

NAME: ASIF ERFAN KHAN

ROLL NUMBER: 546

COURSE: MSc CS

SUBJECT: BIOINFORMATICS

PRACTICAL: 1-10

INDEX				
NO	DATE	TITLE	PAGE NO	SIGN
1	08-08-22	Complementary DNA Sequence	2	
2	08-08-22	Identity of Two protein sequence	4	
3	26-08-22	Pairwise Sequence Alignment	6	
4	18-08-22	Similarity between two protein sequence	8	
5	21-08-22	Multiple Sequence Alignment	10	
6		Motif Finding	15	
7		Perform a BLAST search on any genes sequence and write a code to count the no of repetition of each nucleotide in the sequence	16	
8	29-09-22	Regular Expression	17	
9	22-09-22	Fingerprint	18	
10		Retrieving 3D structure from PDB		

PRACTICAL 1

Aim: Write a Python/Java code to perform pairwise alignment. Take 2 sequences from user and calculate the score.

```
se1=input("Enter the first sequence::")
se2=input("Enter the second sequence::")
seq1=list(se1)
seq2=list(se2)
score=[]
```

```
def Pairwise_alignment(a,b):
```

```
    gap(a,b)
    print(a)
    print(b)
    value=0
    length=len(a)
    for i in range(0,length):
        if(a[i]==b[i]):
            score.append('1')
            value=value+1
        else:
            score.append('0')
    print(score)
    print(value)
```

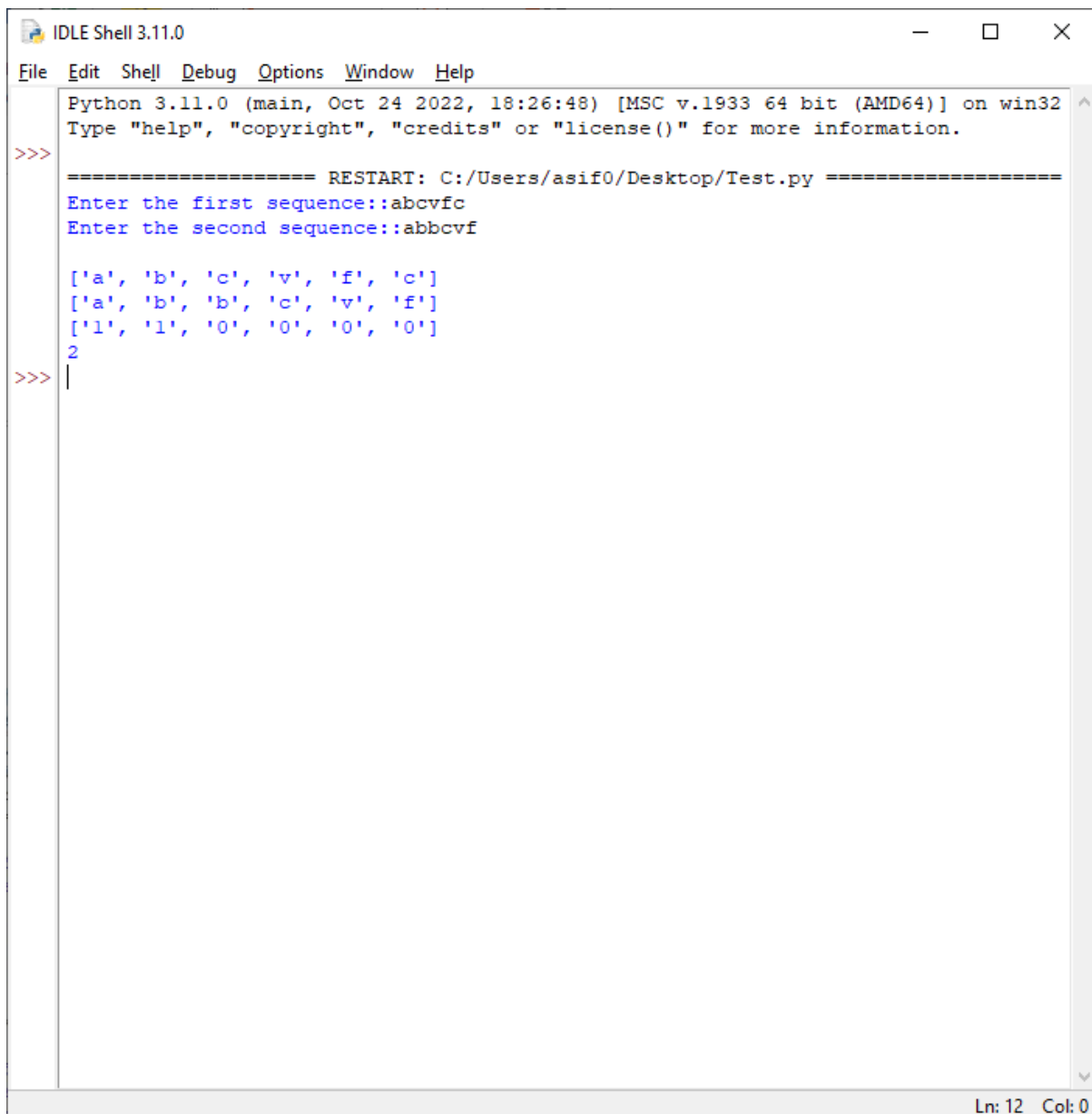
```
def gap(a,b):
```

```
    if(len(a)==len(b)):
        print()
    else:
        k=int(input("enter the position to insert::"))
        if (len(a)<len(b)):
```

```
    a.insert(k,'-')
else:
    b.insert(k,'-')
return(a,b)
```

Pairwise_alignment(seq1,seq2)

OUTPUT:



```
IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/asif0/Desktop/Test.py =====
Enter the first sequence::abcvfc
Enter the second sequence::abbcvf

['a', 'b', 'c', 'v', 'f', 'c']
['a', 'b', 'b', 'c', 'v', 'f']
['1', '1', '0', '0', '0', '0']
2
>>> |
```

Ln: 12 Col: 0

PRACTICAL 2

Aim: Write a Python/Java code to find the identity value of a given sequences. Take the sequence from user.

```
se1=input("Enter the first sequence::")
```

```
se2=input("Enter the second sequence::")
```

```
seq1=list(se1)
```

```
seq2=list(se2)
```

```
def find_identity(a,b):
```

```
    gap(a,b)
```

```
    print(a)
```

```
    print(b)
```

```
    score=0
```

```
    length=len(a)
```

```
    total_elements=len(a)*len(b)
```

```
    for i in range(0,length):
```

```
        for j in range(0,length):
```

```
            if(a[i]==b[j]):
```

```
                score=score+1
```

```
    identity=(score/total_elements)*100
```

```
    print("Matching Score::",score)
```

```
    print("Identity of the sequences::",identity)
```

```
def gap(a,b):
```

```
    if(len(a)==len(b)):
```

```
        print()
```

```
    else:
```

```
        k=int(input("enter the position to insert gap ::"))
```

