

Computer Lab 1d – Protein Data Bank and PDB files:

Exercise 1: Explore the PDB website

Answer the questions **highlighted in yellow** and send the answers to **achin14@jhu.edu** either in the body of an email or as an attached file with your JHEDID in the name (e.g. *JHEDID_lab1b.txt* if made with vim or *JHEDID_lab1b.docx* if made with MSWord).

Open a Browser Window and go to <http://www.wwpdb.org>.

The Protein Data Bank, known simply as the PDB, is the central depository for files that contain the coordinates of known protein structures. Biomolecular structural models are deposited to this site and the entire archive of deposited structures can be downloaded from here. You should be aware of this site for its general information.

There are four members of the wwPDB: Protein Data Bank Japan (PDBj), Protein Data Bank in Europe (PDBe), the Research Collaboratory for Structural Bioinformatics Protein Data Bank (RCSB) and the Biological Magnetic Resonance Data Bank (BMRB). Each member has its own web site.

The RCSB is located at Rutgers University. Although used worldwide it is primarily a USA project. The PDBj and PDBe use the same structural information from the PDB and integrate it into relational databases containing extensive additional information on each protein in the structural database. These three sites are basically competing sites that use different approaches to the same information.

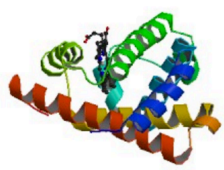
Open the RCSB site at **www.rcsb.org**. First, look for the number of experimentally-determined protein structures in the PDB. Click on **Search** tab at the top, then click on **PDB statistics** tab, and scroll down to and click the [Growth of Released Structures Per Year](#) link to have a look at the PDB content growth graph.

Although the rate of growth is leveling off there are still over 120,000 structures deposited. In order to use this information, we need to understand the tools available for accessing the databases with protein structural information. And as you will see today, there are numerous protein databases on the web.

1. What experimental method accounts for the majority of solved molecular structures? NMR, X-ray diffraction, or electron microscopy? (Hint: Go back one page and read the [Summary Table of Released Entries](#), a link near the top of the page page.)

Let's get started by picking an interesting protein to explore.

Type **Myoglobin** in the search window at the top and click **Search**. Scroll down to where the individual structure list begins. One of the top items should look something like the following (it can change daily):



5B84

X-ray crystal structure of met I107Y sperm whale myoglobin


[Liao, F.](#), [Yuan, H.](#), [Du, K.J.](#), [You, Y.](#), [Gao, S.Q.](#), [Wen, G.B.](#), [Lin, Y.W.](#), [Tan, X.](#)

(2016) Mol Biosyst

Released: 8/24/2016
Method: X-ray Diffraction
Residue Count: 153

Macromolecule:
 Myoglobin (protein)
Unique Ligands: [HEM](#)

[Download File](#) [View File](#) ☒



In the example above, **5B84**, is the **PDB ID code** or just **pdrcode**. You will frequently access the protein databank using a pdrcode.

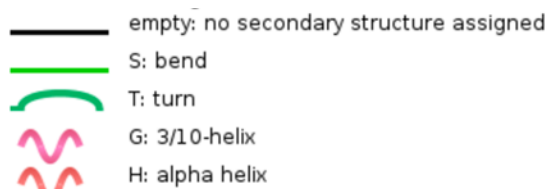
Notice that there are many such structures of myoglobin; some crystallized under different conditions, some with mutations, some with different ligands on the heme, etc. We will look at a specific structure. Enter the pdrcode, **104m**, in the search window at the top and click **Search**. This should bring you to a page containing information for the following entry: 104M, SPERM WHALE MYOGLOBIN N-BUTYL ISOCYANIDE AT PH 7.0.

Experimental details are listed on this page in addition to links to further information at the PDB and links to many other databases that contain information on this protein.

Click on the **Sequence** tab near the top of the page. You should see the following amino acid sequence information together with secondary structure information (the colors may not be the same as shown here). The amino acid sequence of a protein is called the **primary structure**.

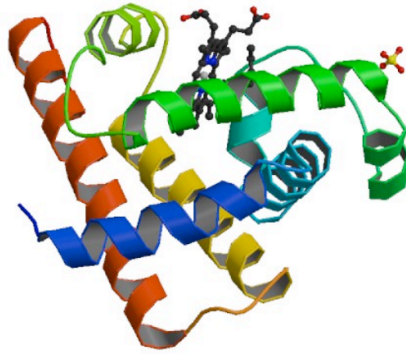


The following icons represent **secondary structures** along the sequence of a protein.



