***Practicals 1***

*-BS19B032*

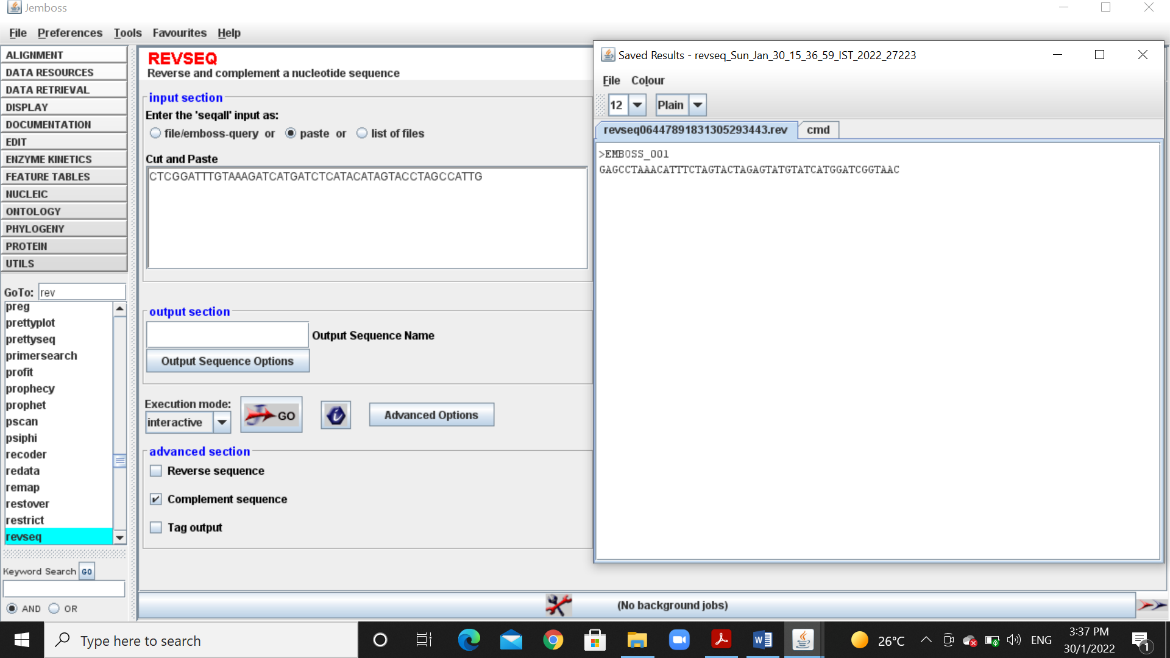
*-R. Vasantha Kumar*

*1) I have downloaded emboss in my computer, as per the demo shown in the class.*

*2) The complementary strand for the given sequence is:*

*GAGCCTAAACATTTCTAGTACTAGAGTATGTATCATGGATCGGTAAC*

*(\*not reversed).*

******

*3) This is a python program that is used to find the complementary strand of given sequence:*

*The output is:*

seq=input("Enter your DNA sequence:") #taking input sequence

comp\_seq=""

for i in range(len(seq)):

if(seq[i]=='A'):

comp\_seq+='T' #creating complementary sequence

if(seq[i]=='T'):

comp\_seq+='A'

if(seq[i]=='C'): #A to T; T to A; C to G; G to C

comp\_seq+='G'

if(seq[i]=='G'):

comp\_seq+='C'

print(comp\_seq)