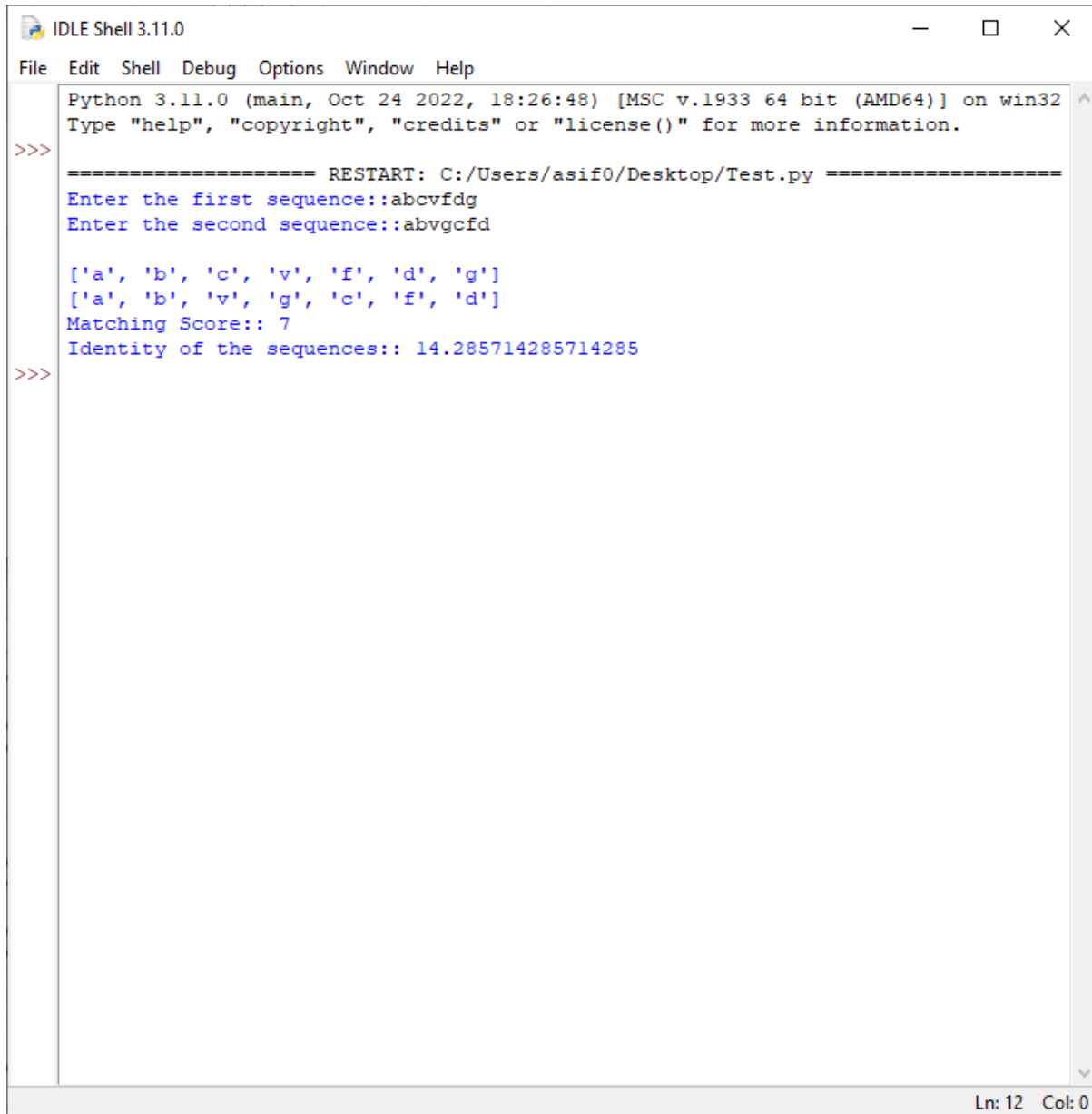
```
        if (len(a)<len(b)):
```

```
            a.insert(k,'-')
```

```
        else:
```

```
b.insert(k, '-')  
  
return(a,b)  
  
find_identity(seq1,seq2)
```

OUTPUT:



```
IDLE Shell 3.11.0  
File Edit Shell Debug Options Window Help  
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>> ===== RESTART: C:/Users/asif0/Desktop/Test.py =====  
Enter the first sequence::abcvfdg  
Enter the second sequence::abvgcfd  
  
['a', 'b', 'c', 'v', 'f', 'd', 'g']  
['a', 'b', 'v', 'g', 'c', 'f', 'd']  
Matching Score:: 7  
Identity of the sequences:: 14.285714285714285  
>>>  
  
Ln: 12 Col: 0
```

PRACTICAL 3

Aim: Write a Python/Java code to find the Similarity value of a given sequences. Take the sequence from user.

```
sequence_one=input("Enter the first sequence: ")
sequence_two=input("Enter the second sequence: ")
how_many=int(input("How many elements for similarity condition?"))
similarities=[]

for i in range(0,how_many):
    a=input("Enter an element: ")
    c=int(input("How many elements is it similar to? "))
    similarities.append([])
    similarities[i].append(a)

    for j in range(0,c):
        b=input("What is it similar to? ")

        similarities[i].append(b)

def compare(o,t,s):
    print(o)
    print(t)
    print(s)
    #checking if similar
    score=0
    for i in range(len(o)):
        for j in range(len(s)):

            if o[i] in s[j] and t[i] in s[j] and o[i] != t[i]:

                score+=1
    #calculating similarity
```

```
similarity= (score*100)/len(o)
```

```
return similarity
```

```
print(compare(list(sequence_one),list(sequence_two),similarities,"%")
```

OUTPUT:



```
IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/asif0/Desktop/Test.py =====
Enter the first sequence: abcvdgfhijk
Enter the second sequence: abgcvfghi
How many elements for similarity condition? 2
Enter an element: a
How many elements is it similar to? 2
What is it similar to? j
What is it similar to? i
Enter an element: c
How many elements is it similar to? 3
What is it similar to? v
What is it similar to? f
What is it similar to? g
['a', 'b', 'c', 'v', 'd', 'g', 'f', 'h', 'i', 'j', 'k']
['a', 'b', 'g', 'c', 'v', 'f', 'g', 'h', 'i', 'j', 'i']
[['a', 'j', 'i'], ['c', 'v', 'f', 'g']]
36.363636363637 %
>>> |
```

Ln: 21 Col: 0

PRACTICAL 4

Aim: Enter genome of five different organism and write a python/java program to find consensus sequence using Multiple Sequence Alignment (MSA) technique.

```
import java.io.*;

import java.util.*;

public class Consensus
{
    public static void main(String str[]) throws IOException
    {
        int n, i,j,k,count;

        String seq[],cons[];

        ArrayList<Integer> a = new ArrayList<Integer>();

        ArrayList s = new ArrayList();

        BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

        System.out.println("Enter the no of Sequences");

        n=Integer.parseInt(br.readLine());

        seq=new String[n];

        System.out.println("Enter sequences");

        for(i=0;i<n;i++)

            seq[i]=br.readLine();

        cons=new String[seq[0].length()];

        for(j=0;j<seq[0].length();j++)

            cons[j]=" ";

        for(j=0;j<seq[0].length();j++)

        {

            a.clear();

            s.clear();

            for(i=0;i<n;i++)

            {

                count=1;
```

```

for(k=i+1;k<n;k++)
{

    if(seq[i].charAt(j)==seq[k].charAt(j))
        count++;

}

System.out.println("count="+count);
a.add(count);
s.add(seq[i].charAt(j));
}

/**Updated Snippet 1**/
Set<String> set = new HashSet<>(s);
ArrayList setlist = new ArrayList(set);
Collections.sort(setlist);
if (setlist.contains('-') && setlist.size()==2){
    cons[j]+="-"+setlist.get(1);
}
else if (setlist.size()==1){
    cons[j]+="-"+setlist.get(0);
}
else{
    int m = Collections.max(a);
    int index=a.indexOf(m);
    System.out.println("Max="+m);
    cons[j]+=s.get(index);
    System.out.println("index="+index);
    for(i=index+1;i<a.size();i++)
    {
        if(a.get(i)==m)
            cons[j]+="/" +s.get(i);
    }
}

