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		

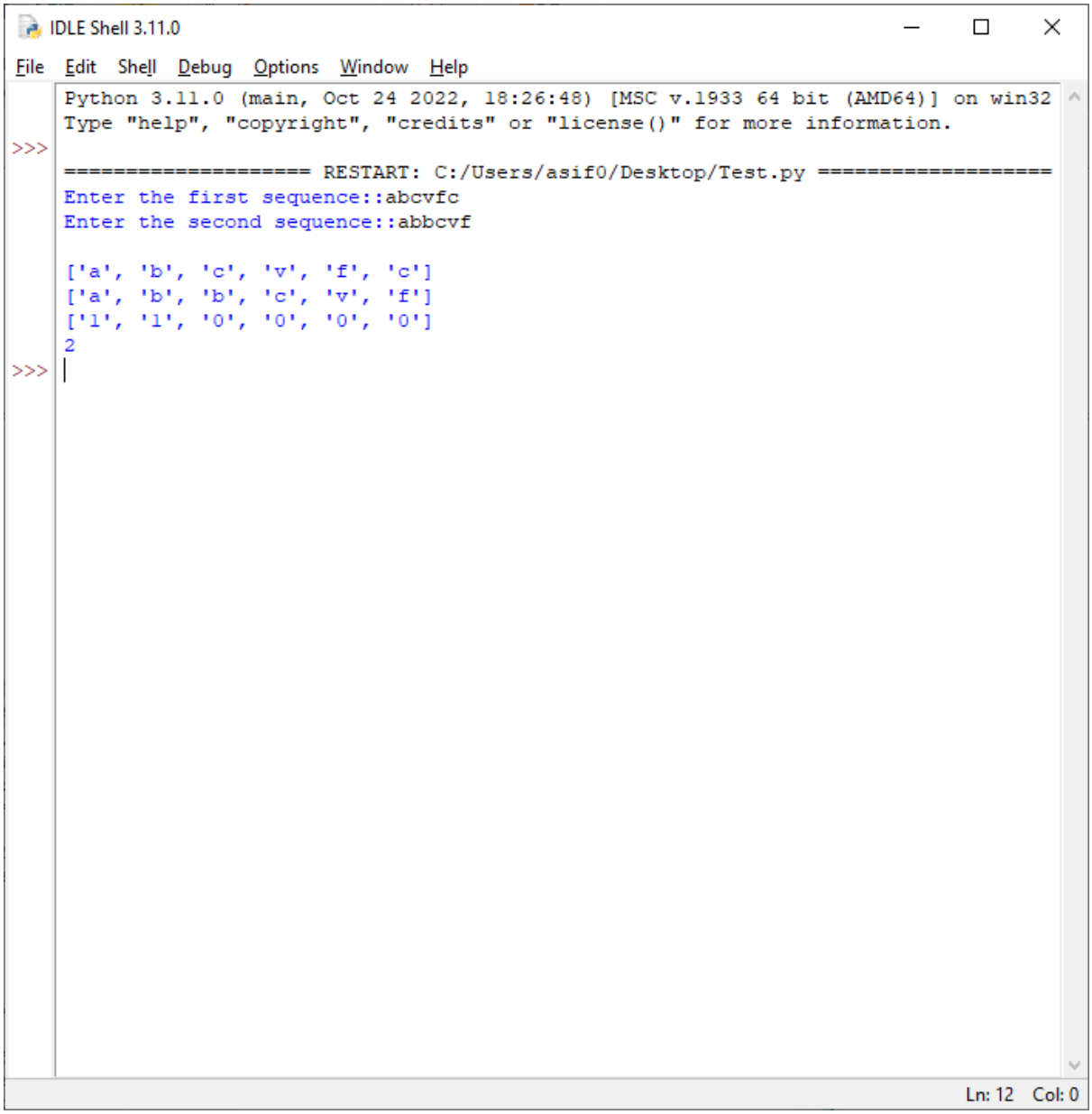
PRACTICAL 1

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:



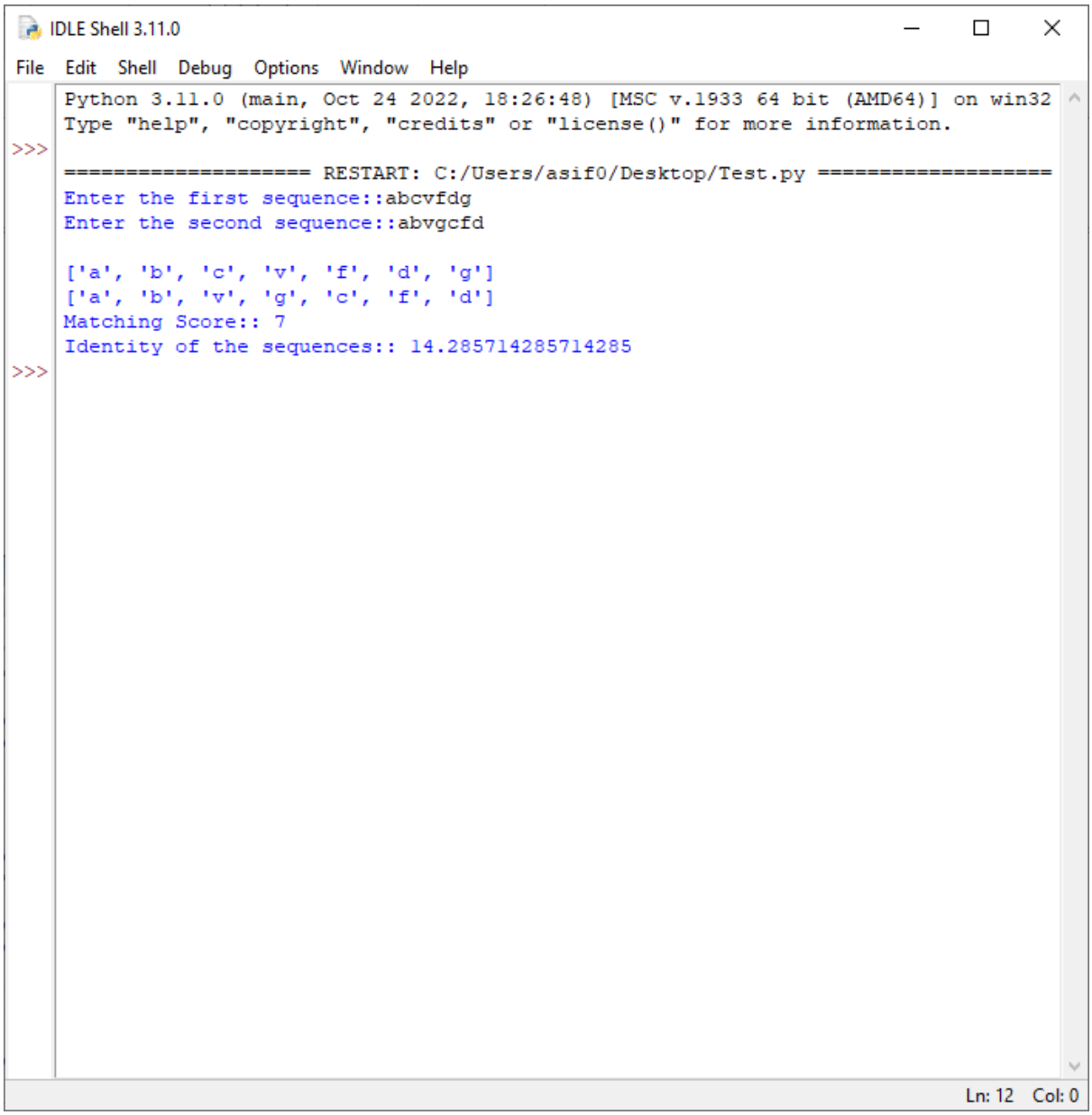
PRACTICAL 2

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:



