Practical 9

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

Q1 set of sequences

i AMENLNMDLLYMAAAVMMGLAAIGAAIGIGILGGKFLEGAARQPDL IPLLRTQFFIVMGLVDAIPMIAVGLGLYVMFAVA

ii ADVSAAVGATGQSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGY TYWEGGDEGAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRV GDQNLGSSLNFEDRIGAGLKFANGQSVGVRAIHYSNAGLKQPNDGIE SYSLFYKIPI

iii MALLPAAPGAPARATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYA WTGLLTASKPSLHHATATPEYLAALKQKSRHAA

Solution. In order to achieve the above, we first write the codes for obtaining composition, calculating hamming distance and calculating euclidean distance between two sequences.
Now, we have three sequences, hence three pairs of sequences to compare closeness. (i) and (ii), (ii) and (iii), (iii) and (i). After comparing them we can easily predict which pair is closest to each other based on the given two metrics.

```python
amino_acid_list =
{"A": 0,"R": 1,"N": 2,"D": 3,"C": 4,
 "Q": 5,"E": 6,"G": 7,"H": 8,"I": 9,
 "L":10,"K":11,"M":12,"F":13,"P":14,
 "S":15,"T":16,"W":17,"Y":18,"V":19}

# Obtain the composition of amino acids (not in percentage)
def composition(seq):
    amino_acid_count = [0 for i in range(20)]
    for i in range(len(seq)):
        amino_acid_count[amino_acid_list[seq[i]]] += 1

    len_seq = len(seq)
    for i in range(len(amino_acid_count)):
        amino_acid_count[i] /= len_seq

    return amino_acid_count

# Function to compute hamming distance between any two sequences
def hamming_distance(c1, c2):
    dist = 0
    for i in range(len(c1)):
        dist += abs(c1[i] - c2[i])
    return dist

# Function to compute euclidean distance between any two sequences
def euclidean_distance(c1, c2):
    dist = 0
    for i in range(len(c1)):
        dist += (c1[i] - c2[i]) ** 2
    dist = dist ** 0.5
    return dist
```

```

seq1 =
"AMENLNMDLLYMAAAVMMGLAAIGAAIGIGILGGKFLEGAARQPDLIPLLRTQF
FIVMGLVDAIPMIAVGLGLYVMFAVA"
seq2 =
"AADVSAAVGATGQSGMTYRLGLSWDWDKSWWQTSTGRLTGYWDAGYTYWEGGDE
GAGKHSLSFAPVFVYEFAGDSIKPFIEAGIGVAAFSGTRVGDQNLGSSLNFEDRI
GAGLKFANGQSVGVRAIHYSNAGLKQPNDGIESYSLFYKIPI"
seq3 =
"MALLPAAPGAPARATPTRWPVGCFNRPWTKWSYDEALDGIKAAGYAWTGLLTAS
KPSLHHATATPEYLAALKQKSRHAA"

comp1 = composition(seq1)
comp2 = composition(seq2)
comp3 = composition(seq3)

# Finding best hamming distance
ham1 = hamming_distance(comp1, comp2)
ham2 = hamming_distance(comp2, comp3)
ham3 = hamming_distance(comp3, comp1)
print(ham1, ham2, ham3)

if ham1 < ham2 and ham1 < ham3:
    print("Sequence 1 and 2 are closest to each other via hamming distance")

# Finding best euclidean distance
euc1 = euclidean_distance(comp1, comp2)
euc2 = euclidean_distance(comp2, comp3)
euc3 = euclidean_distance(comp3, comp1)
print(euc1, euc2, euc3)

if euc1 < euc2 and euc1 < euc3:
    print("Sequence 1 and 2 are closest to each other via hamming distance")


# Output
0.6657284768211922 0.726632576075111 0.8433544303797469
Sequence 1 and 2 are closest to each other via hamming distance
0.20106216842153507 0.20112952107271118 0.2208681669138957
Sequence 1 and 2 are closest to each other via hamming distance
```

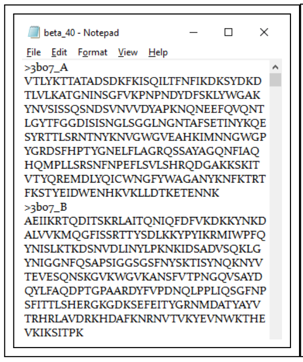
Listing 1. Functions to compute α helix and β strand ranges

Pair of sequences	**Hamming distance**	**Euclidean distance**
(i) and (ii)	0.6657284768211922	0.20106216842153507
(ii) and (iii)	0.726632576075111	0.20112952107271118
(iii) and (i)	0.8433544303797469	0.2208681669138957

- **Closest pair based on hamming distance**: In case of the comparison of pair of sequences based on **hamming distance**, we can see that the **sequences 1 and 2 are closest**.
- **Closest pair based on euclidean distance**: In case of the comparison of pair of sequences based on **euclidean distance**, we can see that the **sequences 1 and 2 are closest**.

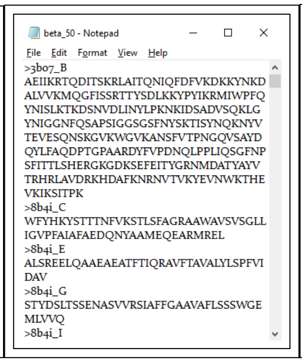
Question 2. Get the non-redundant sequences of beta barrel membrane proteins with sequence identities of less than 40%, 50%, 75% and 90% using CD-HIT.