```

```

    }
    }
}

System.out.println("Consensus=");
for(j=0;j<seq[0].length();j++){
    /**Updated Snippet 2**/
    if(cons[j].length()==2)
        System.out.print(cons[j].toLowerCase());
    else if(cons[j].length()==3)
        System.out.print(cons[j].replace("-", ""));
    else
        System.out.print(cons[j]);
    }
}
}

```

OUTPUT:

```

C:\Windows\system32\cmd.exe
Enter the no of Sequences
5
Enter sequences
ACTG
TCGA
TAATG
TCGA
TAA
count=1
count=3
count=2
count=1
count=1
Max=3
index=1
count=2
count=1
count=1
count=1
count=1
Max=2
index=0
count=3
count=1
count=2
count=1
count=1
Max=3
index=0
count=2
count=3
count=1
count=2
count=1
Max=3
index=1
Consensus=
t c t a
C:\Users\admin\Desktop>Pause
Press any key to continue . . .

```

PRACTICAL 5

Aim: Perform a BLAST search on a specific gene sequence of a specify organism.

Steps:

Go to the National Center for Biotechnology Information Site

<https://www.ncbi.nlm.nih.gov/>

Select Nucleotide from All Databases and find any organism in a search bar

National Library of Medicine
National Center for Biotechnology Information

An official website of the United States government
[Here's how you know](#)

Log in

Nucleotide
Search

NCBI Home
Resource List (A-Z)
All Resources
Chemicals & Bioassays
Data & Software
DNA & RNA
Domains & Structures
Genes & Expression
Genetics & Medicine
Genomes & Maps
Homology
Literature
Proteins
Sequence Analysis
Taxonomy
Training & Tutorials
Variation

Welcome to NCBI
The National Center for Biotechnology Information advances science and health by providing access to biomedical and genomic information.
[About the NCBI](#) | [Mission](#) | [Organization](#) | [NCBI News & Blog](#)

Submit
Deposit data or manuscripts into NCBI databases

Download
Transfer NCBI data to your computer

Learn
Find help documents, attend a class or watch a tutorial

Develop
Use NCBI APIs and code libraries to build applications

Analyze
Identify an NCBI tool for your data analysis task

Research
Explore NCBI research and collaborative projects

Popular Resources
PubMed
Bookshelf
PubMed Central
BLAST
Nucleotide
Genome
SNP
Gene
Protein
PubChem

NCBI News & Blog
NEW! Streamlining ClinVar Submission of Assertion Criteria
18 Nov 2022
ClinVar is a freely available submission-driven database for information about
Re-evaluating the BLAST Nucleotide Database (nt)
17 Nov 2022
The ongoing sequencing revolution has resulted in environmental growth of the
RefSeq Release 215
15 Nov 2022

☐ [Mosquito flavivirus NS5 gene for polyprotein .partial cds .strain .YDFV/Sept/2013](#)
4. 939 bp linear RNA
Accession: AB981187.1 GI: 824555718
[Protein](#) [PubMed](#) [Taxonomy](#)
[GenBank](#) [FASTA](#) [Graphics](#)

☐ [Mosquito flavivirus gene for polyprotein .complete cds .strain .YDFV/Oct/2013](#)
5. 10,863 bp linear RNA
Accession: AB981186.1 GI: 824555716
[Protein](#) [PubMed](#) [Taxonomy](#)
[GenBank](#) [FASTA](#) [Graphics](#)

☐ [Hangzhou flavivirus 3 isolate YW.FY92 .complete genome](#)
6. 9,631 bp linear RNA
Accession: M2209680.1 GI: 2168955026
[BioProject](#) [BioSample](#) [Protein](#) [Taxonomy](#)
[GenBank](#) [FASTA](#) [Graphics](#)

☐ [Tembusu virus flavivirus polyprotein \(flavivirus polyprotein gene\) .gene .complete cds](#)
7. 10,990 bp linear RNA
Accession: NC_015843.2 GI: 381333920
[Assembly](#) [BioProject](#) [Protein](#) [PubMed](#) [Taxonomy](#)
[GenBank](#) [FASTA](#) [Graphics](#)

☐ [Culex flavivirus gene for polyprotein .partial cds .clone .oeprc .flavi .ns5 .24](#)
8. 268 bp linear RNA
Accession: LC227582.1 GI: 1247173336
[Protein](#) [Taxonomy](#)
[GenBank](#) [FASTA](#) [Graphics](#)

☐ [Cacipacore virus flavivirus polyprotein \(flavivirus polyprotein gene\) and truncated polyprotein \(flavivirus polyprotein gene\) .genes .complete cds](#)
9. 10,284 bp linear RNA
Accession: NC_026623.1 GI: 765702599
[Assembly](#) [BioProject](#) [Protein](#) [PubMed](#) [Taxonomy](#)
[GenBank](#) [FASTA](#) [Graphics](#)

https://www.ncbi.nlm.nih.gov/nuccore/NC_015843.2

National Library of Medicine
National Center for Biotechnology Information

An official website of the United States government
[Here's how you know](#)

Log in

Nucleotide
Advanced
Search
Help

GenBank
Send to
Change region shown
Customize view

Tembusu virus flavivirus polyprotein (flavivirus polyprotein gene) gene, complete cds
NCBI Reference Sequence: NC_015843.2
[FASTA](#) [Graphics](#)
Go to:

LOCUS NC_015843 10990 bp ss-RNA linear VRL 13-AUG-2018
DEFINITION Tembusu virus flavivirus polyprotein (flavivirus polyprotein gene) gene, complete cds.
ACCESSION NC_015843 NC_016958 NC_018670
VERSION NC_015843.2
DBLINK BioProject: [PRJNA485481](#)
KEYWORDS RefSeq.
SOURCE Tembusu virus (THUV)
ORGANISM [Tembusu virus](#)
Viruses; Riboviria; Orthornavirae; Kitrinoviricota; Flaviviricetes; Amarillivirales; Flaviviridae; Flavivirus.

REFERENCE
AUTHORS Han,K., Huang,X., Li,Y., Zhao,D., Liu,Y., Zhou,X., You,Y. and Xie,X.
TITLE Complete genome sequence of goose tembusu virus, isolated from jiangnan white geese in jiangsu, china
JOURNAL Genome Announc 1 (2), E0623612 (2013)
PUBMED 23516233
REMARK Publication Status: Online-Only
REFERENCE 2 (bases 1 to 10990)
AUTHORS Huang,X., Han,K., Zhao,D., Liu,Y., Zhang,J., Niu,H., Zhang,K.,

Analyze this sequence
Run BLAST
Pick Primers
Highlight Sequence Features
Find in this Sequence

Articles about the flavivirus polyprotein gene gene
Substantial Attenuation of Virulence of Tembusu Virus Strain PS Is Determined by a [J Virol. 2021]
Identification of a Neutralizing Monoclonal Antibody That Recognizes a Uniq [Viruses. 2020]
A Single Mutation at Position 156 in the Envelope Protein of Tembusu Virus Is Respo [J Virol. 2018]
See all...