PRACTICAL 3

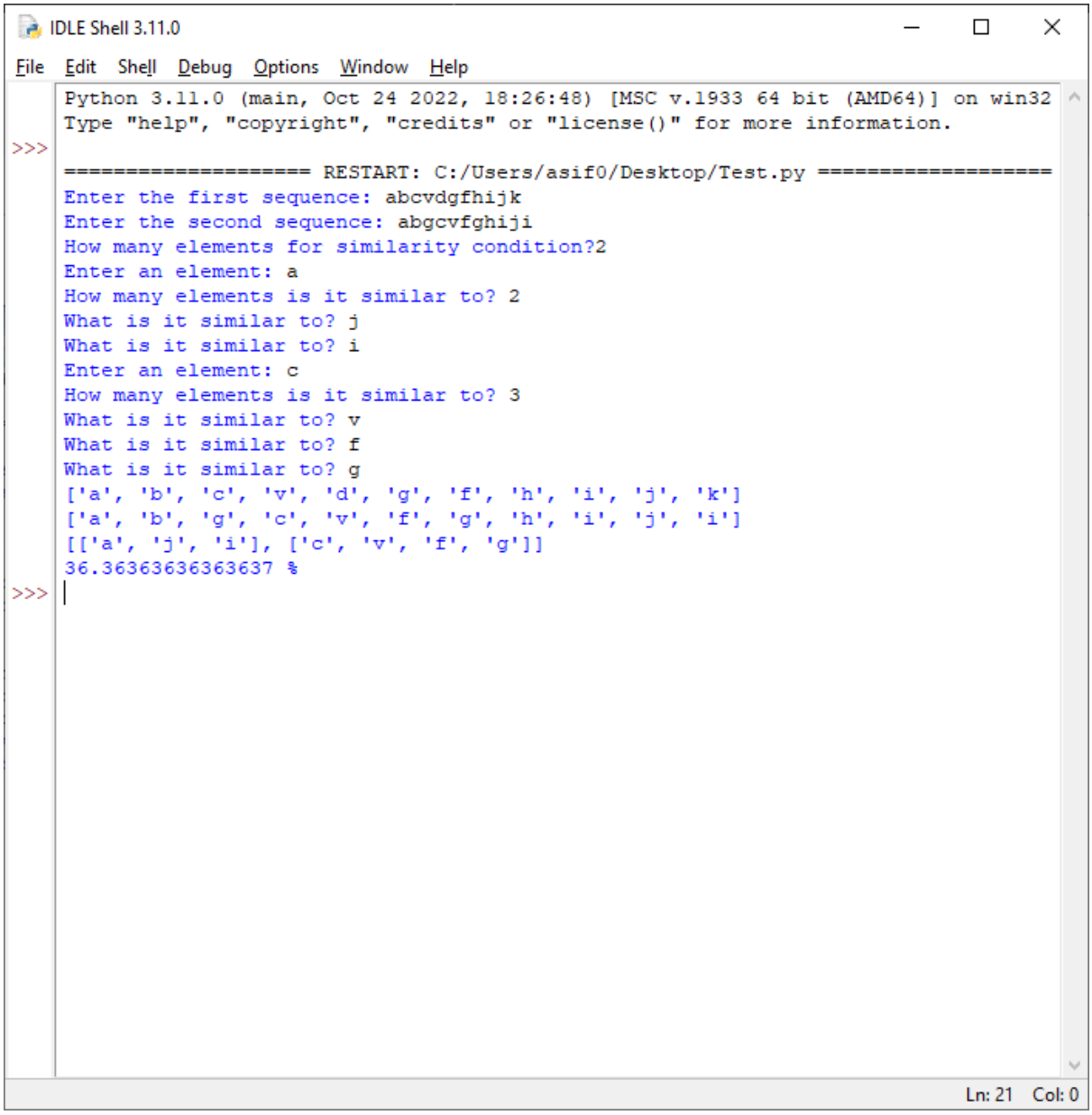
Aim: Write a Python/Java code to find the Similarity value of a given sequences. Take the sequence from user.

```
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:



PRACTICAL 4

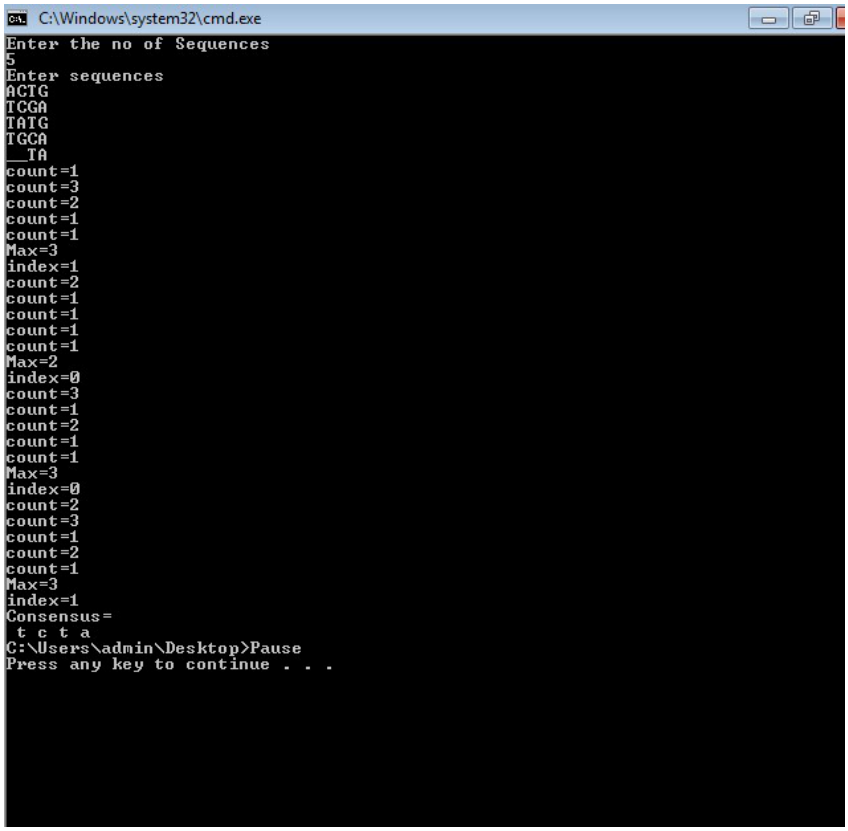
Aim: Enter genome of five different organism and write a python/java program to find consensus sequence using Multiple Sequence Alignment (MSA) technique.

```
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:



PRACTICAL 5

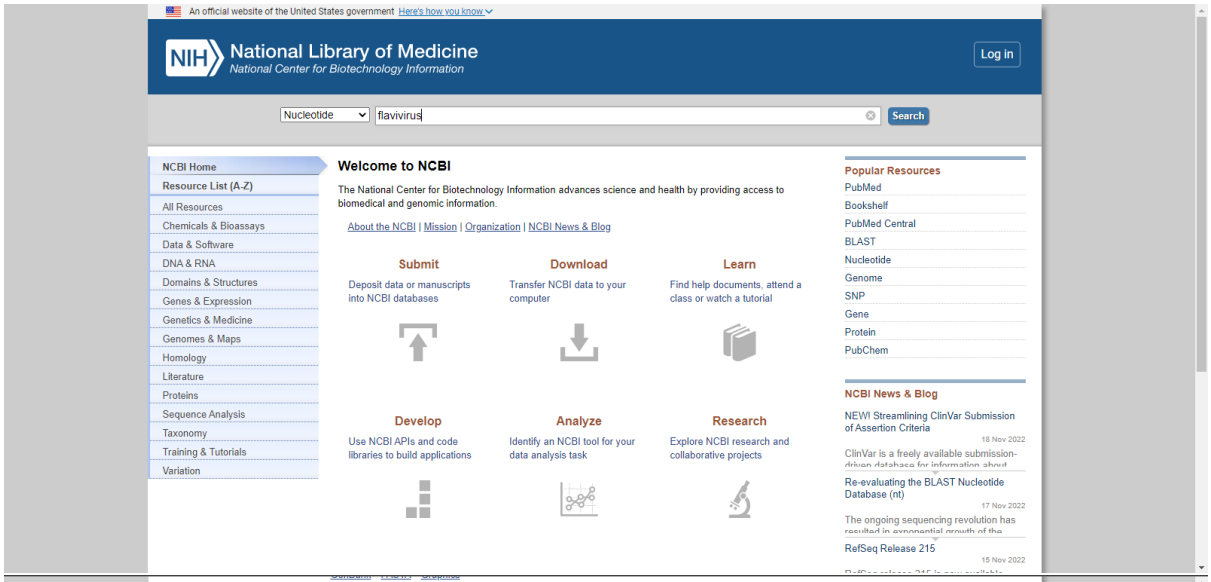
Aim: Perform a BLAST search on a specific gene sequence of a specify organism.