Solution. Firstly, go to the website of **PDBTM** database (http://pdbtm.enzim.hu/). It is a comprehensive transmembrane protein selction of the Protein Data Bank (PDB). Now, download the redundant beta barrel protein sequences (TMB) in FASTA format. Now, the task is to obtain the non-redundant sequences. For this, we use CD-HIT software for different cut-offs mentioned in the question.
Install the MobaXterm (https://mobaxterm.mobatek.net/download-home-edition.html), and access the lab linux system through the SSH command given. The input file (**beta.fasta**) is already present, along with the CD Hits for different cutoffs. Download the text files for cutoffs of 40%, 50%, 75%, and 90%.



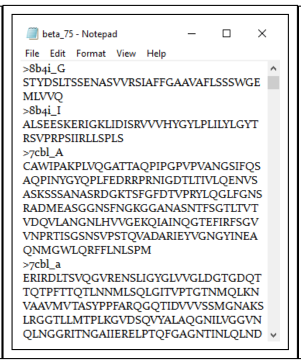
```
>3b07_A
VTLYKTTATADSDKFKISQILTFNFIKDKSYDKD
TLVLKATGNINSGFVKPNPNDYDFSKLYWGAK
YNVSISSQSNDSVNVVDYAPKNQNEEFQVQNT
LGYTFGGDISISNGLSGGLNGNTAFSETINYKQE
SYRTTLSRNTNYKNVGWGVEAHKIMNNGWGP
YGRDSFHPTYGNELFLAGRQSSAYAGQNFIAQ
HQMPLLSRSNFNPEFLSVLSHRQDGAKKSKIT
VTYQREMDLYQICWNGFYWAGANYKNFKTRT
FKSTYEIDWENHKVKLLDTKETENNK
>3b07_B
AEIIKRTQDITSKRLAITQNIQFDFVKDKKYNKD
ALVVKMQGFISSRTTYSDLKKYPYIKRMIWPFQ
YNISLKTKDSNVDLINYLPKNKIDSADVSQKLG
YNIGGNFQSAPSIGGSGSFNYSKTISYNQKNYV
TEVESQNSKGVKWGVKANSFVTPNGQVSAYD
QYLFAQDPTGPAARDYFVPDNQLPPLIQSGFNP
SFITTLSHERGKGDKSEFEITYGRNMDATYAYV
TRHRLAVDRKHDAFKNRNVTVKYEVNWKTHE
VKIKSITPK
```

(A) beta_40



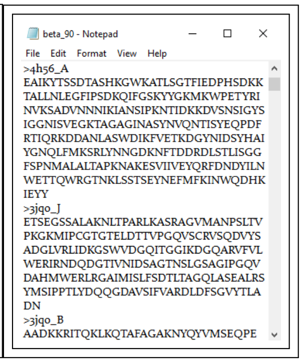
```
>3b07_B
AEIIKRTQDITSKRLAITQNIQFDFVKDKKYNKD
ALVVKMQGFISSRTTYSDLKKYPYIKRMIWPFQ
YNISLKTKDSNVDLINYLPKNKIDSADVSQKLG
YNIGGNFQSAPSIGGSGSFNYSKTISYNQKNYV
TEVESQNSKGVKWGVKANSFVTPNGQVSAYD
QYLFAQDPTGPAARDYFVPDNQLPPLIQSGFNP
SFITTLSHERGKGDKSEFEITYGRNMDATYAYV
TRHRLAVDRKHDAFKNRNVTVKYEVNWKTHE
VKIKSITPK
>8b4i_C
WFYHKYSTTTNFVKSTLSFAGRAAWAVSVSGLL
IGVPFAIAFAEDQNYAAMEQEARMREL
>8b4i_E
ALSREELQAAEAEATFTIQRAVFTAVALYLSPFVI
DAV
>8b4i_G
STYDSLTSSENASVVRSIAFFGAAVAFLSSSWGE
MLVVQ
>8b4i_I
```

(B) beta_50



```
>8b4i_G
STYDSLTSSENASVVRSIAFFGAAVAFLSSSWGE
MLVVQ
>8b4i_I
ALSEESKERIGKLIDISRVVVHYGYLPLILYLGYT
RSVPRPSIIRLLSPLS
>7cbl_A
CAWIPAKPLVQGATTAQPIPGPVPVANGSIFQS
AQPINYGYQPLFEDRRPRNIGDTLTIVLQENVS
ASKSSSANASRDGKTSFGFDTVPRYLQGLFGNS
RADMEASGGNSFNGKGGANASNTFSGTLTVT
VDQVLANGNLHVVGEKQIAINQGTEFIRFSGV
VNPRTISGSNSVPSTQVADARIEYVGNGYINEA
QNMGWLQRFFLNLSPM
>7cbl_a
ERIRDLTSVQGVRENSLIGYGLVVGLDGTGDQT
TQTPFTTQTLNNMLSQLGITVPTGTNMQLKN
VAAVMVTASYPPFARQGQTIDVVVSSMGNAKS
LRGGTLLMTPLKGVDSQVYALAQGNILVGGVN
QLNGGRITNGAIIERELPTQFGAGNTINLQLND
```