Reference sequence information
RefSeq protein product

Run BLAST option we have to select

Enter accession number(s), gi(s), or FASTA sequence(s) [Clear](#) Query subrange [?](#)

NC_015843.2 From To

Or, upload file [Choose file](#) No file chosen [?](#)

Job Title

☐ Align two or more sequences [?](#)

Choose Search Set

Database ☒ Standard databases (nr etc.) ☐ rRNA/ITS databases ☐ Genomic + transcript databases ☐ Betacoronavirus

Organism Nucleotide collection (nr/nt) [?](#)

☐ Enter organism name or id--completions will be suggested ☐ exclude [Add organism](#)

☐ Enter organism common name, binomial, or tax id. Only 20 top taxa will be shown [?](#)

Exclude ☐ Models (XM/XP) ☐ Uncultured/environmental sample sequences

Limit to ☐ Sequences from type material

Entrez Query [YouTube](#) [Create custom database](#)

Enter an Entrez query to limit search [?](#)

Program Selection

Optimize for ☒ Highly similar sequences (megablast) ☐ More dissimilar sequences (discontiguous megablast) ☐ Somewhat similar sequences (blastn)

Choose a BLAST algorithm [?](#)

BLAST Search database Nucleotide collection (nr/nt) using Megablast (Optimize for highly similar sequences)

☐ Show results in a new window

[+ Algorithm parameters](#)

BLAST

Descriptions	Graphic Summary	Alignments	Taxonomy					
Sequences producing significant alignments								
Download Select columns Show 100								
<input checked="" type="checkbox"/> select all 100 sequences selected								
GenBank Graphics Distance tree of results MSA Viewer								
Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
<input checked="" type="checkbox"/> Tembusu virus strain JS804 .complete genome	Tembusu virus	20295	20295	100%	0.0	100.00%	10990	JF895923.2
<input checked="" type="checkbox"/> Tembusu virus isolate SD2010 .complete genome	Tembusu virus	20098	20098	100%	0.0	99.67%	10990	MN649260.1
<input checked="" type="checkbox"/> Tembusu virus strain JS/2010 .complete genome	Tembusu virus	20064	20064	99%	0.0	99.64%	10990	JX273153.1
<input checked="" type="checkbox"/> Duck egg-drop syndrome virus strain byd1 .complete genome	Duck egg-drop syndrome virus	20048	20048	99%	0.0	99.61%	10990	JQ920420.1
<input checked="" type="checkbox"/> Tembusu virus isolate Tembusu virus strain .complete genome	Tembusu virus	20026	20026	99%	0.0	99.57%	10989	KF192951.1
<input checked="" type="checkbox"/> Duck Tembusu virus isolate df-2 .complete genome	Duck Tembusu virus	20020	20020	99%	0.0	99.56%	10990	KJ489355.1
<input checked="" type="checkbox"/> Duck egg-drop syndrome virus strain JXSP .complete genome	Duck egg-drop syndrome virus	20020	20020	99%	0.0	99.56%	10990	JQ920423.1
<input checked="" type="checkbox"/> Tembusu virus isolate HB2010 .complete genome	Tembusu virus	20018	20018	100%	0.0	99.55%	10990	MN649262.1
<input checked="" type="checkbox"/> Tembusu virus isolate YY5 .complete genome	Tembusu virus	20015	20015	99%	0.0	99.55%	10990	JF270480.1
<input checked="" type="checkbox"/> Tembusu virus isolate SDMS .complete genome	Tembusu virus	20009	20009	99%	0.0	99.54%	10990	KC333867.1
<input checked="" type="checkbox"/> Tembusu virus isolate ZJ-6 .complete genome	Tembusu virus	20009	20009	99%	0.0	99.54%	10990	JF459991.1
<input checked="" type="checkbox"/> Tembusu virus strain AH-F10 from China .complete genome	Tembusu virus	20004	20004	99%	0.0	99.54%	10990	KM102539.1
<input checked="" type="checkbox"/> Duck egg-drop syndrome virus strain pigeon .complete genome	Duck egg-drop syndrome virus	20004	20004	99%	0.0	99.54%	10990	JQ920425.1
<input checked="" type="checkbox"/> Tembusu virus genomic RNA .complete genome .strain: TMUV-YY10Du	Tembusu virus	19998	19998	99%	0.0	99.53%	10990	AB917088.1
<input checked="" type="checkbox"/> Duck Tembusu virus strain BZ_2010 .complete genome	Duck Tembusu virus	19998	19998	99%	0.0	99.53%	10990	KC990540.1
<input checked="" type="checkbox"/> Duck egg-drop syndrome virus strain duan .complete genome	Duck egg-drop syndrome virus	19998	19998	99%	0.0	99.53%	10990	JQ920421.1
<input checked="" type="checkbox"/> Duck Tembusu virus strain GDH01 .complete genome	Duck Tembusu virus	19989	19989	99%	0.0	99.51%	10990	KT824876.1
<input checked="" type="checkbox"/> Tembusu virus isolate pYY150902 polyprotein gene .complete cds	Tembusu virus	19981	19981	99%	0.0	99.50%	10990	MF522175.1

Here the result will be display

Tembusu virus isolate SD2010, complete genome
Sequence ID: [MN649260.1](#) Length: 10990 Number of Matches: 1

Range 1: 1 to 10990 [GenBank](#) [Graphics](#) [Next Match](#) [Previous Match](#)

Score	Expect	Identities	Gaps	Strand
20098 bits(10883)	0.0	10954/10990(99%)	0/10990(0%)	Plus/Plus
Query 1	AGAAGTTCGCCTGTGTGAAC	TATTCCAAACAGCTTTTGGAGTAGTGC	GTGTGAACGTAA	60
Sbjct 1	AGAAGTTCATCTGTGTGAAC	TATTCCAAACAGCTTTTGGAGTAGTGC	GTGTGAACGTAA	60
Query 61	ACACAGTTTGAACGTTTTTTTGGATAGAGACA	ACTATGCTAAACAAAAAC	CAGGAAGACC	120
Sbjct 61	ACACAGTTTGAACGTTTTTTTGGATAGAGACA	ACTATGCTAAACAAAAAC	CAGGAAGACC	120
Query 121	CGGCTCAGGCCGGGTTGTCAATATGCTAAAGC	GC	GGAACTCCGCTAGC	180
Sbjct 121	CGGCTCAGGCCGGGTTGTCAATATGCTAAAGC	GC	GGAACTCCGCTAGC	180
Query 181	GCGGATAAAGAGGACGATTGATGGGGTCTGAGAGG	AGCAGGACCCATAAGGTTTGTGCT		240
Sbjct 181	GCGGATAAAGAGGACGATTGATGGGGTCTGAGAGG	AGCAGGACCCATAAGGTTTGTGCT		240
Query 241	GGCTCTACTGACTTTCTTCAAGTTTACAGCCCTGAGGCC	CAACATTGGAATGCTGAAGAG		300
Sbjct 241	GGCTCTACTGACTTTCTTCAAGTTTACAGCCCTGAGGCC	CAACATTGGAATGCTGAAGAG		300
Query 301	ATGGAAGCTGGTTGGAGTTAATGAGGCGACCAACATCTG	AAAAAGCTTCAAGCTGACAT		360
Sbjct 301	ATGGAAGCTGGTTGGAGTTAATGAGGCGACTA	AACATCTGAAAAAGCTTCAAGCTGACAT		360
Query 361	TGGACAGATGCTCAGCGGACTGAATAAGCGGAAGGC	GAAACGTC	CGGGGGGGAGTTGCTC	420
Sbjct 361	TGGACAGATGCTCAGCGGACTGAATAAGCGGAAGGC	GAAACGTC	CGGGGGGGAGTTGCTC	420
Query 421	TTGGATCATTATGTTACTCCCGATAGTTGCTGGGCTGAAGCT	TGGAACTATAATGGTAG		480
Sbjct 421	TTGGATCATTATGTTACTCCCGATAGTTGCTGGGCTGAAGCT	TGGAACTATAATGGTAG		480
Query 481	AGTTTTGGCCACTTTAAATAAGACTGATGTATCAGACTT	GCTAGTCATTCCAATAACGGC		540
Sbjct 481	AGTTTTGGCCACTTTAAATAAGACCGATGTATCAGACTT	GCTAGTCATTCCAATAACGGC		540
Query 541	TGGCAGCAATGGATGCGTCTACGTGCTCTAGATGTGGGACT	AATGTGTCAAGATGACAT		600

