BT 3040: BIOINFORMATICS

Assignment 9

Atharva Mandar Phatak | BE21B009
Department of Biotechnology

Indian Institute of Technology
Madras

Q1) 1. Identify the pair of sequences which are close to each other using Hamming and Euclidean distance methods.

(i)AMENLNMDLLYMAAAVMMGLAAIGAAIGIGILGGKFLEGAARQPDLIP LLRTQFFIVMGLVDAIPMIAVG

LGLYVMFAVA

(ii)AADVSAAVGATGQSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAG YTYWEGGDEGAGKHSLSFAP

VFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRIGAGLKFAN GQSVGVRAIHYSNAGLKQPN

DGIESYSLFYKIPI

(iii)MALLPAAPGAPARATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYA WTGLLTASKPSLHHATATPEY

LAALKQKSRHAA

```
#Q1 | Atharva Mandar Phatak | BE21B009 | Assignment 9 | BT3040 |

def calculate_aa_percentage(sequence):
    # Define the 20 standard amino acids
    amino_acids = "ACDEFGHIKLMNPQRSTVWY"

    # Initialize a dictionary to store the percentage occurrence of each amino acid
    aa_percentage = {aa: 0 for aa in amino_acids}

    # Count occurrences of each amino acid in the sequence
    total_aa_count = 0
    for aa in sequence:
        if aa in amino_acids:
            aa_percentage[aa] += 1
            total_aa_count += 1

    # Calculate percentage occurrence of each amino acid
    if total_aa_count > 0:
        for aa in aa_percentage:
            aa_percentage[aa] = (aa_percentage[aa] / total_aa_count) * 100

    # Sort the dictionary by keys
    sorted_aa_percentage = {k: v for k, v in sorted(aa_percentage.items())}

    return sorted_aa_percentage

def hamming(dict1, dict2):
    # Initialize the sum of differences
    sum_of_differences = 0
```

```
    # Iterate through each key in the dictionaries
    for key in dict1.keys():
        # Calculate the difference between the values of corresponding keys
        difference = dict1[key] - dict2[key]

        # Take the modulus of the difference
        difference_mod = abs(difference)  # Assuming you meant mod 100

        # Add the modulus to the sum
        sum_of_differences += difference_mod

    return sum_of_differences



def euclidean(dict1, dict2):
    # Initialize the sum of squared differences
    sum_of_squared_differences = 0

    # Iterate through each key in the dictionaries
    for key in dict1.keys():
        # Calculate the difference between the values of corresponding keys
        difference = dict1[key] - dict2[key]

        # Square the difference
        squared_difference = difference ** 2

        # Add the squared difference to the sum
        sum_of_squared_differences += squared_difference

    # Take the square root of the sum of squared differences
    sqrt_sum = (sum_of_squared_differences)**(0.5)
    return sqrt_sum



dict1 = calculate_aa_percentage(seq1)
dict2 = calculate_aa_percentage(seq2)
dict3 = calculate_aa_percentage(seq3)

hammering_1_2 = hamming(dict1, dict2)
hammering_1_3 = hamming(dict1, dict3)
hammering_2_3 = hamming(dict2, dict3)

eucledian_1_2= euclidean(dict1, dict2)
eucledian_1_3= euclidean(dict1, dict3)
eucledian_2_3= euclidean(dict2, dict3)

print(f"For pair 1 and 2, Hamming Distance is {hammering_1_2}")
```

```
print(f"For pair 1 and 3, Hamming Distance is {hammering_1_3}")
print(f"For pair 2 and 3, Hamming Distance is {hammering_2_3}")
print("")
print(f"For pair 1 and 2, Euclidean Distance is {eucledian_1_2}")
print(f"For pair 1 and 3, Euclidean Distance is {eucledian_1_3}")
print(f"For pair 2 and 3, Euclidean Distance is {eucledian_2_3}")
```

Output:

```
For pair 1 and 2, Hamming Distance is 66.5728476821192
For pair 1 and 3, Hamming Distance is 84.33544303797467
For pair 2 and 3, Hamming Distance is 72.66325760751111

For pair 1 and 2, Euclidean Distance is 20.1062168421535
For pair 1 and 3, Euclidean Distance is 22.086816691389576
For pair 2 and 3, Euclidean Distance is 20.112952107271116
```

Pair	Hamming Distance	Euclidean Distance
Seq 1 and Seq 2	66.57	20.11
Seq 1 and Seq 3	84.34	22.09
Seq 2 and Seq 3	72.66	20.11
Shortest	**66.57**	**20.11**