(C) beta_75



```
>4h56_A
EAIKYTSSDTASHKGWKATLSGTFIEDPHSDKK
TALLNLEGFIPSDKQIFGSKYYGKMKWPETYRI
NVKSADVNNNIKIANSIPKNTIDKKDVSNSIGYS
IGGNISVEGKTAGAGINASYNVQNTISYEQPDF
RTIQRKDDANLASWDIKFVETKDGYNIDSYHAI
YGNQLFMKSRLYNNGDKNFTDDRDLSTLISGG
FSPNMALALTAPKNAKESVIIVEYQRFDNDYILN
WETTQWRGTNKLSSTSEYNEFMFKINWQDHK
IEYY
>3jqo_J
ETSEGSSALAKNLTPARLKASRAGVMANPSLTV
PKGKMIPCGTGTELDTTVPGQVSCRVSQDVYS
ADGLVRLIDKGSWVDGQITGGIKDGQARVFVL
WERIRNDQDGTIVNIDSAGTNSLGSAGIPGQV
DAHMWERLRGAIMISLFSDTLTAGQLASEALRS
YMSIPPTLYDQQGDAVSIFVARDLDFSGVYTLA
DN
>3jqo_B
AADKKRITQKLKQTAFAGAKNYQYVMSEQPE
```

(D) beta_90

FIGURE 1. The FASTA files of cluster outputs from CD-HIT

To find the number of clusters, we just need to count the number of sequences in the above downloaded FASTA files. The code to count them is pretty simple as shown below:

```python
# All the 4 downloaded files are saved inside the folder named "Assignment9"
for i in os.listdir("Assignment9"):
    if "txt" in i:
        input_file = os.path.join("Assignment9", i)
        fasta_sequences = SeqIO.parse(open(input_file),'fasta')
        count = 0
        for fasta in fasta_sequences:
            count += 1
        print(i, count)
# Output:
beta_40.txt 240
beta_50.txt 265
beta_75.txt 330
beta_90.txt 370
```

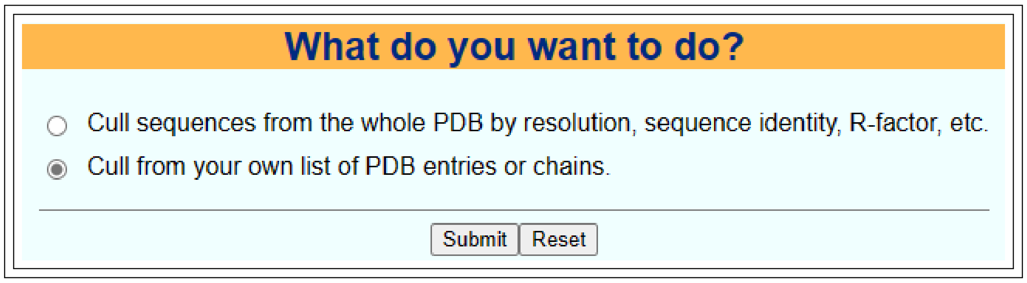
LISTING 2. Code to count the number of clusters

The results from **CD-HIT** are:

Identity (in %)	**Number of Clusters**
40	240
50	265
75	330
90	370

Question 3. 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/`).

Solution. Firstly, go to the website of **PISCES** (`https://dunbrack.fccc.edu/pisces/`). In this method of clustering, sequence identities for PDB sequences are determined by creating a hidden Markov model for every unique PDB sequence with the program HHblits and searching the resulting collection of HMMs with each individual HMM with the program HHsearch.
Now, we shall be using the same set of redundant beta barrel protein sequences in FASTA format as input. Since, we are giving a custom input, we navigate from the main page on PISCES via "*Access the server to create your own lists*". Now, we choose to cull from our own list of PDB entries or chains.



What do you want to do?

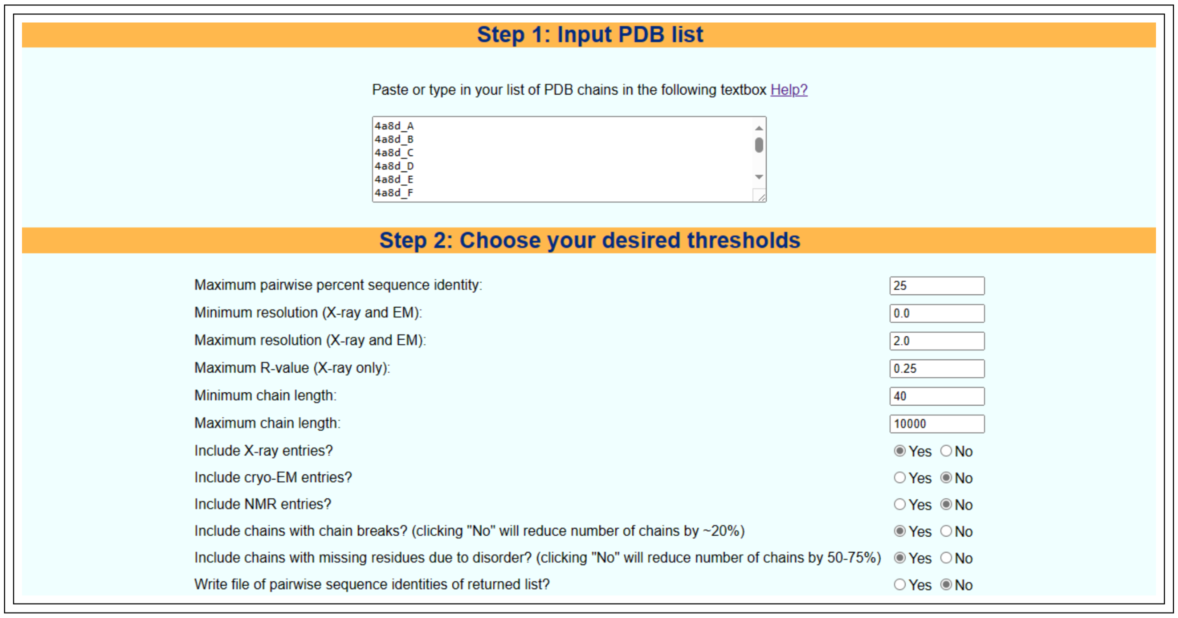
- ○ Cull sequences from the whole PDB by resolution, sequence identity, R-factor, etc.
- [✓] Cull from your own list of PDB entries or chains.