*GAGCCTAAACATTTCTAGTACTAGAGTATGTATCATGGATCGGTAAC*

*4****a****) Using EMBOSS the protein sequence for given DNA sequence is:*

*Reading Frame 1:*

*DIVNSKKVHAMRKEQKRKQGKQRSMGSPMDYSPLPIDKHEPEFGPCRRKLDG*

*Reading Frame 2:*

*TL\*TVKKSMQCARSRRGSRASSAPWALPWTTLLCPSTSMSLNLVHAEENWMG*

*Reading Frame 3:*

*HCEQ\*KSPCNAQGAEEEAGQAALHGLSHGLLSSAHRQA\*A\*IWSMQKKTGWX*

*Reading Frame -1:*

*PIQFSSAWTKFRLMLVDGQRRVVHGRAHGALLALLPLLLLAHCMDFFTVHNV*

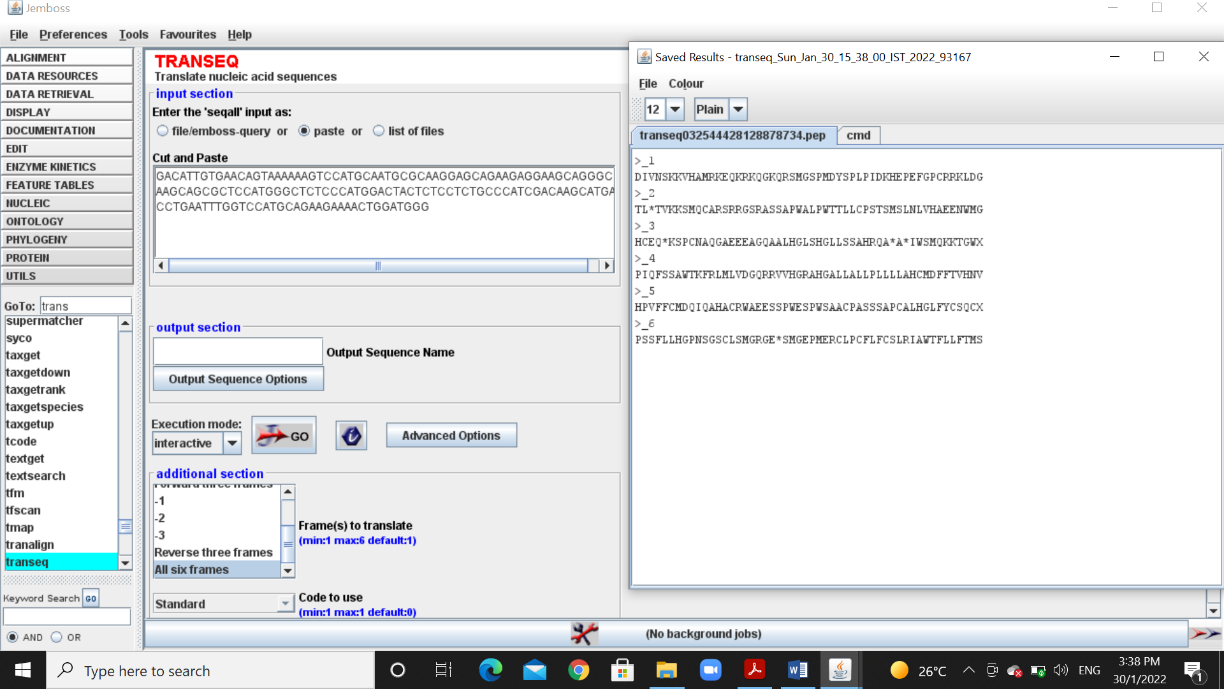
*Reading Frame -2:*

*HPVFFCMDQIQAHACRWAEESSPWESPWSAACPASSSAPCALHGLFYCSQCX*

*Reading Frame -3:*

*PSSFLLHGPNSGSCLSMGRGE\*SMGEPMERCLPCFLFCSLRIAWTFLLFTMS*

*4****b****) By checking with the above sequences we got, the reading frame corresponding to given sequence is:* ***-1.***

**

*5) This is a python program to translate given DNA sequence into protein sequence:*

*seq=input("Enter your DNA sequence:")*

*mrna\_seq=""*

*protein=""*

*for i in range(len(seq)):*

*if(seq[i]=='T'):*

*mrna\_seq+='U'*

*else:*

*mrna\_seq+=seq[i]*

*length=0*

*if(len(mrna\_seq)%3==0):*

*length=len(mrna\_seq)*

*elif(len(mrna\_seq)%3==1):*

*length=len(mrna\_seq)-1*

*else:*

*length=len(mrna\_seq)-2*

*for i in range(0,length,3):*

*if(mrna\_seq[i]=='A'):*

*if(mrna\_seq[i+1]=='A'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='K'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='N'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='N'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='K'*