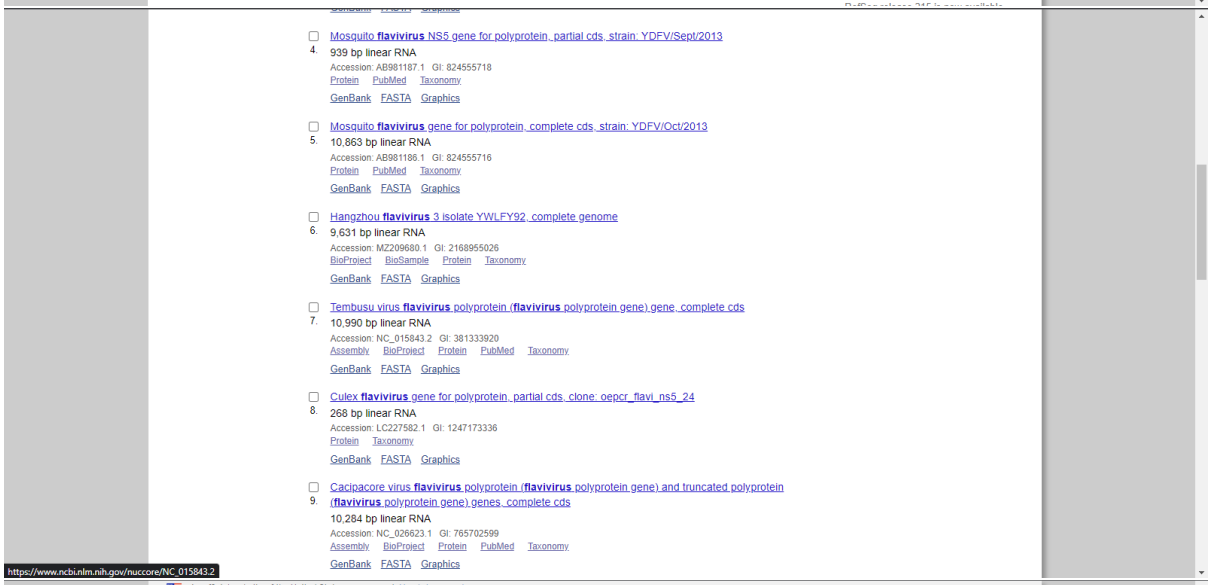
Steps:

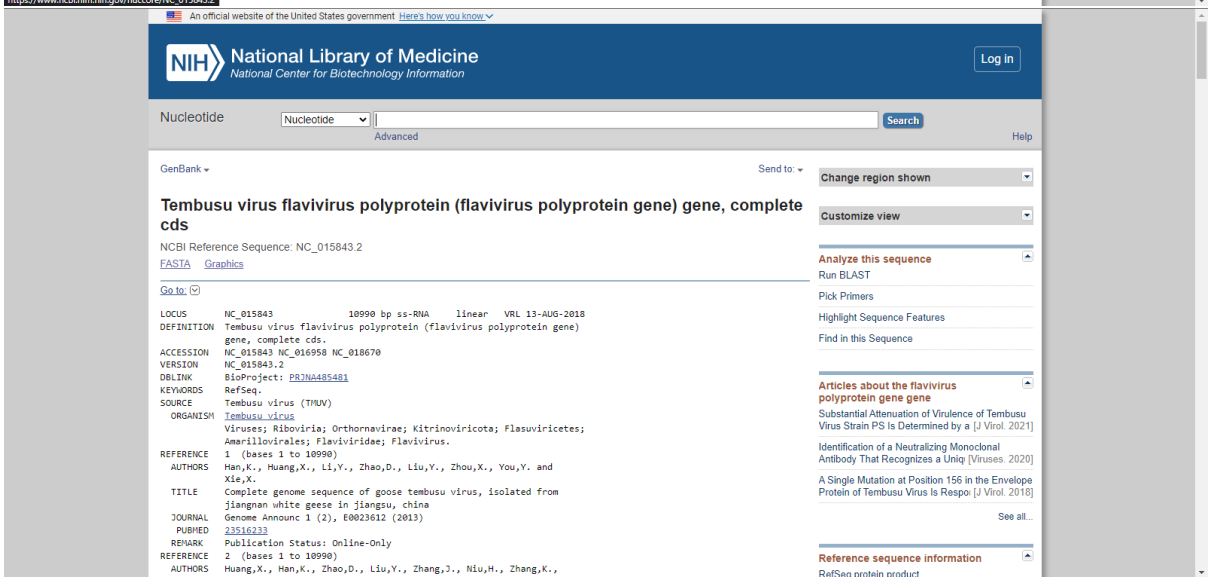
Go to the National Center for Biotechnology Information Site

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

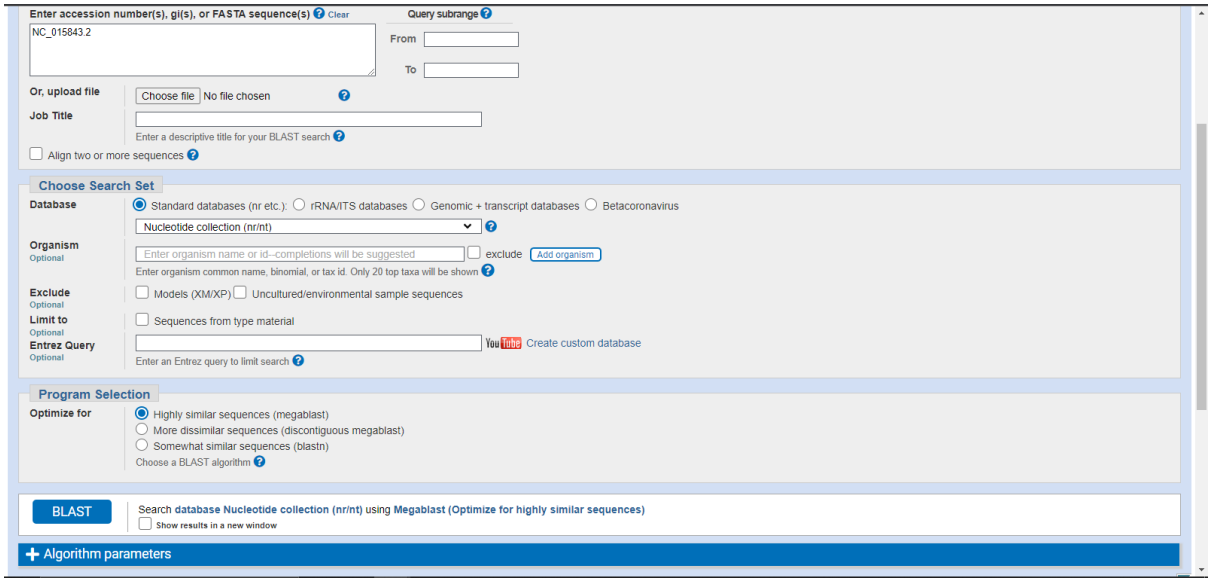
Select Nucleotide from All Databases and find any organism in a search bar