FIGURE 2. Cull from your own list of PDB entries

Now, we need to paste only the list of PDB chains from the input file, not the entire FASTA file itself. So, writing a code to extract that is very simple as shown below:

```python
from Bio import SeqIO
input_file = "beta.fasta"
fasta_sequences = SeqIO.parse(open(input_file),'fasta')
for fasta in fasta_sequences:
    name, sequence = fasta.id, str(fasta.seq)
    print(name)
```

LISTING 3. Obtaining PDB chains from FASTA file



Step 1: Input PDB list

Paste or type in your list of PDB chains in the following textbox Help?

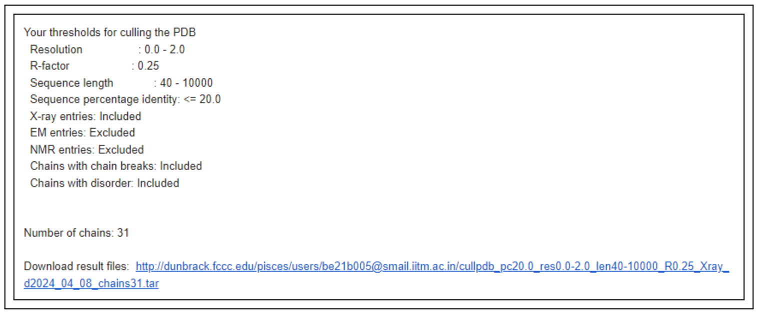
```
4a8d_A
4a8d_B
4a8d_C
4a8d_D
4a8d_E
4a8d_F
```

Step 2: Choose your desired thresholds

Maximum pairwise percent sequence identity:	25
Minimum resolution (X-ray and EM):	0.0
Maximum resolution (X-ray and EM):	2.0
Maximum R-value (X-ray only):	0.25
Minimum chain length:	40
Maximum chain length:	10000
Include X-ray entries?	[✓ Yes] No
Include cryo-EM entries?	Yes [✓ No]
Include NMR entries?	Yes [✓ No]
Include chains with chain breaks? (clicking "No" will reduce number of chains by ~20%)	[✓ Yes] No
Include chains with missing residues due to disorder? (clicking "No" will reduce number of chains by 50-75%)	[✓ Yes] No
Write file of pairwise sequence identities of returned list?	Yes [✓ No]

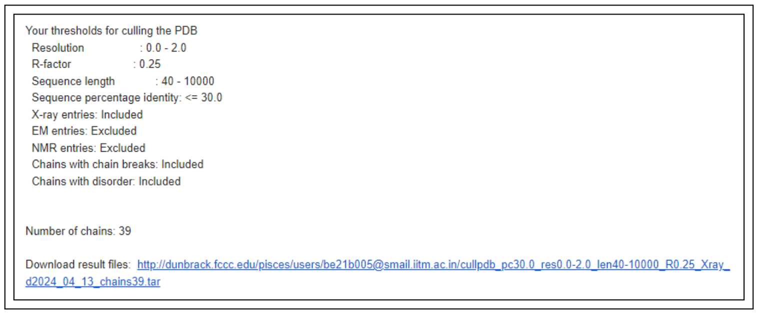
FIGURE 3. Input PDB chains and setting parameters

Now, we extract the PDB chain names, we copy the list and paste it in the "Input PDB list" section as shown in the image above. The second step is to choose the desired thresholds, which is essentially the cut-off percentages to be fixed. I need to iteratively change the Maximum pairwise percent sequence identity to 20%, 30%, 40%, and 50%. Their corresponding server output images are included below:



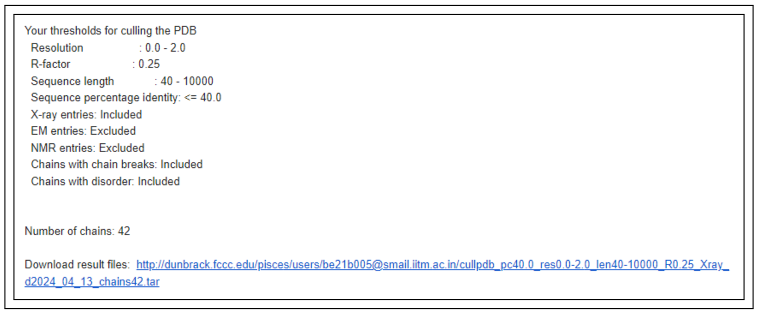
```
Your thresholds for culling the PDB
  Resolution               : 0.0 - 2.0
  R-factor               : 0.25
  Sequence length           : 40 - 10000
  Sequence percentage identity: <= 20.0
  X-ray entries: Included
  EM entries: Excluded
  NMR entries: Excluded
  Chains with chain breaks: Included
  Chains with disorder: Included


Number of chains: 31

Download result files:  http://dunbrack.fccc.edu/pisces/users/be21b005@smail.iitm.ac.in/cullpdb_pc20.0_res0.0-2.0_len40-10000_R0.25_Xray_
d2024_04_08_chains31.tar
```