PRACTICAL 6

Aim: Write a Python/Java code to find motif in a given sequence.

```
import random
l=int(input("Enter the length of motif"))
file=open("mot.txt","r")
r=file.read()
print("Sequence",r)
size=len(r)
print("Size of the sequence",size)
pos=random.randint(0,len(r)-5)
#pos=1
print("Position",pos)
motif=r[pos:pos+l]
print("Motif",motif)
i=pos+1
while(i<=size-1):
    if(motif==r[i:i+1]):
        str1=r[i:i+1]
print("Match motif",str1)
file1=open("motoutput.txt","a")
file1.write(str1+" ")
i+=1
```

OUTPUT:


```
*IDLE Shell 3.11.0*
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/asif0/Desktop/Test.py =====
Enter the length of motif4
Sequence AGAAGTTCGAGAAGCCGTAGT
Size of the sequence 21
Position 6
Motif TCGA
|
Ln: 10 Col: 0
```

PRACTICAL 7

Aim: Perform a BLAST search on any genes sequence and write a java/python code to count the no of repetition of each nucleotide in the sequence.

```
file=open("genes.txt","r")
```

```
r=file.read()
```

```
size=len(r)
```

```
score_A=0
```

```
score_C=0
```

```
score_T=0
```

```
score_G=0
```

```
for i in range(size):
```

```
    if(r[i]=='A'):
```

```
        score_A+=1
```

```
    elif (r[i]=='C'):
```

```
        score_C+=1
```

```
    elif (r[i]=='T'):
```

```
        score_T+=1
```

```
    elif (r[i]=='G'):
```

```
        score_G+=1
```

```
print("score of A is ",score_A)
```

```
print("score of C is ",score_C)
```

```
print("score of T is ",score_T)
```

```
print("score of G is ",score_G)
```

OUTPUT:



The screenshot shows the IDLE Shell 3.11.0 window. The title bar reads "IDLE Shell 3.11.0". The menu bar includes "File", "Edit", "Shell", "Debug", "Options", "Window", and "Help". The main text area displays the following content:

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
===== RESTART: C:/Users/asif0/Desktop/Test.py =====
score of A is 7
score of C is 3
score of T is 4
score of G is 7
>>> |
```

The status bar at the bottom right indicates "Ln: 9 Col: 0".

PRACTICAL 8

Aim: Generate a regular expression enter three protein sequence of three different organism. Write Python/Java code to find regular expression for these sequences.

```
def gen_reg_exp(seq_list, no_of_col):  
    final_list=[]  
    for colnum in range(no_of_col):  
        collist=[]  
        for colseq in seq_list:  
            collist.append(colseq[colnum])  
        if len(set(collist))==len(collist):  
            #print(final_list)  
            final_list.append('x')  
        else:  
            if len(set(collist))==1:  
                final_list.append(collist[0])  
            else:  
                final_list.append("".join(set(collist)))  
            display_output(final_list)  
def display_output(final_list):  
    print(*final_list, sep='-')  
no_of_seq=int(input("Enter the number of sequence: "))  
print("Enter all the sequences")  
seq_list=[]  
for _ in range(no_of_seq):  
    seq_list.append(list(map(str, input("").split())))  
gen_reg_exp(seq_list, len(seq_list[0]))
```

OUTPUT:

```
IDLE Shell 3.11.0
File Edit Shell Debug Options Window Help
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/asif0/Desktop/Test.py =====
Enter the number of sequence: 4
Enter all the sequences
A D L G A V F A L C D R Y F Q
... S D V G P R S C F C E R F Y Q
... A D L G R T Q L R C D R Y Y Q
... A D I G Q P H S L C E R Y F Q
... SA-D-IVL-G-x-x-x-x-FRL-C-ED-R-YF-YF-Q
AS
AS-D-LV
AS-D-LV-G-x-x-x-x-C-ED
AS-D-LV-G-x-x-x-x-C-ED-R-YF
AS-D-LV-G-x-x-x-x-C-ED-R-YF-YF
AS
AS-D-LIV
AS-D-LIV-G-x-x-x-x-RLF
AS-D-LIV-G-x-x-x-x-RLF-C-ED
AS-D-LIV-G-x-x-x-x-RLF-C-ED-R-YF
AS-D-LIV-G-x-x-x-x-RLF-C-ED-R-YF-YF
>>>
```

Ln: 23 Col: 0

PRACTICAL 9

Aim: Enter six protein sequence of different organism and write a program to find a fingerprint of sequence.

```
def solve_fingerprint(seq_list, no_of_col):  
    seq_dict=dict()  
    for colnum in range(no_of_col):  
        counta,countc,countt,countg=0,0,0,0  
        for colseq in seq_list:  
            if colseq[colnum]=='A':  
                counta+=1  
            elif colseq[colnum]=='T':  
                countt+=1  
            elif colseq[colnum]=='C':  
                countc+=1  
            elif colseq[colnum]=='G':  
                countg+=1  
            seq_dict[colnum]=[counta,countc,countt,countg]  
        display_results(seq_dict)  
  
def display_results(seq_dict):  
    print("\tA \tC \tT \tG")  
    for key in seq_dict:  
        print("\n",*seq_dict[key],sep="\t")  
  
no_of_seq=int(input("Enter the number of sequence: "))  
print("Enter all the sequences")  
seq_list=[]  
  
for _ in range(no_of_seq):  
    seq_list.append(list(map(str, input("").split()))))
```

solve_fingerprint(seq_list,len(seq_list[0]))

OUTPUT:

```
Python 3.11.0 (main, Oct 24 2022, 18:26:48) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>>
===== RESTART: C:/Users/asiFO/Desktop/Test.py =====
Enter the number of sequence: 4
Enter all the sequences
A C T G A T G
... AT C A G A A
... AT A A G C A
... A G T T A G C