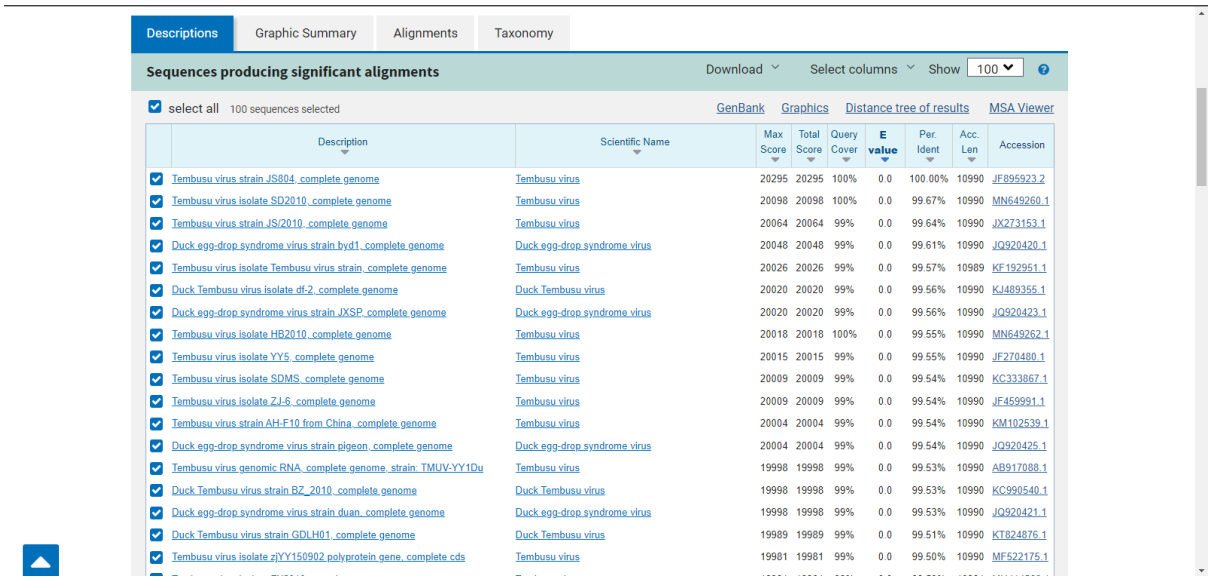




Run BLAST option we have to select



BLAST



Description	Scientific Name	Max Score	Total Score	Query Cover	E value	Per. Ident	Acc. Len	Accession
Tembusu virus strain JS804, complete genome	Tembusu virus	20295	20295	100%	0.0	100.00%	10990	JF895923.2
Tembusu virus isolate SD2010, complete genome	Tembusu virus	20098	20098	100%	0.0	99.67%	10990	MN649260.1
Tembusu virus strain JS/2010, complete genome	Tembusu virus	20064	20064	99%	0.0	99.64%	10990	JX273153.1
Duck egg-drop syndrome virus strain byd1, complete genome	Duck egg-drop syndrome virus	20048	20048	99%	0.0	99.61%	10990	JQ920420.1
Tembusu virus isolate Tembusu virus strain, complete genome	Tembusu virus	20026	20026	99%	0.0	99.57%	10989	KF192951.1
Duck Tembusu virus isolate df-2, complete genome	Duck Tembusu virus	20020	20020	99%	0.0	99.56%	10990	KJ489355.1
Duck egg-drop syndrome virus strain JXSP, complete genome	Duck egg-drop syndrome virus	20020	20020	99%	0.0	99.56%	10990	JQ920423.1
Tembusu virus isolate HB2010, complete genome	Tembusu virus	20018	20018	100%	0.0	99.55%	10990	MN649262.1
Tembusu virus isolate YY5, complete genome	Tembusu virus	20015	20015	99%	0.0	99.55%	10990	JF270480.1
Tembusu virus isolate SDMS, complete genome	Tembusu virus	20009	20009	99%	0.0	99.54%	10990	KC333867.1
Tembusu virus isolate ZJ-6, complete genome	Tembusu virus	20009	20009	99%	0.0	99.54%	10990	JF459991.1
Tembusu virus strain AH-F10 from China, complete genome	Tembusu virus	20004	20004	99%	0.0	99.54%	10990	KM102539.1
Duck egg-drop syndrome virus strain pigeon, complete genome	Duck egg-drop syndrome virus	20004	20004	99%	0.0	99.54%	10990	JQ920425.1
Tembusu virus genomic RNA, complete genome, strain: TMUV-YY1Du	Tembusu virus	19998	19998	99%	0.0	99.53%	10990	AB917088.1
Duck Tembusu virus strain BZ_2010, complete genome	Duck Tembusu virus	19998	19998	99%	0.0	99.53%	10990	KC990540.1
Duck egg-drop syndrome virus strain duan, complete genome	Duck egg-drop syndrome virus	19998	19998	99%	0.0	99.53%	10990	JQ920421.1
Duck Tembusu virus strain GDLH01, complete genome	Duck Tembusu virus	19989	19989	99%	0.0	99.51%	10990	KT824876.1
Tembusu virus isolate zjYY150902 polyprotein gene, complete cds	Tembusu virus	19981	19981	99%	0.0	99.50%	10990	MF522175.1

Here the result will be display

Download GenBank Graphics ▼ Next ▲ Previous ◀◀ Descriptions

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      AGAAGTTCGCCTGTGTGAACTTATTCCAAACAGCTTTTGGAGTAGTGCGTGTGAACGTAA  60
              |||||||| |||||||||||||||||||||||| |||||||||||||||||||||||||||
Sbjct  1      AGAAGTTCATCTGTGTGAACTTATTCCAAACAGTTTTTGGAGTAGTGCGTGTGAACGTAA  60

Query  61     ACACAGTTTGAACGTTTTTTGGATAGAGACAACTATGTCTAACAAAAAACCAGGAAGACC  120
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  61     ACACAGTTTGAACGTTTTTTGGATAGAGACAACTATGTCTAACAAAAAACCAGGAAGACC  120

Query  121    CGGCTCAGGCCGGGTTGTCAATATGCTAAAGCGCGGAACGTCCCGCGGAAATCCGCTAGC  180
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  121    CGGCTCAGGCCGGGTTGTCAATATGCTAAAGCGCGGAACGTCCCGCGGAAATCCGCTAGC  180

Query  181    GCGGATAAAGAGGACGATTGATGGGGTCCTGAGAGGAGCAGGACCCATAAGGTTTGTGCT  240
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  181    GCGGATAAAGAGGACGATTGATGGGGTCCTGAGAGGAGCAGGACCCATAAGGTTTGTGCT  240

Query  241    GGCTCTACTGACTTTCTTCAAGTTTACAGCCCTGAGGCCAACCATTGGAATGCTGAAGAG  300
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  241    GGCTCTACTGACTTTCTTCAAGTTTACAGCCCTGAGGCCAACCATTGGAATGCTGAAGAG  300

Query  301    ATGGAAGCTGGTTGGAGTTAATGAGGCGACCAAACATCTGAAAAGCTTCAAGCGTGACAT  360
              |||||||||||||||||||||||||||| |||||||||||||||||||||||||||||||
Sbjct  301    ATGGAAGCTGGTTGGAGTTAATGAGGCGACTAAACATCTGAAAAGCTTCAAGCGTGACAT  360

Query  361    TGGACAGATGCTCGACGGACTGAATAAGCGGAAGGCGAAACGTCgggggggAGTTGCTC  420
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  361    TGGACAGATGCTCGACGGACTGAATAAGCGGAAGGCGAAACGTCGGGGGGGGAGTTGCTC  420

Query  421    TTGGATCATTATGTTACTCCCGATAGTTGCTGGGCTGAAGCTTGGAAACTATAATGGTAG  480
              ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  421    TTGGATCATTATGTTACTCCCGATAGTTGCTGGGCTGAAGCTTGGAAACTATAATGGTAG  480