FIGURE 4. beta_20 output generated by PISCES culling server



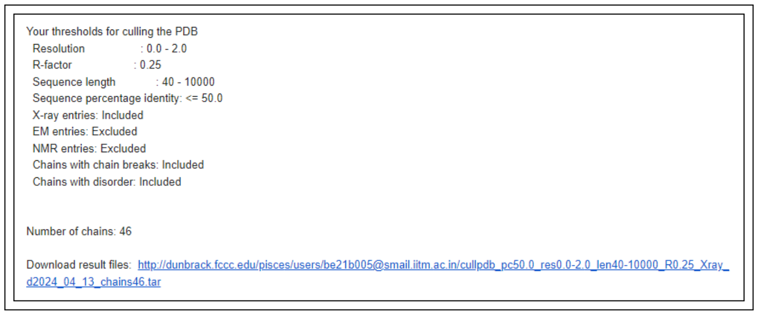
```
Your thresholds for culling the PDB
  Resolution               : 0.0 - 2.0
  R-factor               : 0.25
  Sequence length           : 40 - 10000
  Sequence percentage identity: <= 30.0
  X-ray entries: Included
  EM entries: Excluded
  NMR entries: Excluded
  Chains with chain breaks: Included
  Chains with disorder: Included


Number of chains: 39

Download result files:  http://dunbrack.fccc.edu/pisces/users/be21b005@smail.iitm.ac.in/cullpdb_pc30.0_res0.0-2.0_len40-10000_R0.25_Xray_
d2024_04_13_chains39.tar
```

FIGURE 5. beta_30 output generated by PISCES culling server



```
Your thresholds for culling the PDB
  Resolution               : 0.0 - 2.0
  R-factor               : 0.25
  Sequence length           : 40 - 10000
  Sequence percentage identity: <= 40.0
  X-ray entries: Included
  EM entries: Excluded
  NMR entries: Excluded
  Chains with chain breaks: Included
  Chains with disorder: Included


Number of chains: 42

Download result files:  http://dunbrack.fccc.edu/pisces/users/be21b005@smail.iitm.ac.in/cullpdb_pc40.0_res0.0-2.0_len40-10000_R0.25_Xray_
d2024_04_13_chains42.tar
```

FIGURE 6. beta_40 output generated by PISCES culling server



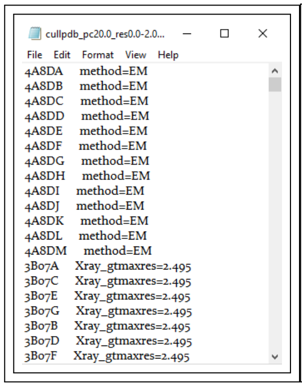
```
Your thresholds for culling the PDB
  Resolution               : 0.0 - 2.0
  R-factor               : 0.25
  Sequence length           : 40 - 10000
  Sequence percentage identity: <= 50.0
  X-ray entries: Included
  EM entries: Excluded
  NMR entries: Excluded
  Chains with chain breaks: Included
  Chains with disorder: Included


Number of chains: 46

Download result files:  http://dunbrack.fccc.edu/pisces/users/be21b005@smail.iitm.ac.in/cullpdb_pc50.0_res0.0-2.0_len40-10000_R0.25_Xray_
d2024_04_13_chains46.tar
```

FIGURE 7. beta_50 output generated by PISCES culling server

Now, the **tar** files or the result files can be downloaded to analyze the output sequences and understand them.
The results from **PISCES culling server** are:

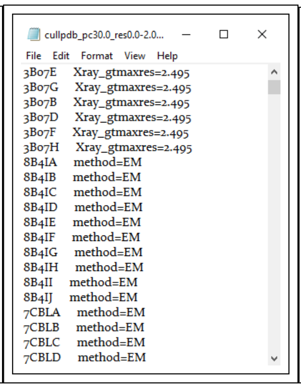
Identity (in %)	**Number of Clusters**
20	31
30	39
40	42
50	46

The above table gives the information of the number of clusters that have been obtained through PISCES (A Protein Sequence Culling Server). However, the PISCES also gives various additional information. It gives an output list of accession IDs with sequence length, structural determination method, resolution, and R-factor (if available) and a file of sequences in FASTA format.



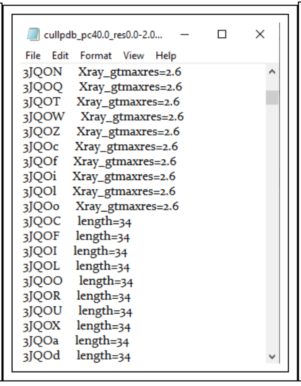
```
4A8DA    method=EM
4A8DB    method=EM
4A8DC    method=EM
4A8DD    method=EM
4A8DE    method=EM
4A8DF    method=EM
4A8DG    method=EM
4A8DH    method=EM
4A8DI    method=EM
4A8DJ    method=EM
4A8DK    method=EM
4A8DL    method=EM
4A8DM    method=EM
3B07A    Xray_gtmaxres=2.495
3B07C    Xray_gtmaxres=2.495
3B07E    Xray_gtmaxres=2.495
3B07G    Xray_gtmaxres=2.495
3B07B    Xray_gtmaxres=2.495
3B07D    Xray_gtmaxres=2.495
3B07F    Xray_gtmaxres=2.495
```