      A      C      T      G
      0      0      0      1
      A      C      T      G
      0      0      0      1
      0      0      0      1
      A      C      T      G
      0      0      0      1
      A      C      T      G
      0      0      0      1
      1      0      0      1
      A      C      T      G
      0      0      0      1
      1      0      0      1
      0      0      0      1
      A      C      T      G
      0      0      0      1
      A      C      T      G
      0      0      0      1
      1      0      0      1
      A      C      T      G
      0      0      0      1
      1      0      0      2
      A      C      T      G
      0      0      0      1
```

PRACTICAL 10

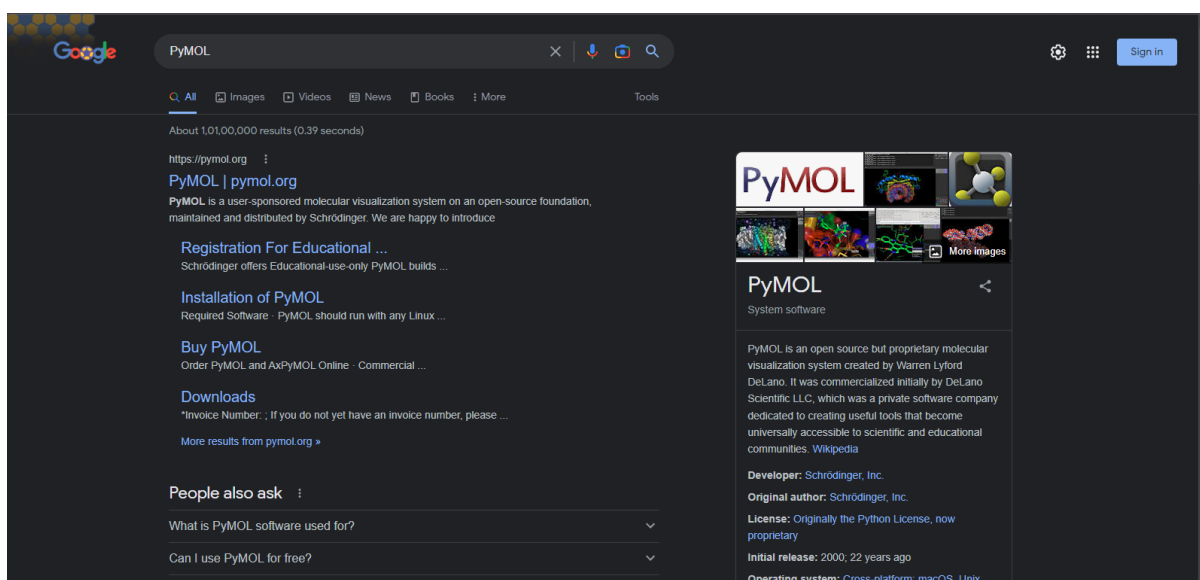
Aim: Retrieving 3D structure from PDB

To perform the current practical, you'll be needing two things

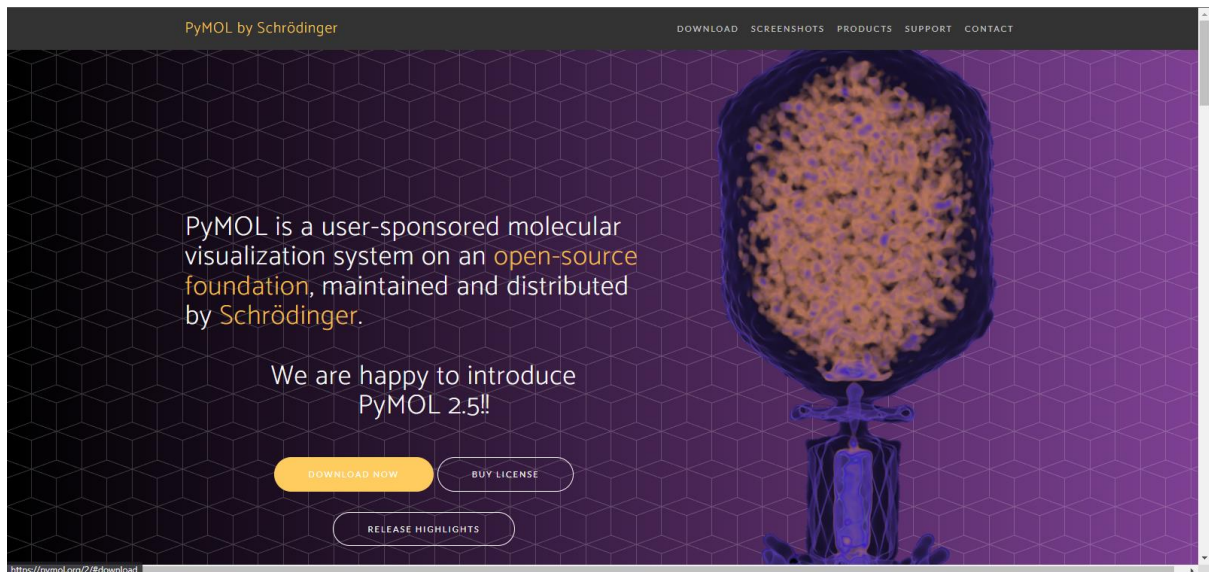
- I. PyMOL (software)
- II. Protein in .pdb format

Installing PyMOL Software

First, we need to install PyMOL. To do so open google and simply search PyMOL.

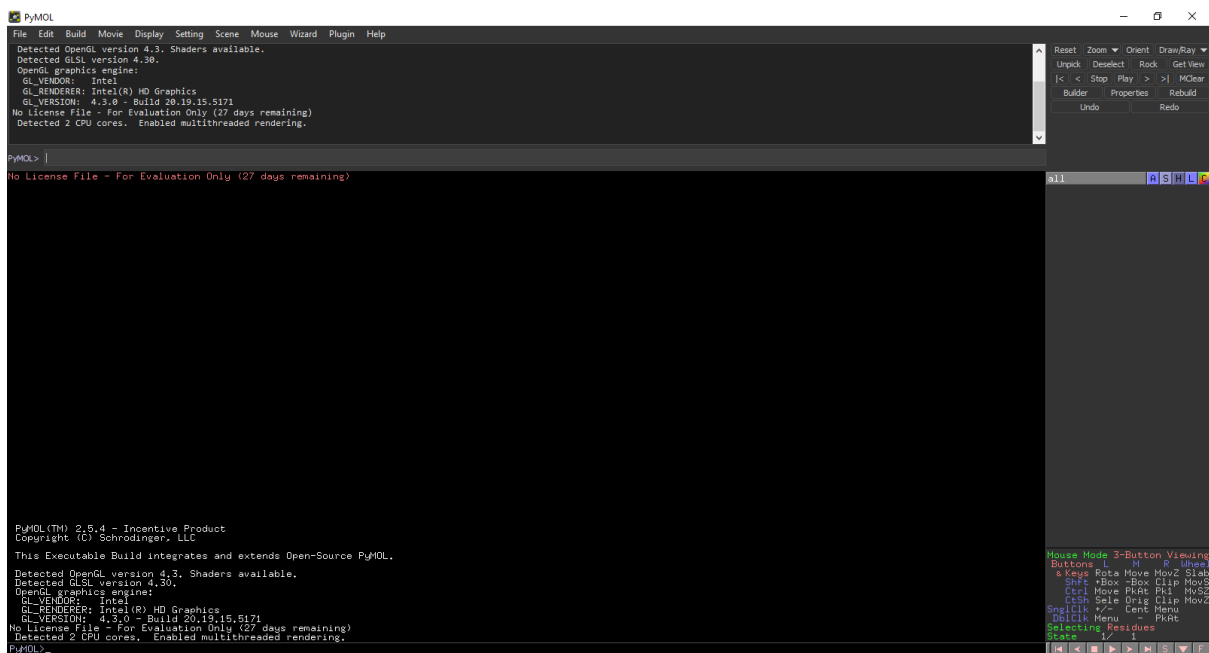


Open the first link and click Download Now.



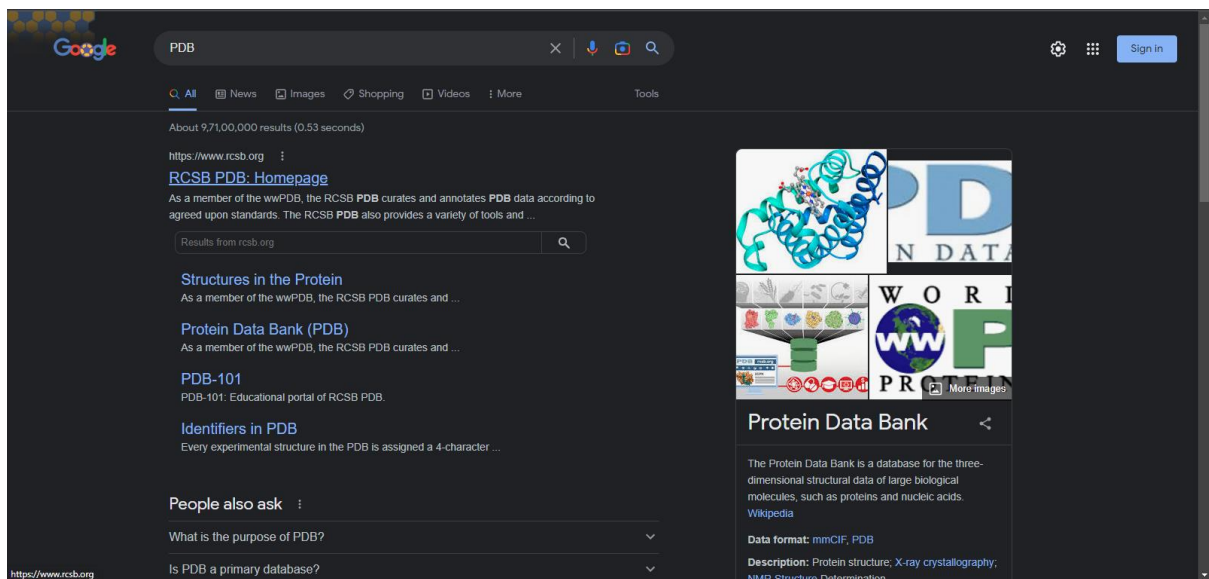
When done downloading install the software.

This is how the interface of the software looks like.



Downloading Protein in .pdb format

To download Protein open, google and search pdb or simply visit <https://www.rcsb.org/>



Open the first link named “RCSB PDB: Homepage”

On the search bar search for “hemoglobin”



You’ll see all the proteins listed below like this.

RCSB PDB Deposit Search Visualize Analyze Download Learn More Documentation Careers MyPDB Contact us

Structure Determination Methodology
☐ experimental (3,451)

Scientific Name of Source Organism
☐ Homo sapiens (773)
☐ Escherichia coli (115)
☐ Bos taurus (88)
☐ Mus musculus (88)
☐ Anadara inaequalis (57)
☐ Saccharomyces cerevisiae (55)
☐ Pylaster carolinianus (54)
☐ Equus caballus (51)
☐ Amphibia ornata (50)
☐ Adromobacter xylosoxidans (51)
[More...](#)

Taxonomy
☐ Eukaryota (1,878)
☐ Bacteria (1,504)
☐ Archaea (41)
☐ Ribozyme (38)
☐ other sequences (34)
☐ Drosophila (12)
☐ unclassified sequences (8)
☐ Virogenia (3)
☐ Monodnaviria (2)
☐ Nidovirales (2)

Experimental Method
☐ X-RAY DIFFRACTION (3,196)