2. What secondary structure type is the most abundantly found in myoglobin?

Now let's look at an image generated from the coordinates specified in the PDB file for 104M. Go back to the front page for entry 104M (**Structure Summary** tab). At the left side you should see a figure similar to the one shown below. Notice that α -helix secondary structure is prevalent in myoglobin in agreement with the icons shown over the sequence above.



PDB ID 104M Ribbon Diagram from PDB

Exercise 2: Explore a PDB file itself

The PDB file contains the data for a protein model in the protein data bank. You should be familiar with its contents and format because you will be creating and analyzing PDB files for the remainder of this course.

Click on the **Display Files** link on the upper right hand corner of the myoglobin **Summary** page. Select the **PDB Format**

Look at the first column. It starts with HEADER, TITLE, COMPND...etc. These first column entries are useful when parsing a PDB file.

The top portion of a PDB file contains critical information about the protein.

3. What organism is the protein from?**4. Who determined the structure?** (Hint: Authors)

From the SOURCE and KEYWDS lines you can read that Myoglobin is the oxygen transport macromolecule in skeletal muscle.

Scroll down the file to the REMARK 2 RESOLUTION line.

5. What is the resolution for this X-ray determined structure?**6. Why is this number important?**

Scroll down the file until you get to the ATOM information (Where the word, ATOM, is the first word in the line). The ATOM line contains the information we will be most concerned with in this course because the three-dimensional coordinates of every atom in the structure are stored here.

The format of a PDB file is very specific – certain data types are in certain columns. Here is a list of what is possible in each character column (1-80) in a typical PDB file.

COLUMNS	DATA TYPE	FIELD	DEFINITION
1 - 6	Record name	"ATOM "	
7 - 11	Integer	serial	Atom serial number.
13 - 16	Atom	name	Atom name.
17	Character	altLoc	Alternate location indicator.
18 - 20	Residue name	resName	Residue name.
22	Character	chainID	Chain identifier.
23 - 26	Integer	resSeq	Residue sequence number.
27	AChar	iCode	Code for insertion of residues.
31 - 38	Real(8.3)	x	Orthogonal coordinates for X in Angstroms.
39 - 46	Real(8.3)	y	Orthogonal coordinates for Y in Angstroms.
47 - 54	Real(8.3)	z	Orthogonal coordinates for Z in Angstroms.
55 - 60	Real(6.2)	occupancy	Occupancy.
61 - 66	Real(6.2)	tempFactor	Temperature factor.
77 - 78	LString(2)	element	Element symbol, right-justified.
79 - 80	LString(2)	charge	Charge on the atom.

Below is an example of the ATOM information with column labels and column numbers. The column numbers and labels are not in PDB files you have to recognize the fields.

	ATOM#	Name	A.A.	Chain	Residue#	X	Y	Z	Occupancy	B-factor
0000000001111111112222222223333333334444444445555555556666666667										
1234567890123456789012345678901234567890123456789012345678901234567890										
ATOM	1	N	VAL	A	1	-3.813	14.951	14.094	1.00	21.25
ATOM	2	CA	VAL	A	1	-3.172	15.504	15.319	1.00	20.50
ATOM	3	C	VAL	A	1	-2.378	14.420	16.035	1.00	19.55
ATOM	4	O	VAL	A	1	-2.827	13.270	16.148	1.00	20.11
ATOM	5	CB	VAL	A	1	-4.222	16.067	16.308	1.00	21.32
ATOM	6	CG1	VAL	A	1	-5.022	17.183	15.656	1.00	21.96
ATOM	7	CG2	VAL	A	1	-5.164	14.968	16.741	1.00	22.04
ATOM	8	N	LEU	A	2	-1.189	14.779	16.501	1.00	17.36
ATOM	9	CA	LEU	A	2	-0.366	13.827	17.223	1.00	15.95
ATOM	10	C	LEU	A	2	-0.855	13.707	18.651	1.00	15.27
ATOM	11	O	LEU	A	2	-1.277	14.693	19.265	1.00	15.85
ATOM	12	CB	LEU	A	2	1.101	14.259	17.248	1.00	14.61
ATOM	13	CG	LEU	A	2	1.980	14.080	16.004	1.00	14.41
ATOM	14	CD1	LEU	A	2	1.591	15.071	14.904	1.00	13.31
ATOM	15	CD2	LEU	A	2	3.436	14.301	16.418	1.00	13.39

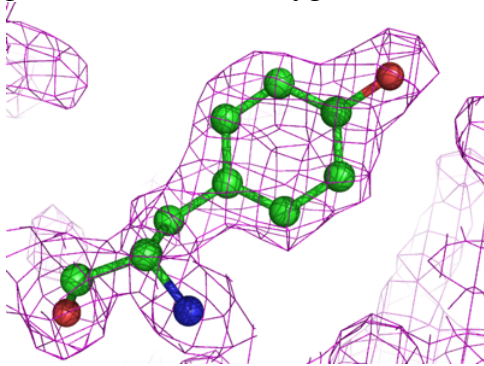
The atoms are listed sequentially from the N to the C terminus of the protein. The order of atoms is: Backbone (N, CA, C, O) first, then the sidechain atoms. (You may see variations on this, but they are supposed to be in this order.)