(A) cullpdb_pc20



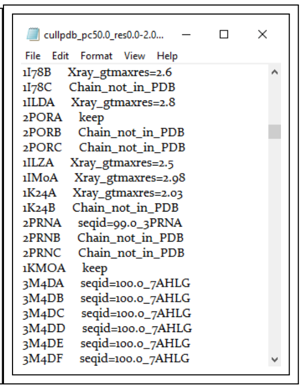
```
3B07E    Xray_gtmaxres=2.495
3B07G    Xray_gtmaxres=2.495
3B07B    Xray_gtmaxres=2.495
3B07D    Xray_gtmaxres=2.495
3B07F    Xray_gtmaxres=2.495
3B07H    Xray_gtmaxres=2.495
8B4IA    method=EM
8B4IB    method=EM
8B4IC    method=EM
8B4ID    method=EM
8B4IE    method=EM
8B4IF    method=EM
8B4IG    method=EM
8B4IH    method=EM
8B4II    method=EM
8B4IJ    method=EM
7CBLA    method=EM
7CBLB    method=EM
7CBLC    method=EM
7CBLD    method=EM
```

(B) cullpdb_pc30



```
3JQON    Xray_gtmaxres=2.6
3JQOQ    Xray_gtmaxres=2.6
3JQOT    Xray_gtmaxres=2.6
3JQOW    Xray_gtmaxres=2.6
3JQOZ    Xray_gtmaxres=2.6
3JQOc    Xray_gtmaxres=2.6
3JQOf    Xray_gtmaxres=2.6
3JQOi    Xray_gtmaxres=2.6
3JQOl    Xray_gtmaxres=2.6
3JQOo    Xray_gtmaxres=2.6
3JQOC    length=34
3JQOF    length=34
3JQOI    length=34
3JQOL    length=34
3JQOO    length=34
3JQOR    length=34
3JQOU    length=34
3JQOX    length=34
3JQOa    length=34
3JQOd    length=34
```

(C) cullpdb_pc40

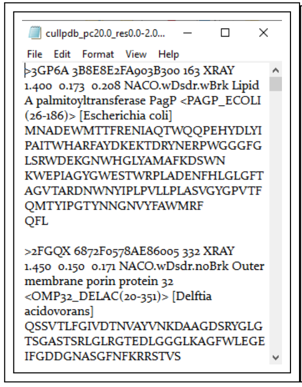


```
1I78B    Xray_gtmaxres=2.6
1I78C    Chain_not_in_PDB
1ILDA    Xray_gtmaxres=2.8
2PORA    keep
2PORB    Chain_not_in_PDB
2PORC    Chain_not_in_PDB
1ILZA    Xray_gtmaxres=2.5
1IM0A    Xray_gtmaxres=2.98
1K24A    Xray_gtmaxres=2.03
1K24B    Chain_not_in_PDB
2PRNA    seqid=99.0_3PRNA
2PRNB    Chain_not_in_PDB
2PRNC    Chain_not_in_PDB
1KMOA    keep
3M4DA    seqid=100.0_7AHLG
3M4DB    seqid=100.0_7AHLG
3M4DC    seqid=100.0_7AHLG
3M4DD    seqid=100.0_7AHLG
3M4DE    seqid=100.0_7AHLG
3M4DF    seqid=100.0_7AHLG
```

(D) cullpdb_pc50

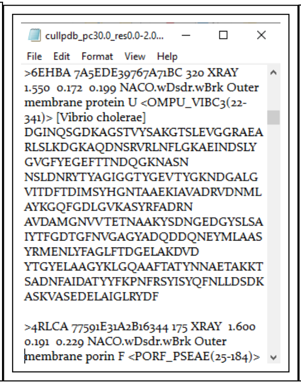
Figure 8. Structure determination in PISCES culling server

PISCES is a public server for culling sets of protein sequences from the Protein Data Bank (PDB) by sequence identity and structural quality criteria. PISCES can provide lists culled from the entire PDB or from lists of PDB entries or chains provided by the user. The sequence identities are obtained from PSI-BLAST alignments with position-specific substitution matrices derived from the non-redundant protein sequence database. PISCES can also cull non-PDB sequences provided by the user as a list of GenBank identifiers, a FASTA format file.



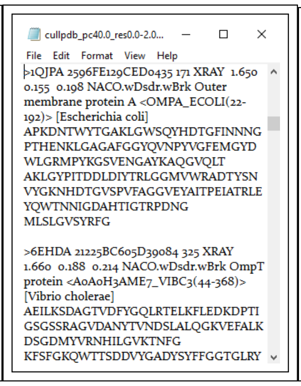
```
>3GP6A 3B8E8E2FA903B300 163 XRAY
1.400  0.173  0.208 NACO.wDsdr.wBrk Lipid
A palmitoyltransferase PagP <PAGP_ECOLI
(26-186)> [Escherichia coli]
MNADEWMTTFRENIAQTWQQPEHYDLYI
PAITWHARFAYDKEKTDRYNERPWGGGFG
LSRWDEKGNWHGLYAMAFKDSWN
KWEPIAGYGWESTWRPLADENFHLGLGFT
AGVTARDNWNYIPLPVLLPLASVGYGPVTF
QMTYIPGTYNNGNVYFAWMRF
QFL

>2FGQX 6872F0578AE86005 332 XRAY
1.450  0.150  0.171 NACO.wDsdr.noBrk Outer
membrane porin protein 32
<OMP32_DELAC(20-351)> [Delftia
acidovorans]
QSSVTLFGIVDTNVAYVNKDAAGDSRYGLG
TSGASTSRLGLRGTEDLGGGLKAGFWLEGE
IFGDDGNASGFNFKRRSTVS
```