*if(mrna\_seq[i+1]=='U'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='I'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='I'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='I'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='M'*

*if(mrna\_seq[i+1]=='C'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='T'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='T'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='T'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='T'*

*if(mrna\_seq[i+1]=='G'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='R'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='S'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='S'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='R'*

*if(mrna\_seq[i]=='U'):*

*if(mrna\_seq[i+1]=='A'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='\*'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='Y'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='Y'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='\*'*

*if(mrna\_seq[i+1]=='U'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='J'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='F'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='F'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='J'*

*if(mrna\_seq[i+1]=='C'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='S'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='S'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='S'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='S'*

*if(mrna\_seq[i+1]=='G'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='\*'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='C'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='C'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='W'*

*if(mrna\_seq[i]=='C'):*

*if(mrna\_seq[i+1]=='A'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='Q'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='H'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='H'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='Q'*

*if(mrna\_seq[i+1]=='U'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='L'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='L'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='L'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='L'*

*if(mrna\_seq[i+1]=='C'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='P'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='P'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='P'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='P'*

*if(mrna\_seq[i+1]=='G'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='R'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='R'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='R'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='R'*

*if(mrna\_seq[i]=='G'):*

*if(mrna\_seq[i+1]=='A'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='E'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='D'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='D'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='E'*

*if(mrna\_seq[i+1]=='U'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='V'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='V'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='V'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='V'*

*if(mrna\_seq[i+1]=='C'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='A'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='A'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='A'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='A'*

*if(mrna\_seq[i+1]=='G'):*

*if(mrna\_seq[i+2]=='A'):*

*protein+='G'*

*if(mrna\_seq[i+2]=='U'):*

*protein+='G'*

*if(mrna\_seq[i+2]=='C'):*

*protein+='G'*

*if(mrna\_seq[i+2]=='G'):*

*protein+='G'*

*print(protein)*

*The protein sequence I got is:*

*DIVNSKKVHAMRKEQKRKQGKQRSMGSPMDYSPLPIDKHEPEFGPCRRKLDG*

*6) The python code to find matches in the string is:*

*seq=input("Enter your sequence:")*

*check\_str=input("Enter the string to search:")*

*matches=0*

*for i in range(len(seq)-len(check\_str)+1):*

*count=0*

*for j in range(len(check\_str)):*

*if(seq[i+j]==check\_str[j]):*

*count=count+1*

*if(count==len(check\_str)):*

*matches=matches+1*

*print("Match location:",i)*

*print("Total matches:",matches)*

*The output of the code when we input string in question 4 and the match string as* ***AAG:***

*Match location: 19*

*Match location: 36*

*Match location: 45*

*Match location: 51*

*Match location: 60*

*Match location: 111*

*Match location: 140*

*Total matches: 7*

*The output of the code when we input string in question 4 and the match string as* ***ACTA:***