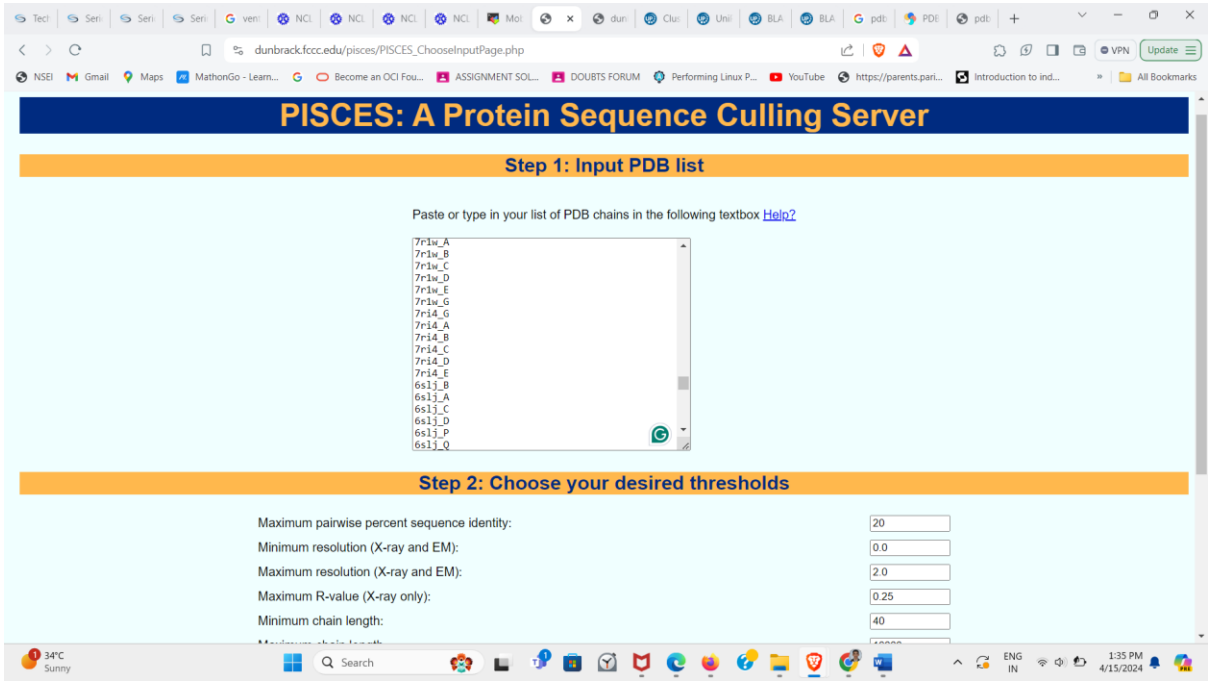
Q2) Get the non-redundant sequences of beta barrel membrane proteins with sequence identities of less than 40%, 50%, 75% and 90% using CD-HIT

CD-HIT	
Percentage Identity	**No. of Clusters**
40%	240
50%	265
75%	330
90%	370

The downloaded files are in the folder

Q3) Get the non-redundant sequences of the same type of proteins with sequence identities of less than 20%, 30%, 40% and 50% using PISCES (https://dunbrack.fccc.edu/pisces/

a. 20% (similarly for others, screenshot attaching only for one)



PISCES	
Percentage Identity	**No. of Clusters**
20%	31
30%	39
40%	42
50%	47

The downloaded files are in the folder

Q4) Compare the results obtained with the cut-offs 40% and 50%.

Comparision				
Resource	**Percentage Identity**	**Mean Length**	**Min Length**	**Max Length**
CD-HIT	40	341.6	11	2124
	50	341.6	11	2124
PISCES	40	390.8	99	77
	50	393.5	99	774

The number of chains for **40%** and **50%** are **42** and **47** respectively

Analysis:

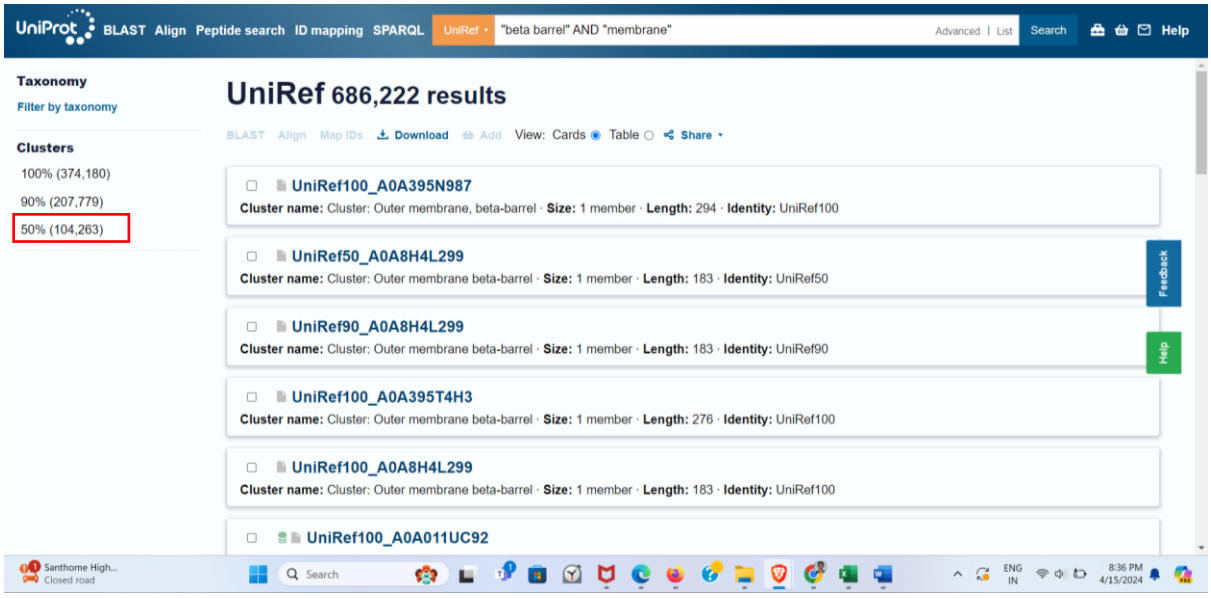
Comparing the length of sequences between PISCES and CD-HIT datasets reveals that, on average, PISCES tends to include longer sequences, possibly favouring complete protein structures, while CD-HIT encompasses a broader range of sequence lengths, from short fragments to longer ones.

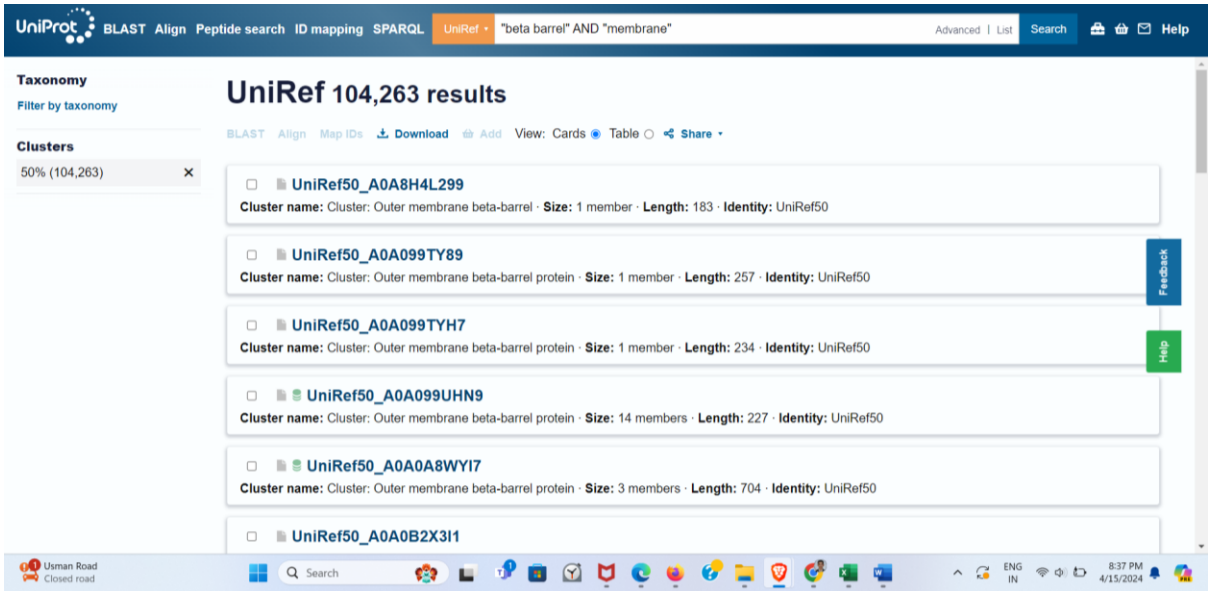
CD-HIT clusters formed with default cutoff values of 40% and 50% for redundant beta barrel membrane proteins resulted in smaller clusters with higher sequence similarity, suggesting tight groupings of very similar sequences.

PISCES selects sequences above a higher cutoff based on structural quality and non-redundancy, indicating a focus on structural integrity. The resolutions of sequences in PISCES datasets show minimal variation between the 40% and 50% cutoffs, indicating consistently high-quality structures.

In terms of unique contributions, CD-HIT is well-suited for studies requiring high sequence similarity, such as sequence homology or evolutionary analysis, while PISCES is more beneficial for structural biology, prioritizing high-quality and unique protein structures for applications like modelling and drug design.

Q5) Extract the data with the cut-off of 50% from UniProt and compare with CD-HIT and PISCES





Comparison:

Clusters with 50% cutoff	
UniProt	104263
CD-HIT	47
PISCES	265