Query  481    AGTTTTGGCCACTTTAAATAAGACTGATGTATCAGACTTGCTAGTCATTCCAATAACGGC  540
              |||||||||||||||||||||||| |||||||||||||||||||||||||| ||||||||
Sbjct  481    AGTTTTGGCCACTTTAAATAAGACCGATGTATCAGACTTGCTAGTCATTCCAACAACGGC  540

Query  541    TGGCAGCAATGGATGCGTCGTACGTGCTCTAGATGTGGGACTAATGTGTCAGGATGACAT  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 writer 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:

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 ====================
score of A is  7
score of C is  3
score of T is  4
score of G is  7
>>>
```

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-x-C-ED
AS-D-LV-G-x-x-x-x-x-C-ED-R-YF
AS-D-LV-G-x-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:

```
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 C T G A T G
... A T C A G A A
... A T 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
Ln: 96  Col: 0
```

PRACTICAL 10

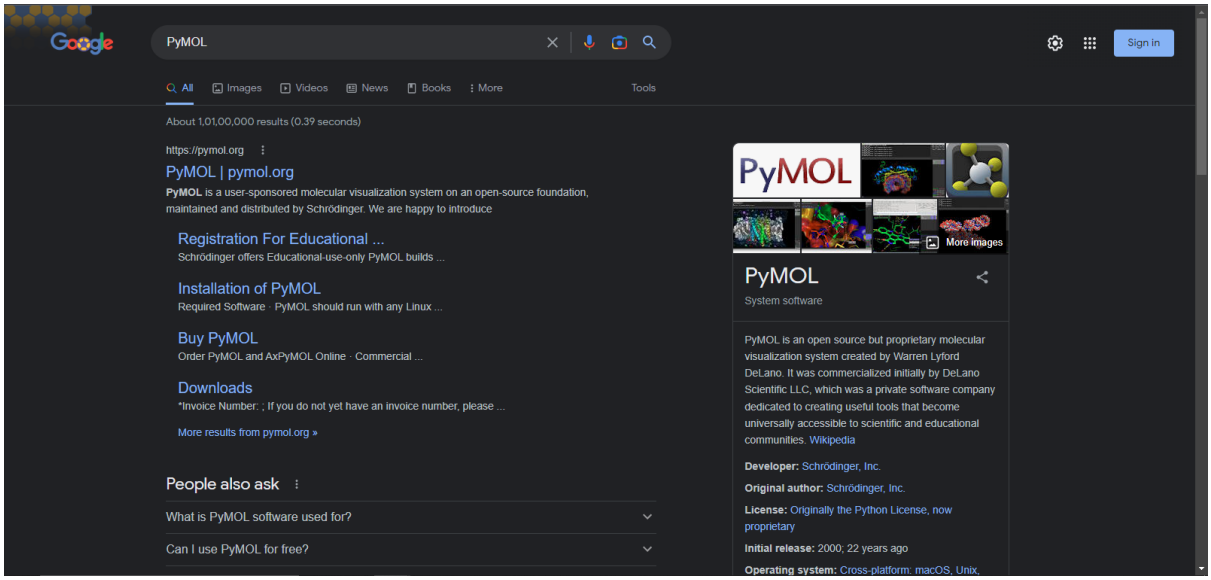
Aim: Retrieving 3D structure from PDB

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

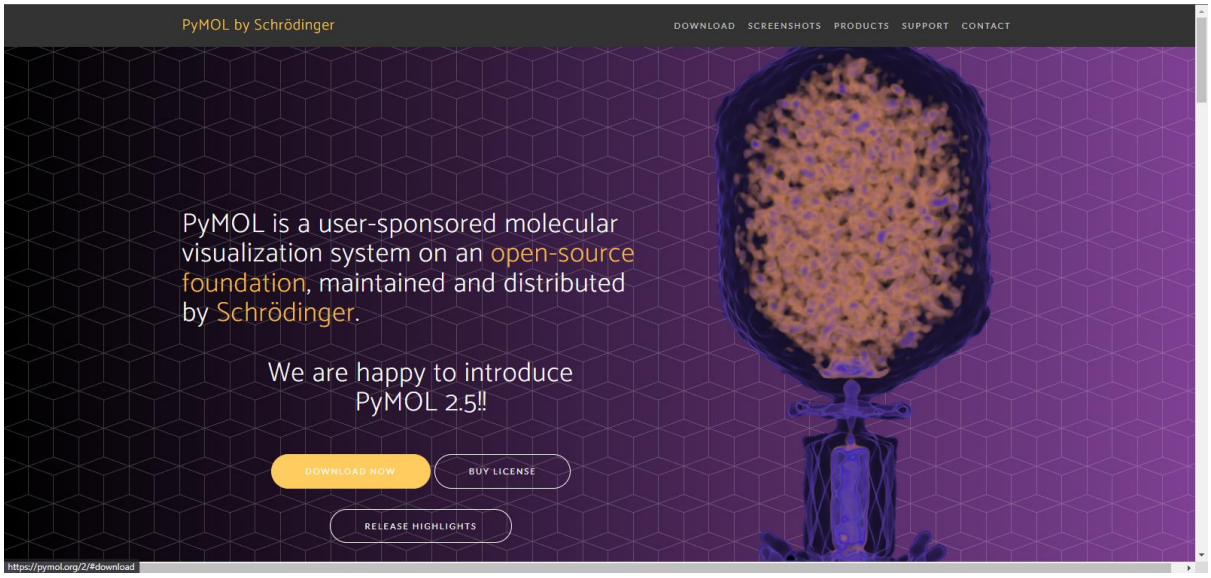
I. PyMOL (software)
II. Protein in .pdb format

Installing PyMOL Software

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

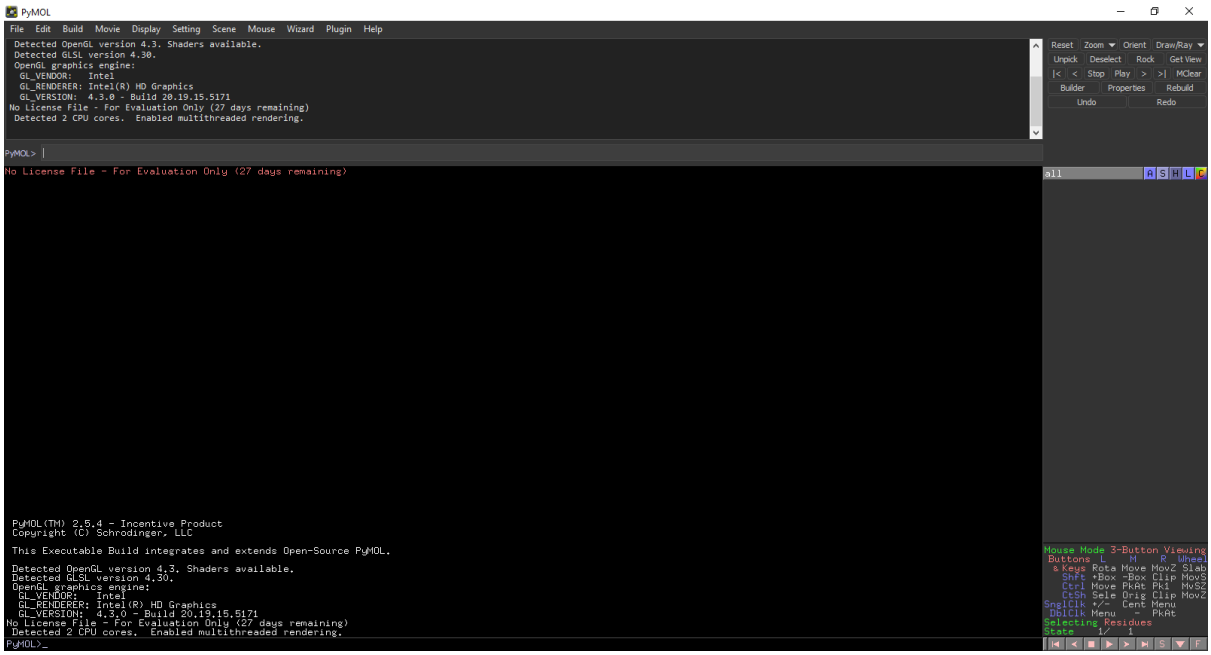


Open the first link and click Download Now.



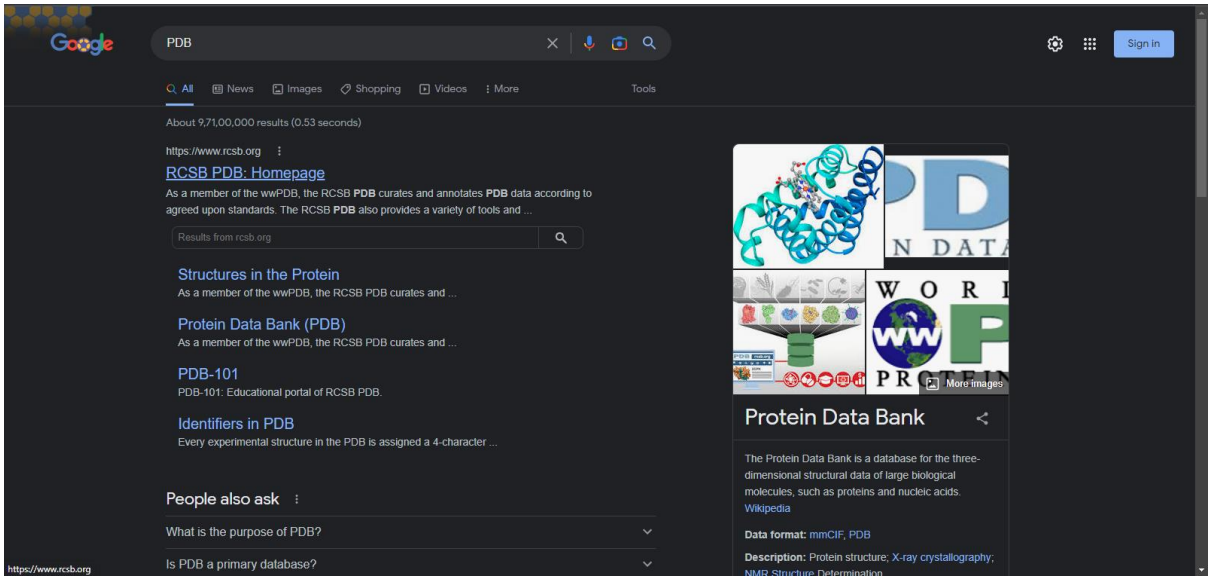
When done downloading install the software.

This is how the interface of the software looks like.



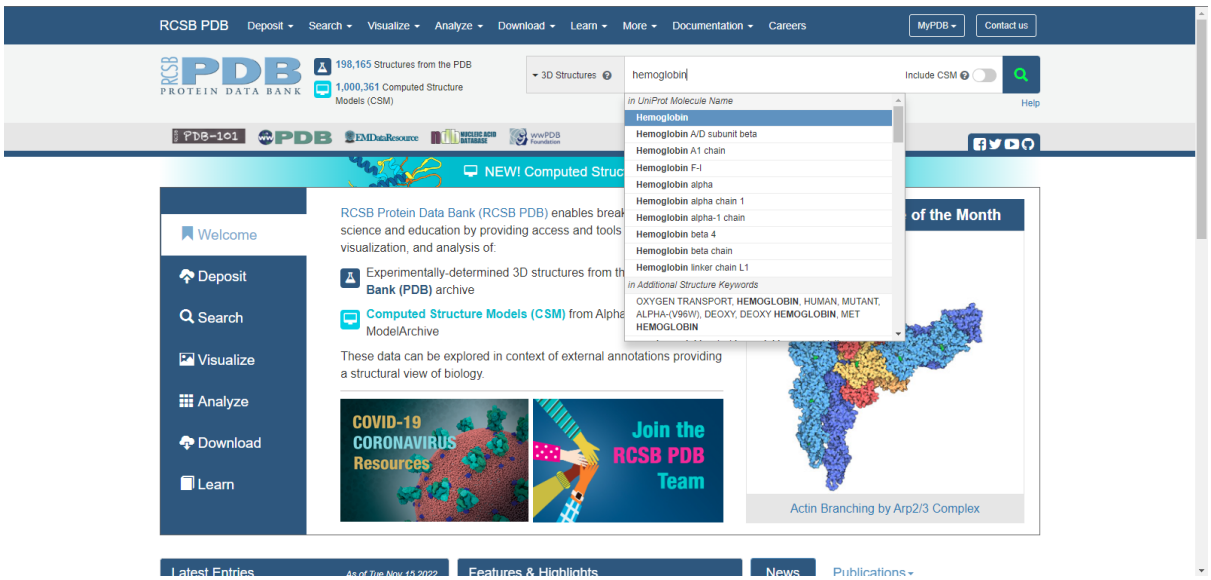
Downloading Protein in .pdb format

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