*Match location: 88*

*Total matches: 1*

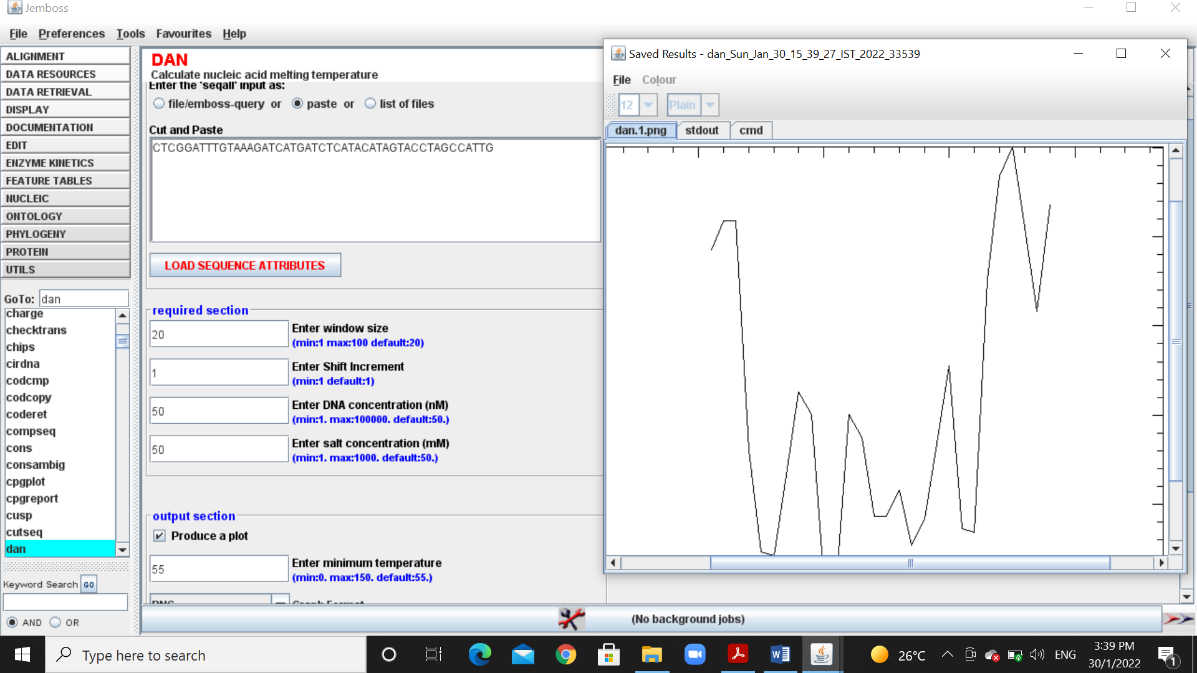
*By running the code with strings of different length, I observed that as string length increases, matches decreases.*

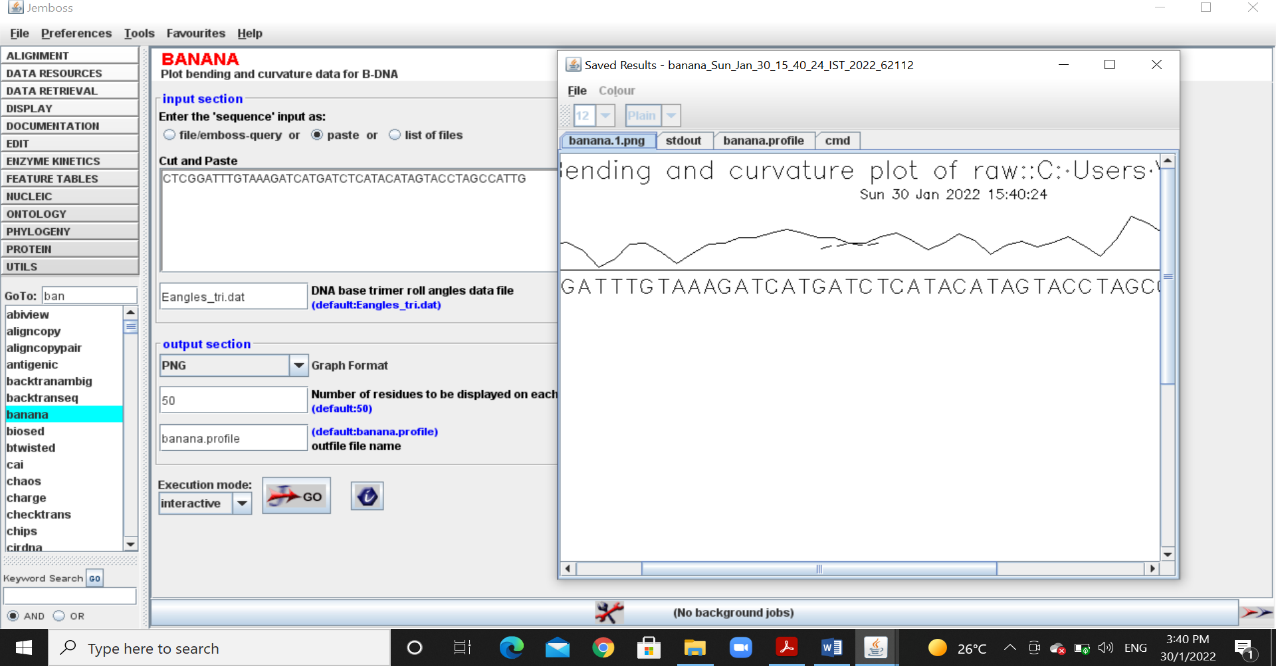
*7) I learnt some of these applications of emboss.*