(A) cullpdb_pc20



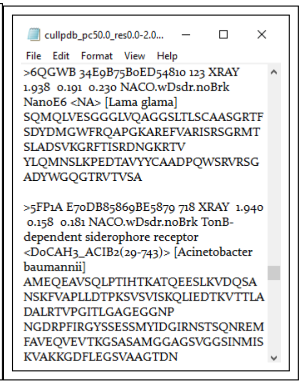
```
>6EHBA 7A5EDE39767A71BC 320 XRAY
1.550  0.172  0.199 NACO.wDsdr.wBrk Outer
membrane protein U <OMPU_VIBC3(22-
341)> [Vibrio cholerae]
DGINQSGDKAGSTVYSAKGTSLEVGGRAEA
RLSLKDGKAQDNSRVRLNFLGKAEINDSLY
GVGFYEGEFTTNDQGKNASN
NSLDNRYTYAGIGGTYGEVTYGKNDGALG
VITDFTDIMSYHGNTAAEKIAVADRVDNML
AYKGQFGDLGVKASYRFADRN
AVDAMGNVVTETNAAKYSDNGEDGYSLSA
IYTFGDTGFNVGAGYADQDDQNEYMLAAS
YRMENLYFAGLFTDGELAKDVD
YTGYELAAGYKLGQAAFTATYNNAETAKKT
SADNFAIDATYYFKPNFRSYISYQFNLLDSDK
ASKVASEDELAIGLRYDF

>4RLCA 77591E31A2B16344 175 XRAY  1.600
0.191  0.229 NACO.wDsdr.wBrk Outer
membrane porin F <PORF_PSEAE(25-184)>
```

(B) cullpdb_pc30



```
>1QJPA 2596FE129CED0435 171 XRAY  1.650
0.155  0.198 NACO.wDsdr.wBrk Outer
membrane protein A <OMPA_ECOLI(22-
192)> [Escherichia coli]
APKDNTWYTGAKLGWSQYHDTGFINNNG
PTHENKLGAGAFGGYQVNPYVGFEMGYD
WLGRMPYKGSVENGAYKAQGVQLT
AKLGYPITDDLDIYTRLGGMVWRADTYSN
VYGKNHDTGVSPVFAGGVEYAITPEIATRLE
YQWTNNIGDAHTIGTRPDNG
MLSLGVSYRFG

>6EHDA 21225BC605D39084 325 XRAY
1.660  0.188  0.214 NACO.wDsdr.wBrk OmpT
protein <A0A0H3AME7_VIBC3(44-368)>
[Vibrio cholerae]
AEILKSDAGTVDFYGQLRTELKFLEDKDPTI
GSGSSRAGVDANYTVNDSLALQGKVEFALK
DSGDMYVRNHILGVKTNFG
KFSFGKQWTTSDDVYGADYSYFFGGTGLRY
```

(C) cullpdb_pc40



```
>6QGWB 34E9B75B0ED54810 123 XRAY
1.938  0.191  0.230 NACO.wDsdr.noBrk
NanoE6 <NA> [Lama glama]
SQMQLVESGGGLVQAGGSLTLSCAASGRTF
SDYDMGWFRQAPGKAREFVARISRSGRMT
SLADSVKGRFTISRDNGKRTV
YLQMNSLKPEDTAVYYCAADPQWSRVRSG
ADYWGQGTRVTVSA

>5FP1A E70DB85869BE5879 718 XRAY  1.940
 0.158  0.181 NACO.wDsdr.noBrk TonB-
dependent siderophore receptor
<D0CAH3_ACIB2(29-743)> [Acinetobacter
baumannii]
AMEQEAVSQLPTIHTKATQEESLKVDQSA
NSKFVAPLLDTPKSVSVISKQLIEDTKVTTLA
DALRTVPGITLGAGEGGNP
NGDRPFIRGYSSESSMYIDGIRNSTSQNREM
FAVEQVEVTKGSASAMGGAGSVGGSINMIS
KVAKKGDFLEGSVAAGTDN
```

(D) cullpdb_pc50

Figure 9. FASTA files sequences output in PISCES culling server

Question 4. Compare the results obtained with the cut-offs 40% and 50%.

Solution. So, we need to compare the cut-offs 40% and 50%.

Comparison between cut-offs 40% and 50% results of PISCES

- Comparison has been done by exact matching of cluster representatives' sequences.
- Since less than 50% is inclusive of less than 40%, the 42 representatives in the latter are common in both the considered cases.
- Apart from the 42 common the former has 4 more cluster representatives.

Comparison between cut-offs 40% and 50% results of CD-HITS

- Comparison has been done by exact matching of cluster representatives' sequences.
- Since less than 50% is inclusive of less than 40%, the representatives in the latter are expected to be completely common in both the considered cases. However, only 235 representatives' sequences are common.
- Apart from the 235 common the 40% case has 5 more cluster representatives, while the 50% case has 30 more cluster representatives.

Comparison between cut-off 40% results of PISCES and CD-HITS

- Comparison has been done by exact matching of cluster representatives' sequences.
- Both methods of identifying representative sequences are quite different. Only 5 representatives' sequences are common.