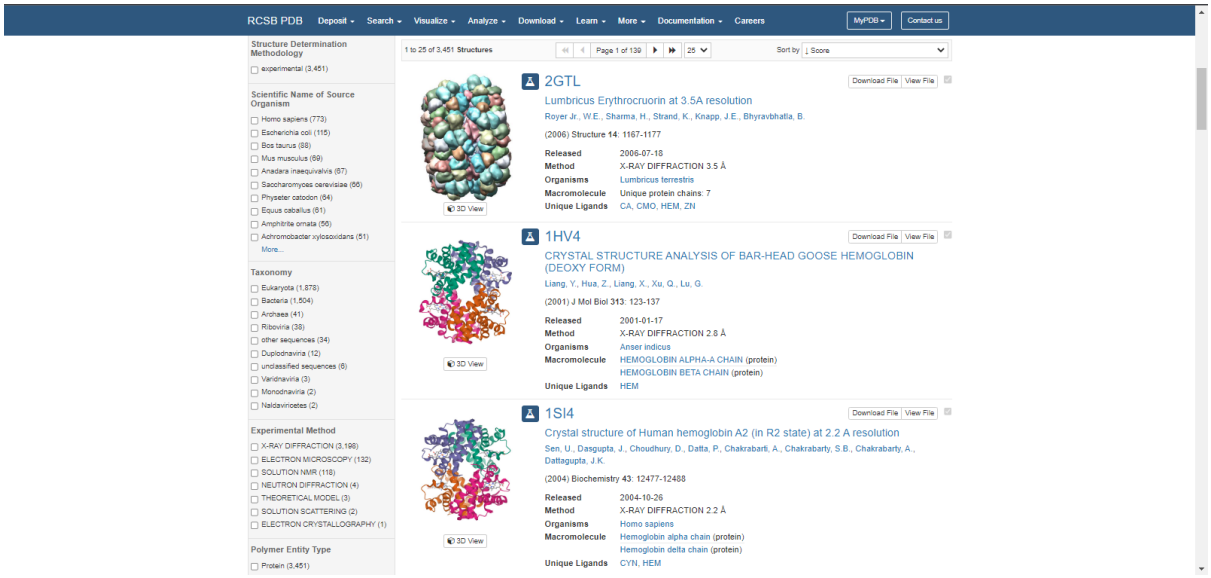


Open the first link named “RCSB PDB: Homepage”

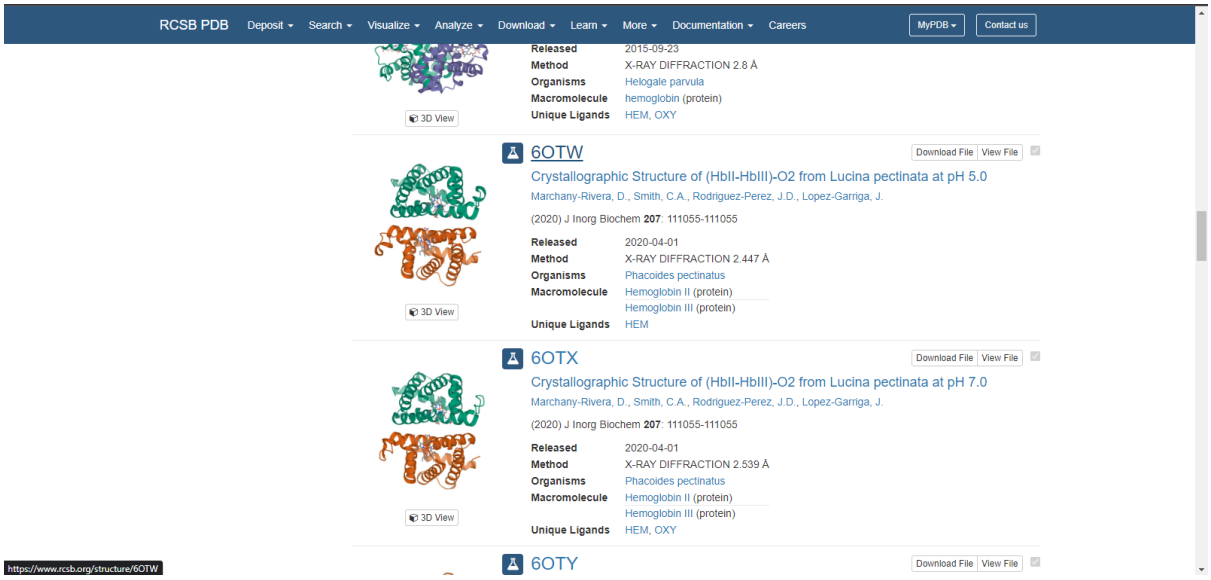
On the search bar search for “hemoglobin”



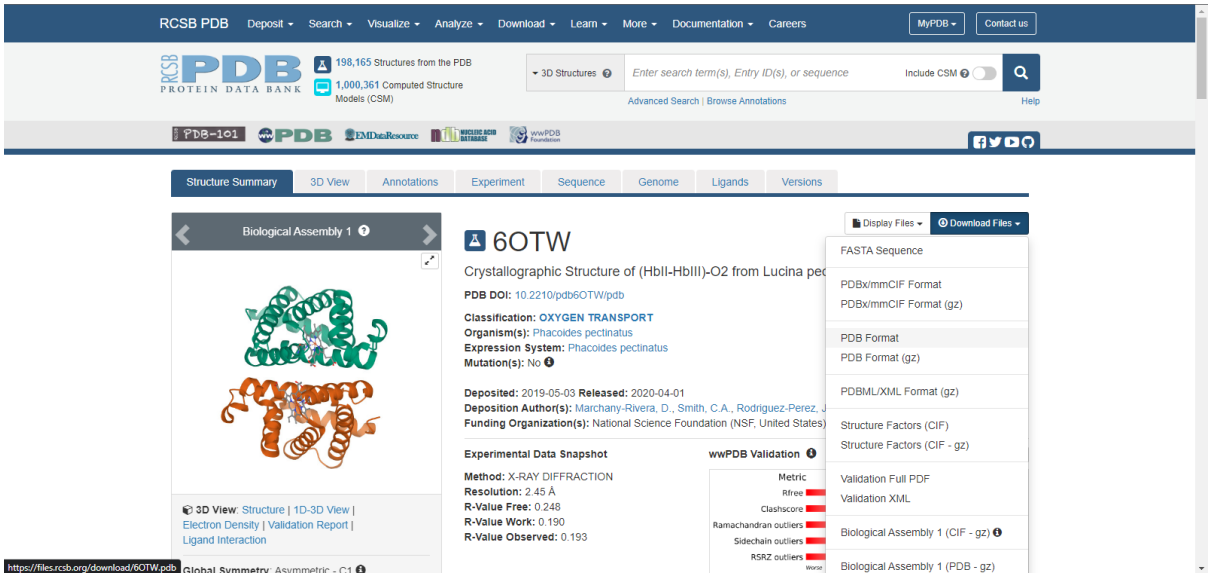
You’ll see all the proteins listed below like this.



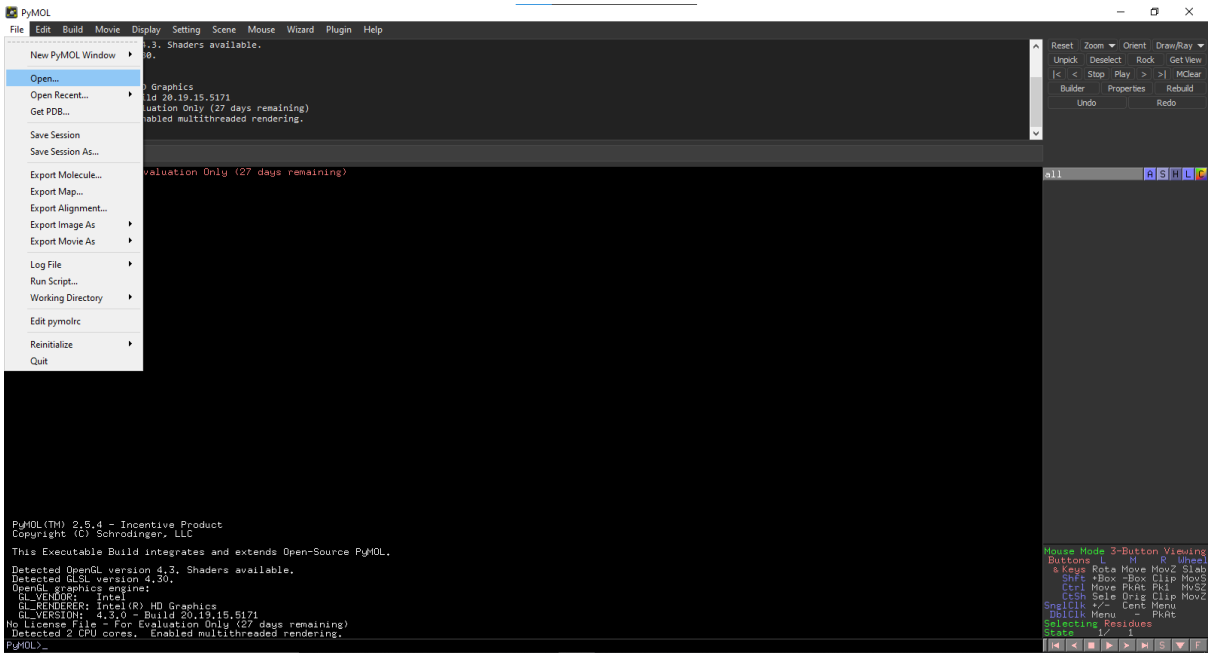
Scroll down until you find “4YU3” and open it or you can directly search for “6otw”



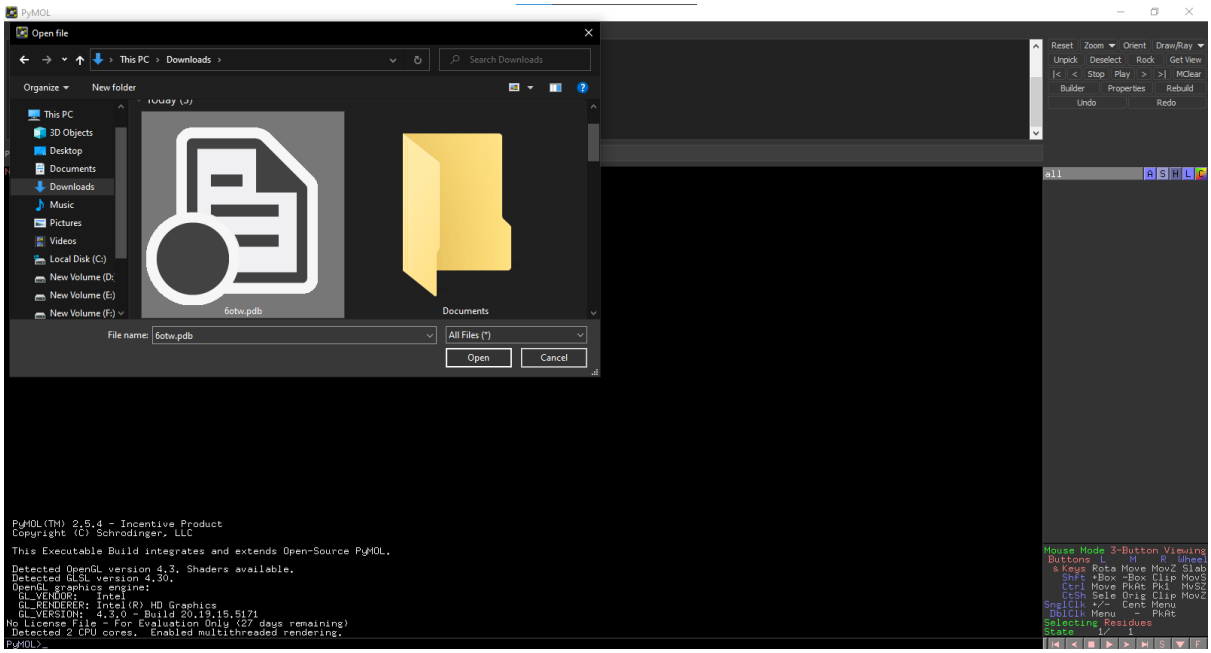
On the right-hand side, you’ll the download option. Click on it and download as PDB format.



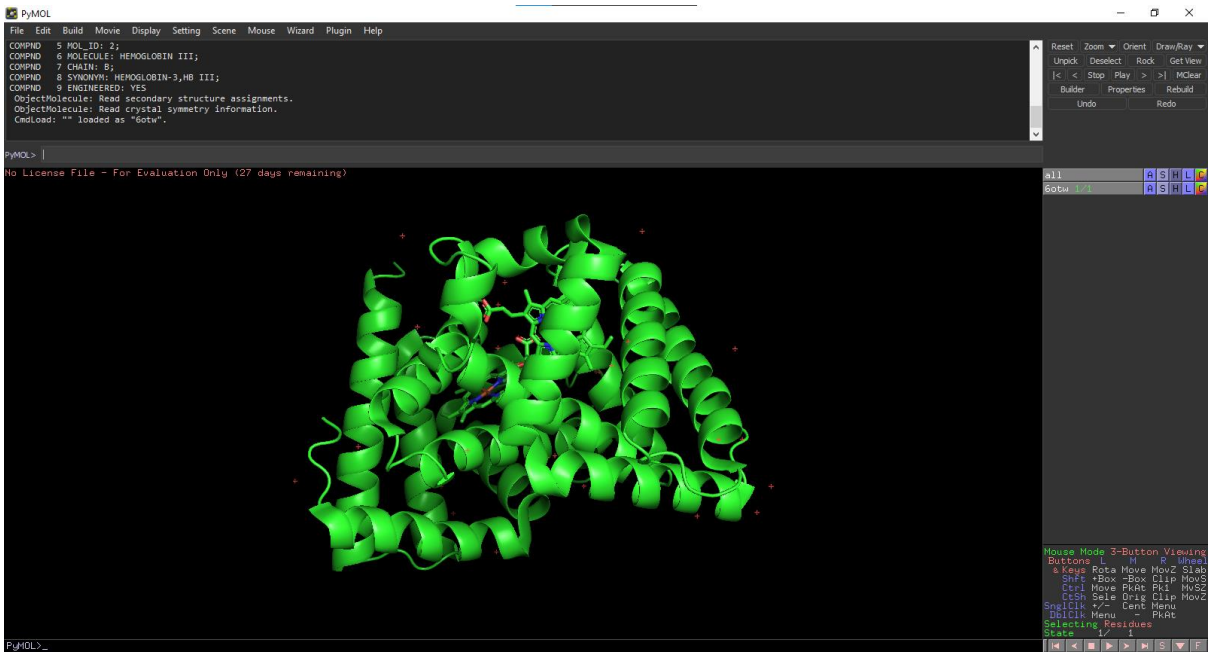
Now that protein is downloaded open PyMOL and on top left corner click File > open

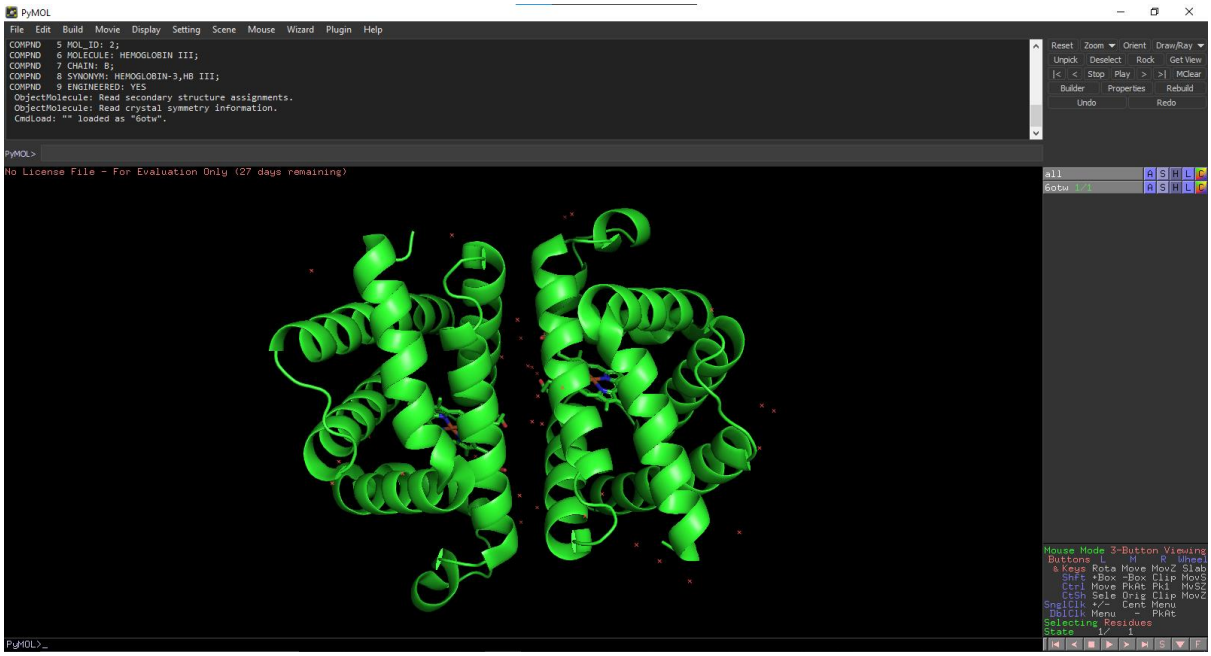


Now browse for the file you just downloaded (6otw.pdb)



This is how the screen of imported file looks like…




PyMOL
File Edit Build Movie Display Setting Scene Mouse Wizard Plugin Help
COMPND 5 MOL_ID: 2;
COMPND 6 MOLECULE: HEMOGLOBIN III;
COMPND 7 CHAIN: B;
COMPND 8 SYNONYM: HEMOGLOBIN-3,HB III;
COMPND 9 ENGINEERED: YES
ObjectMolecule: Read secondary structure assignments.
ObjectMolecule: Read crystal symmetry information.
CmdLoad: "" loaded as "6otw".
Reset Zoom Orient Draw/Ray
Unpick Deselect Rock Get View
Stop Play MClear
Builder Properties Rebuild
Undo Redo
PyMOL>
No License File - For Evaluation Only (27 days remaining)
all
6otw 1/1
Mouse Mode 3-Button Viewing
Buttons L M R Wheel
& Keys Rota Move MovZ Slab
Shft +Box -Box Clip MovS
Ctrl Move PkAt Pk1 MvSZ
CtSh Sele Orig Clip MovZ
SnglClk +/- Cent Menu
DblClk Menu - PkAt
Selecting Residues
State 1/ 1
PyMOL>_