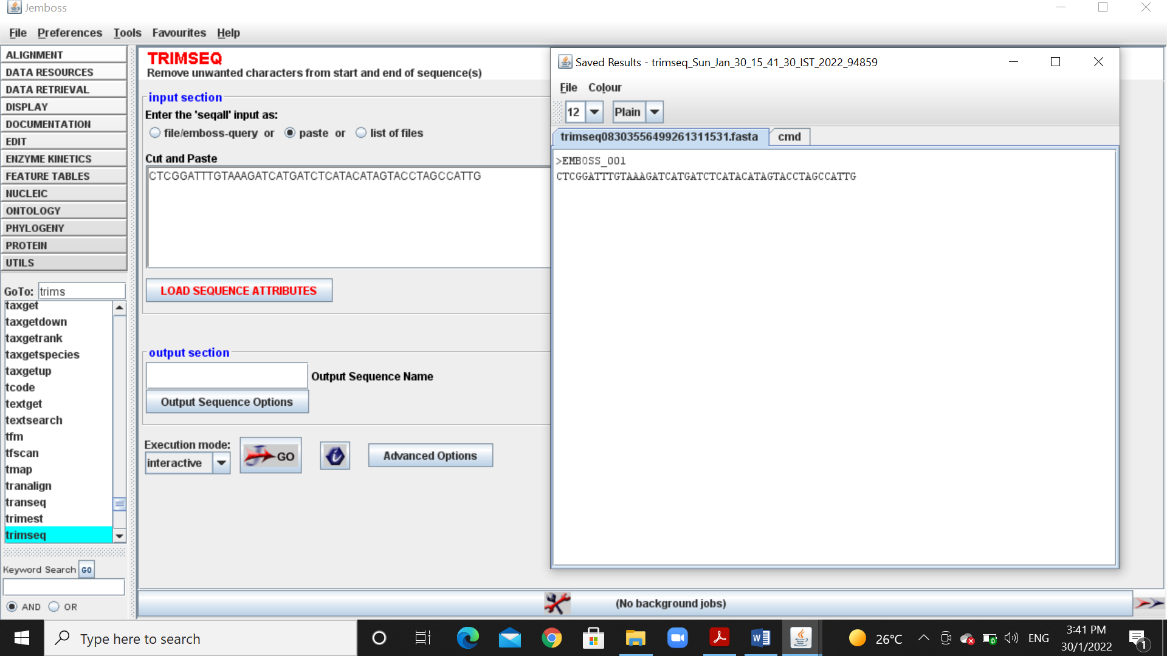
***dan****: This tool in emboss is used to find the melting temperature of the given nucleotide sequence. This tool also provides us with plot on melting temperature.*