Comparison between cut-off 50% results of PISCES and CD-HITs

- Comparison has been done by exact matching of cluster representatives' sequences.
- Both methods of identifying representative sequences are quite different. Only 5 representatives' sequences are common.

```python
input_file = "Assignment9\\pisces_40_fasta.fasta"
fasta_sequences = SeqIO.parse(open(input_file),'fasta')
name_40 = []
seq_40 = []

for fasta in fasta_sequences:
    name, sequence = fasta.id, str(fasta.seq)
    name_40.append(name)
    seq_40.append(sequence)

# Read FASTA files of pisces_50, cd-hit_40, cd-hit_50 in the same way.

a = list(seq_40.intersection(seq_50))
b = list(beta_seq_40.intersection(beta_seq_50))
c = list(beta_seq_40.intersection(seq_40))
d = list(beta_seq_50.intersection(seq_50))
e = list(beta_seq_40.intersection(seq_50))
f = list(beta_seq_50.intersection(seq_40))

print(len(a))

print(len(b))

print(len(c))

for i in range(len(c)):
    indices = list(beta_seq_40).index(c[i])
    indices2 = list(beta_seq_50).index(c[i])
    print(list(beta_name_40)[indices], list(beta_name_50)[indices2])

print(len(d))

for i in range(len(d)):
    indices = list(beta_seq_40).index(d[i])
    indices2 = list(beta_seq_50).index(d[i])
    print(list(beta_name_40)[indices], list(beta_name_50)[indices2])

print(len(e))

print(len(f))

# Output
42
235
5
7lhh_K 7rj5_E
6h04_C 4e1t_A
5wln_G 4gey_A
2o5p_A 5dl6_A
8bys_A 6cd2_A
5
7lhh_K 7rj5_E
6h04_C 4e1t_A
5wln_G 4gey_A
2o5p_A 5dl6_A
8bys_A 6cd2_A
5
5
```

Listing 4. Compare the 40% and 50% cutoff hits for both PISCES and CD-HITS

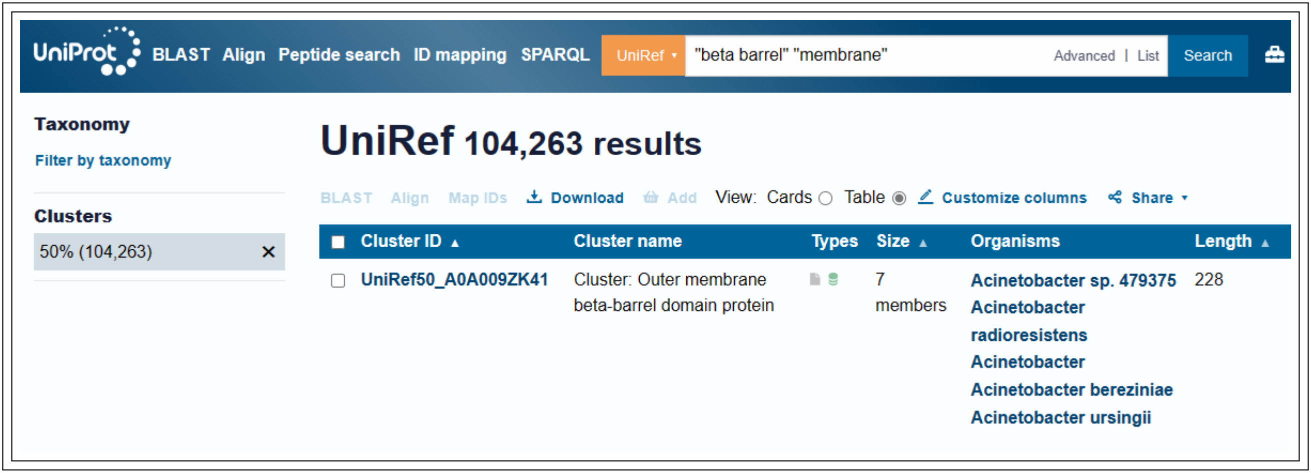
Identity (in %)	**Number of Clusters**
40 (PISCES)	42
50 (PISCES)	46
40 (CD-HITS)	240
50 (CD-HITS)	265

The above table is just a review of the previous two answers that are used to answer this question. The number of clusters are mentioned for the cases of PISCES (40% cutoff), PISCES (50% cutoff), CD-HITS (40% cutoff), and CD-HITS (50% cutoff). In the table below we have the number of common sequences between each pair of the above 4 groups, i.e., 10 pairs.

	40% (PISCES)	**50% (PISCES)**	**40% (CD-HITS)**	**50% (CD-HITS)**
40% (PISCES)	42	42	5	5
50% (PISCES)	42	46	5	5
40% (CD-HITS)	5	5	240	235
50% (CD-HITS)	5	5	235	265

Question 5. Extract the data with the cut-off of 50% from UniProt and compare with CD-HIT and PISCES.

Solution. To obtain the beta barrel membrane proteins' clusters from the UniProtKB, we go to **UniRef** and type in the search query "beta barrel" "membrane" to obtain the clusters.



UniRef 104,263 results

Cluster ID	Cluster name	Types	Size	Organisms	Length
UniRef50_A0A009ZK41	Cluster: Outer membrane beta-barrel domain protein		7 members	Acinetobacter sp. 479375 Acinetobacter radioresistens Acinetobacter Acinetobacter bereziniae Acinetobacter ursingii	228

FIGURE 10. UniRef beta barrel membrane proteins 50% clusters

Now, firstly the IDs present in UniRef are in a quite different format when compared to the IDs in PISCES and CD-HITS. Hence, comparing them is difficult. In the UniRef clusters, we have a reference cluster sequence along with the organism IDs of each cluster. However, in the case of CD-HITs and PISCES, we have the representative sequence for each cluster. Also, the ID naming for UniRef is quite different from the other two.
If we are to compare the reference cluster sequence of UniRef with the representative sequences of PISCES and CD-HITS, there are no common hits. Hence, comparison itself is unfeasible in this case. Hence, the below table has the number of clusters observed in the 50% cutoff of PISCES, CD-HITs, and UniRef.

Clustering method (for 50% identity)	**Number of Clusters obtained**
PISCES	46
CD-HITS	265
UniRef	104,263