☐ ELECTRON MICROSCOPY (132)
☐ SOLUTION NMR (118)
☐ NEUTRON DIFFRACTION (4)
☐ THEORETICAL MODEL (3)
☐ SOLUTION SCATTERING (2)
☐ ELECTRON CRYSTALLOGRAPHY (1)

Polymer Entity Type
☐ Protein (3,451)

1 to 25 of 3,451 Structures Page 1 of 132 25 Sort by Score

2GTL
 Lumbricus Erythrocyruin at 3.5Å resolution
 Royer Jr. W.E., Sharma, H., Strand, K., Knapp, J.E., Bhayrabhatla, B.
 (2006) Structure 14: 1167-1177
 Released 2008-07-18
 Method X-RAY DIFFRACTION 3.5 Å
 Organisms Lumbricus terrestris
 Macromolecule Unique protein chains: 7
 Unique Ligands CA, CMO, HEM, ZN

1HV4
 CRYSTAL STRUCTURE ANALYSIS OF BAR-HEAD GOOSE HEMOGLOBIN (DEOXY FORM)
 Liang, Y., Hua, Z., Liang, X., Xu, Q., Lu, G.
 (2001) J Mol Biol 313: 123-137
 Released 2001-01-17
 Method X-RAY DIFFRACTION 2.8 Å
 Organisms Anser indicus
 Macromolecule HEMOGLOBIN ALPHA-A CHAIN (protein)
 HEMOGLOBIN BETA CHAIN (protein)
 Unique Ligands HEM

1SI4
 Crystal structure of Human hemoglobin A2 (in R2 state) at 2.2 Å resolution
 Sen, U., Dasgupta, J., Choudhury, D., Datta, P., Chakrabarti, A., Chakrabarty, S.B., Chakrabarty, A., Dattagupta, J.K.
 (2004) Biochemistry 43: 12477-12488
 Released 2004-10-26
 Method X-RAY DIFFRACTION 2.2 Å
 Organisms Homo sapiens
 Macromolecule Hemoglobin alpha chain (protein)
 Hemoglobin delta chain (protein)
 Unique Ligands CYN, HEM

Scroll down until you find “4YU3” and open it or you can directly search for “6otw”

RCSB PDB Deposit Search Visualize Analyze Download Learn More Documentation Careers MyPDB Contact us

4YU3
 Crystal structure of Human hemoglobin A2 (in R2 state) at 2.2 Å resolution
 Sen, U., Dasgupta, J., Choudhury, D., Datta, P., Chakrabarti, A., Chakrabarty, S.B., Chakrabarty, A., Dattagupta, J.K.
 (2004) Biochemistry 43: 12477-12488
 Released 2004-10-26
 Method X-RAY DIFFRACTION 2.2 Å
 Organisms Homo sapiens
 Macromolecule Hemoglobin alpha chain (protein)
 Hemoglobin delta chain (protein)
 Unique Ligands CYN, HEM

6OTW
 Crystallographic Structure of (HbII-HbIII)-O2 from Lucina pectinata at pH 5.0
 Marchany-Rivera, D., Smith, C.A., Rodriguez-Perez, J.D., Lopez-Garriga, J.
 (2020) J Inorg Biochem 207: 111055-111055
 Released 2020-04-01
 Method X-RAY DIFFRACTION 2.447 Å
 Organisms Phacoides pectinatus
 Macromolecule Hemoglobin II (protein)
 Hemoglobin III (protein)
 Unique Ligands HEM