***Banana****: This tool in emboss is used to plot bending and curvature of given DNA sequence.*

***trimseq****: This tool helps us to remove unwanted characters like “\*” in start and end of the sequence.*

**

**

**

*8) The python program to find average stacking energy of given sequence is:*

*seq=input("Enter your sequence:")*

*stack\_energy=0*

*for i in range(len(seq)-1):*

*if(seq[i]=='A'):*

*if(seq[i+1]=='A'):*

*stack\_energy=stack\_energy + (-4)*

*if(seq[i+1]=='T'):*

*stack\_energy=stack\_energy + (-7)*

*if(seq[i+1]=='C'):*

*stack\_energy=stack\_energy + (-5)*

*if(seq[i+1]=='G'):*

*stack\_energy=stack\_energy + (-11)*

*if(seq[i]=='T'):*

*if(seq[i+1]=='A'):*

*stack\_energy=stack\_energy + (-7)*

*if(seq[i+1]=='T'):*

*stack\_energy=stack\_energy + (-2)*

*if(seq[i+1]=='C'):*

*stack\_energy=stack\_energy + (-3)*

*if(seq[i+1]=='G'):*

*stack\_energy=stack\_energy + (-4)*

*if(seq[i]=='C'):*

*if(seq[i+1]=='A'):*

*stack\_energy=stack\_energy + (-9)*

*if(seq[i+1]=='T'):*

*stack\_energy=stack\_energy + (-5)*

*if(seq[i+1]=='C'):*

*stack\_energy=stack\_energy + (-6)*

*if(seq[i+1]=='G'):*

*stack\_energy=stack\_energy + (-7)*

*if(seq[i]=='G'):*

*if(seq[i+1]=='A'):*

*stack\_energy=stack\_energy + (-9)*

*if(seq[i+1]=='T'):*

*stack\_energy=stack\_energy + (-6)*

*if(seq[i+1]=='C'):*

*stack\_energy=stack\_energy + (-4)*

*if(seq[i+1]=='G'):*

*stack\_energy=stack\_energy + (11)*

*avg\_stack\_energy=stack\_energy/(len(seq)-1)*

*print("The average stacking energy of given sequence is:",avg\_stack\_energy)*

*When we provide the input sequence given in the question 2, the output average stacking energy we get is:*

