h, s)  
 m = len(c)  
 if m != 0 and m >= 5:  
 print('Helix - %s, Strand - %s, Common segment - %s' % (h, s, c))  
 prop\_helix = 0  
 prop\_strand = 0  
 for k in range(m):  
 prop\_helix += ph[c[k]]  
 prop\_strand += pb[c[k]]  
 if prop\_helix > prop\_strand:  
 hf.append([j, h])  
 else:  
 sf.append([i, s])  
 for i in range(len(hf) - 1, -1, -1):  
 a = ss[hf[i][0]]  
 ss.remove(a)  
 for j in range(len(sf) - 1, -1, -1):  
 b = hs[sf[j][0]]  
 hs.remove(b)  
  
  
 print('\nFinal list of secondary structure segments after comparing -')  
 print('\nAlpha Helix segments')  
 for i in range(len(hs)):  
 print(hs[i])  
 print('\nBeta Strand segments')  
 for i in range(len(ss)):  
 print(ss[i])  
  
  
  
A = "KVFGRCELAAAMKRHGLDNYRGYSLGNWVCAAKFESNFNTQATNRNTDGSTDYGILQINSRWWCNDGRTPGSRNLCNIPCSALLSSDITASVNC"  
Q2(A)

Output:

Alpha helices:

RCELAAAMKRH

WVCAAKFESNF

MNAWVAWRN

TDVQAWIR

Beta strands:

VFGRC

LAAAMKR

WVCAA

TDYGILQIN

AWVAWR

GTDVQAWIRGCRL

Common segments - Conflicting sequences

Helix - RCELAAAMKRH, Strand - LAAAMKR, Common segment - LAAAMKR

Helix - WVCAAKFESNF, Strand - WVCAA, Common segment - WVCAA

Helix - MNAWVAWRN, Strand - AWVAWR, Common segment - AWVAWR

Helix - TDVQAWIR, Strand - GTDVQAWIRGCRL, Common segment - TDVQAWIR

Final list of secondary structure segments -

Alpha Helix segments

RCELAAAMKRH

MNAWVAWRN

Beta Strand segments

VFGRC

WVCAA

TDYGILQIN

GTDVQAWIRGCRL

**Q4 –**

Text, letter

Description automatically generated