6OTX
 Crystallographic Structure of (HbII-HbIII)-O2 from Lucina pectinata at pH 7.0
 Marchany-Rivera, D., Smith, C.A., Rodriguez-Perez, J.D., Lopez-Garriga, J.
 (2020) J Inorg Biochem 207: 111055-111055
 Released 2020-04-01
 Method X-RAY DIFFRACTION 2.539 Å
 Organisms Phacoides pectinatus
 Macromolecule Hemoglobin II (protein)
 Hemoglobin III (protein)
 Unique Ligands HEM, OXY

6OTY

<https://www.rcsb.org/structure/6OTW>

On the right-hand side, you’ll the download option. Click on it and download as PDB format.

RCSB PDB Deposit Search Visualize Analyze Download Learn More Documentation Careers MyPDB Contact us

RCSB PDB PROTEIN DATA BANK 198,165 Structures from the PDB 1,000,361 Computed Structure Models (CSM)

3D Structures Enter search term(s), Entry ID(s), or sequence Include CSM Help

Advanced Search Browse Annotations

PDB-101 PDB EMDB Resource PDBe KEGG PATHWAY

Structure Summary 3D View Annotations Experiment Sequence Genome Ligands Versions

Biological Assembly 1

6OTW

Crystallographic Structure of (HbII-HbIII)-O2 from Lucina pectinosa

PDB DOI: 10.2210/pdb6OTW/pdb

Classification: OXYGEN TRANSPORT

Organism(s): Phacoides pectinatus

Expression System: Phacoides pectinatus

Mutation(s): No

Deposited: 2019-05-03 Released: 2020-04-01

Deposition Author(s): Maichany-Rivera, D., Smith, C.A., Rodriguez-Perez, J.

Funding Organization(s): National Science Foundation (NSF, United States)

Experimental Data Snapshot

Method: X-RAY DIFFRACTION

Resolution: 2.45 Å

R-Value Free: 0.248

R-Value Work: 0.190

R-Value Observed: 0.193

wwPDB Validation

Metric Rtfree Clashscore Ramachandran outliers Sidechain outliers RSQR outliers

Validation Full PDF Validation XML

Biological Assembly 1 (CIF - gz)

Biological Assembly 1 (PDB - gz)

Display Files Download Files

FASTA Sequence

PDBx/mmCIF Format

PDBx/mmCIF Format (gz)

PDB Format

PDB Format (gz)

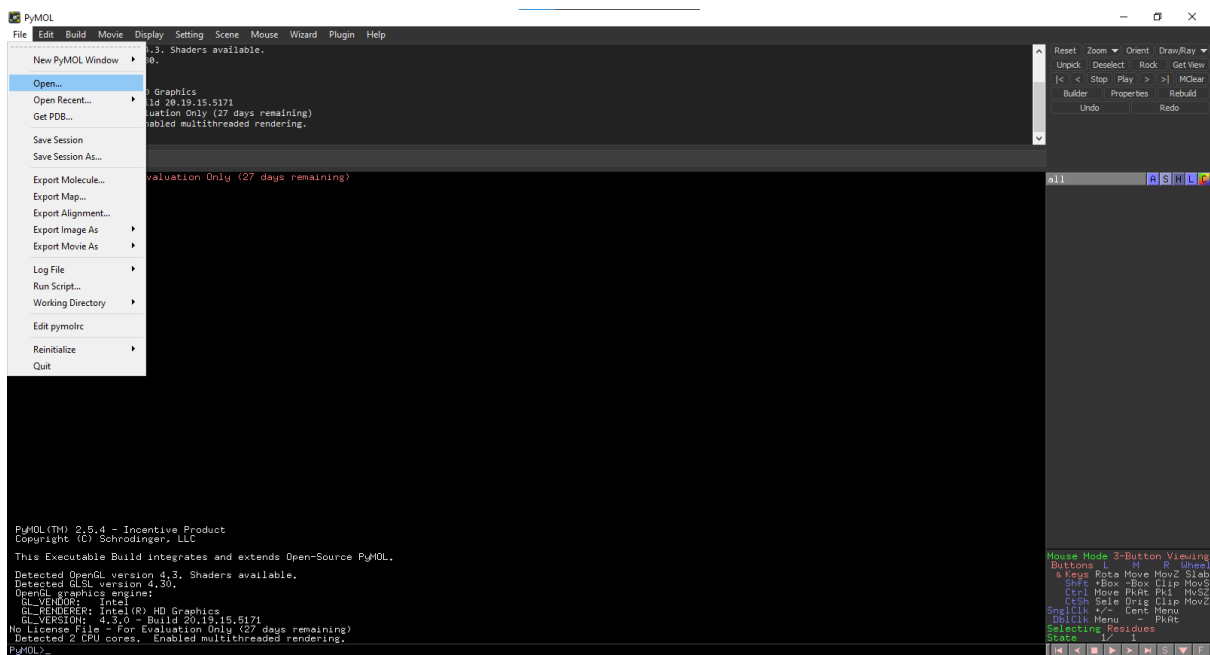
PDBML/XML Format (gz)

Structure Factors (CIF)

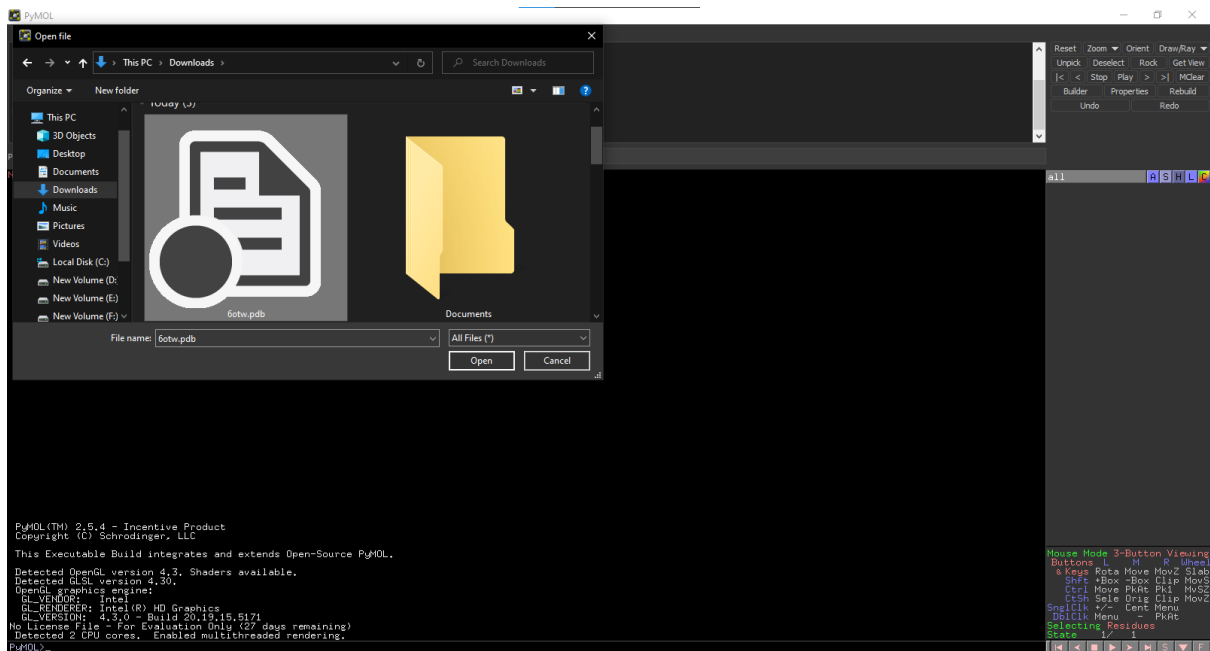
Structure Factors (CIF - gz)

https://files.rcsb.org/download/6OTW.pdb

Now that protein is downloaded open PyMOL and on top left corner click File > open



Now browse for the file you just downloaded (6otw.pdb)



This is how the screen of imported file looks like...