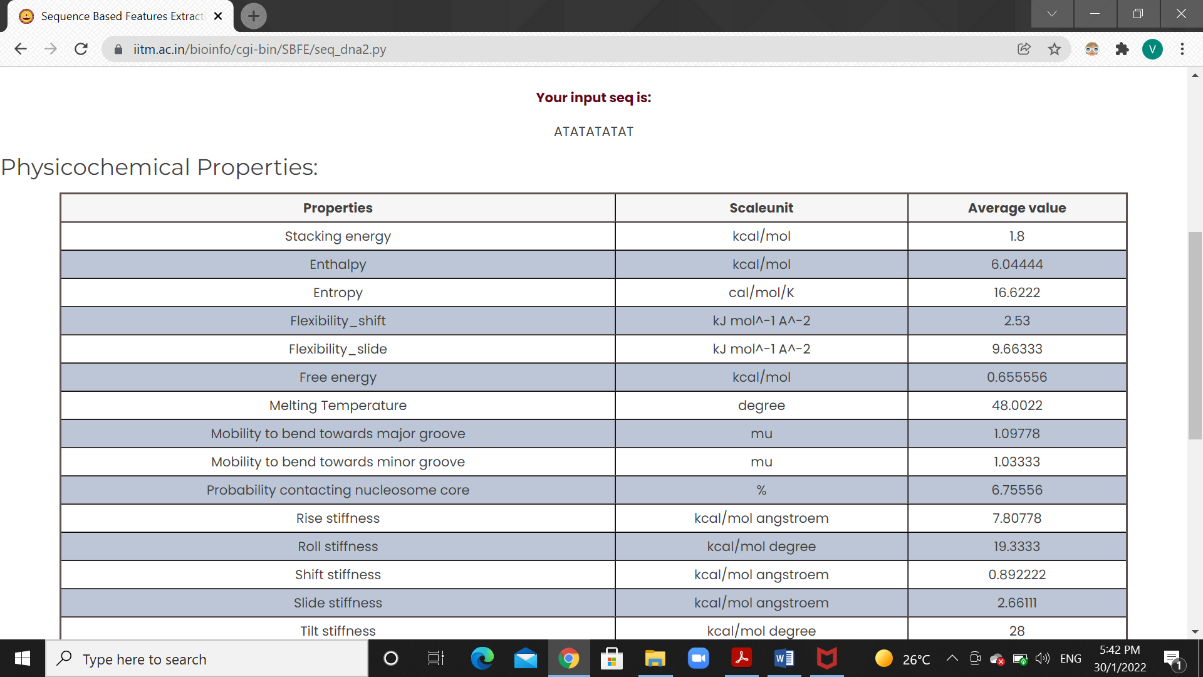
*-5.804347826086956*

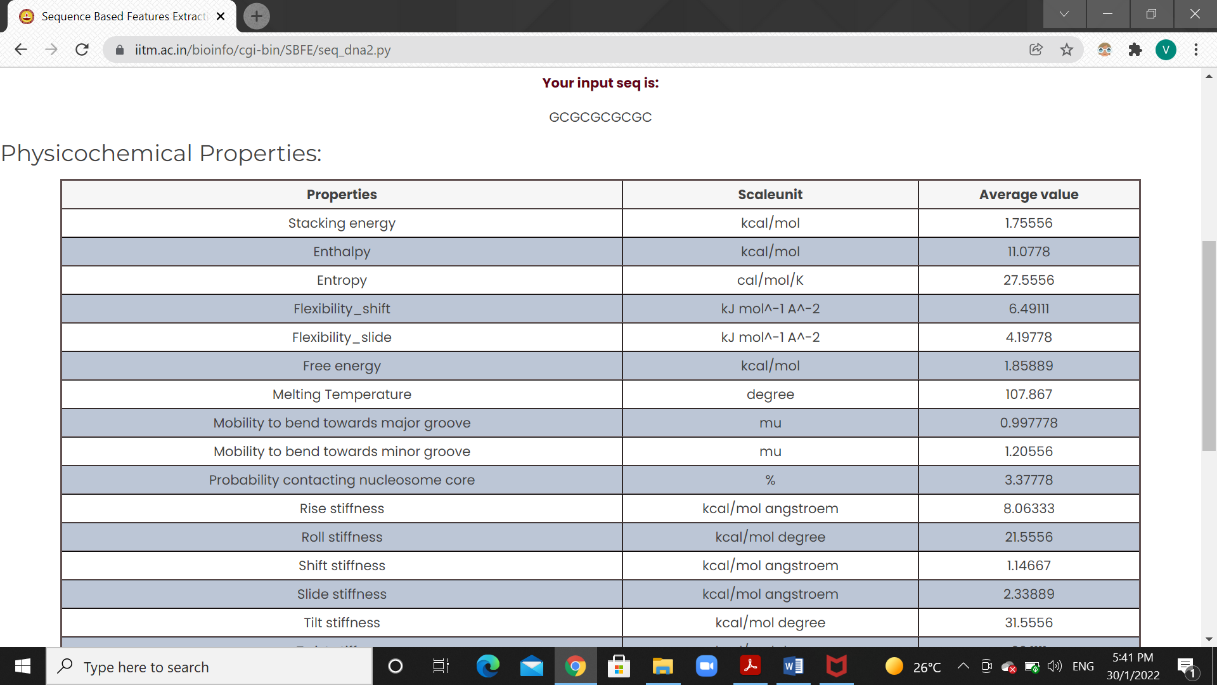
*9) Using the seq2feature app, I calculated the melting point for given sequences:*

*(i) ATATATATAT: 48.0022 degrees*

*(ii) GCGCGCGC: 107.867 degrees*

*As we know that, in a DNA double strand G and C forms three hydrogen bonds between them, whereas, A and T forms only two hydrogen bonds. So, since G and C sequence has many bonds, their melting temperature is higher than A and T sequence.*

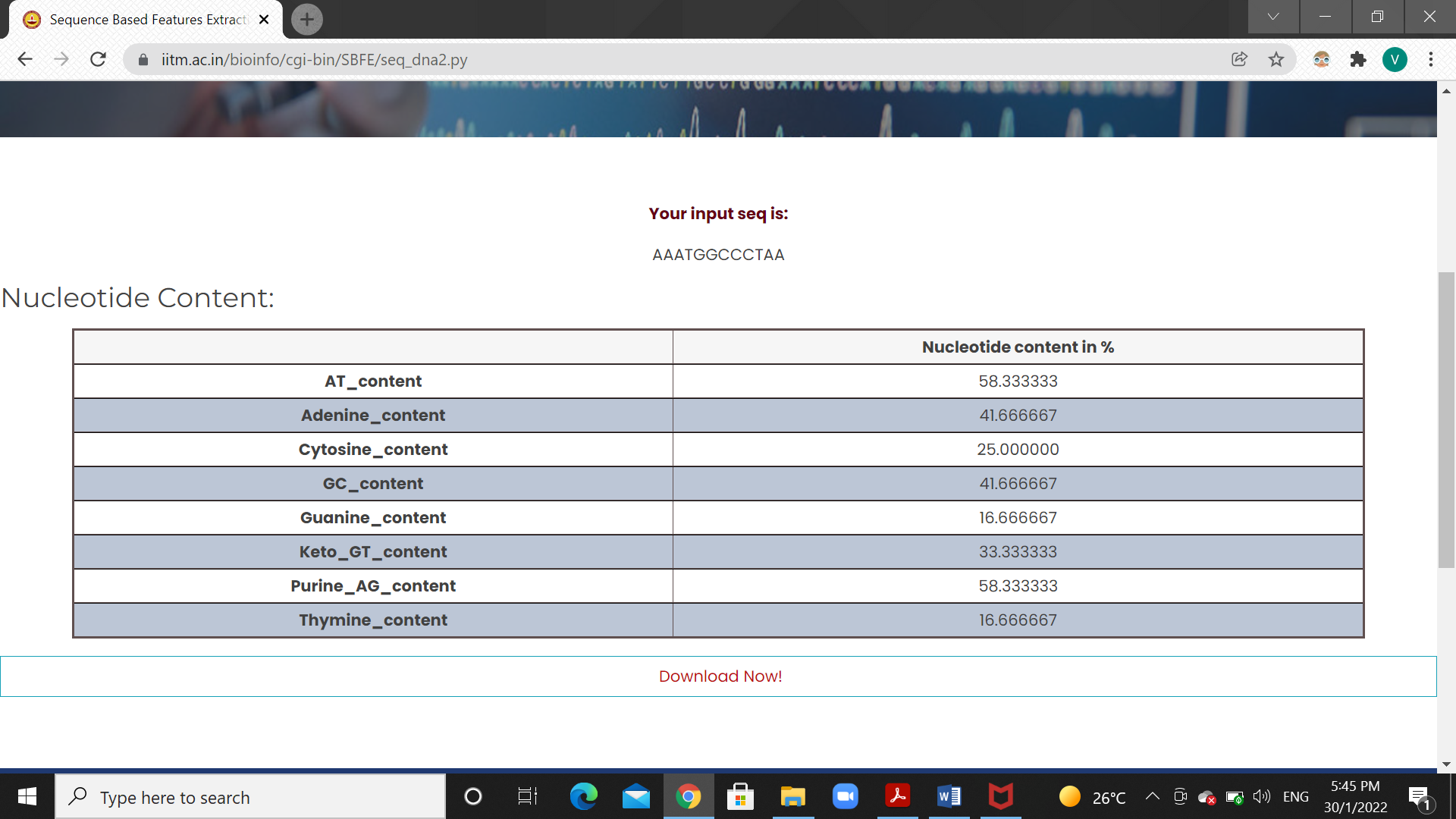
**

**

*10) Using seq2feature tool I found the AT and GC content in the given sequence:*

*AAATGGCCCTAA:*

* *AT content – 58.333333*
* *GC content – 41.666667*

**