Note that the chain ID (A in this case) is not always present. It is only necessary if there are multiple separate polypeptide chains in the structure - but sometimes it is included in a single chain protein like myoglobin. Other optional fields are missing from this example (altLoc, iCode, element, charge). But the above is what you need to recognize for this course.

The 3D coordinates for each atom are given as a Cartesian triplet (x, y, z).

7. What units are the coordinates given in? What is the conversion between this number and nanometers?

It is important to note that the coordinates given are an *interpretation* of the center of the atomic electron cloud density profile calculated from X-ray diffraction measurements. The x, y, z coordinates represent the best available determination of the *likely* position of each atom within the electron cloud density. Below is a figure depicting the atoms (connected by stick bonds) that have been placed inside the electron density map (magenta) of a tyrosine residue in a protein. The colors of the spheres representing atom positions are: red = oxygen, blue = nitrogen, green = carbon.



The **Occupancy** column in a PDB file gives an estimate of the fraction of time a particular atom is actually present at the position defined by the coordinates. If this number is less than 1.00 it means that the atom spends some time in another position implying that there is **static flexibility** in this residue.

8. In the myoglobin PDB file (still displayed in your browser) which atoms are NOT in specific positions all the time? (Hint: The atoms are in a residue beyond 60 but before 70.).

The **B-factor** value in a PDB file is a measure of how “fuzzy” or “smeared” the electron density is for a particular atom. A “smeared” electron density would result in a larger uncertainty in the estimated position of the atom’s center.

If one assumes that the smearing is due only to dynamic flexibility (and not static flexibility or experimental error) then **B** is defined as.

$$B = 8\pi^2 \langle r^2 \rangle$$

Where r^2 is the mean square displacement of the atom (in Å). From this expression it is possible to estimate the root mean square displacement that an atom experiences in the crystal. Presumably the atomic fluctuations would increase at higher temperatures and because of this the B factor is sometimes called the **temperature factor**.

9. What regions of a protein (surface or core) might you expect to have a more smeared electron density map based on what you know about the fluctuations that proteins undergo?

Compare the B-factors for atoms in the side chain of residue 64 against atoms in the peptide backbone. The B factor values for the backbone are less than those for the side chain atoms. 10. Why?

(Hint for exams: Make sure you understand and can discuss the following relationships: Occupancy column -> static flexibility, B factor column -> dynamic flexibility. Now is the time to ask if this is not crystal clear.)

The wwPDB is transitioning from the above file format to a new format called PDBx/mmCIF. In this new format the data column widths may be variable and also be in variable order. See the Appendix for an example.

Now let's check out the European site to see a different flavor of protein data access.

Go to www.ebi.ac.uk/pdbe/ using a browser. This page displays the many resources and services provided by the European project. We will briefly browse the structural information available for myoglobin on the PDBe.

Type **myoglobin** into the search box at the top. Then click **Search**. You should see a list of the myoglobin structural models available from this site. If it is not the top entry enter **104m** in the search box and click **Search**. The resulting page gives much of the same information you viewed earlier at the PDB site. The PDB file can be downloaded from this page and links to other information about myoglobin are also available from this page. The menu on the left provides myoglobin information in many other databases that are covered in a Bioinformatics course.

The purpose of this short introduction to the PDBe site is to make you aware of it. When downloading coordinate files for protein structural models it doesn't matter whether you use the PDB or the PDBe. *They both use the same primary data* but have different value-added features. I encourage you to explore the PDB, PDBe and PDBj sites on your own and decide your preference.

Exercise 2: Python program to create PDB files.

We will inspect a python module called ***lattice.py*** as an example of how to create and write a file which describes the initial 3D configuration of particles in a PDB file. A Python module similar to this one will be used in the coming weeks to create the initial configuration for our molecular simulations. We will go over the statements in ***lattice.py*** and inspect the output from the program in class.

Open a terminal window on the Mac, change to your flashdrive home directory, create a subdirectory for today's lab and fetch a file called *lattice.py* from the cluster.

```
cd /Volumes/[Your_flashdrive]
mkdir lab1d
cd lab1d
sftp compbio2@kirin.kit.jhu.edu
cd Shared
get lattice.py
bye
```

Use the UNIX command **less** (or **view** which is a read-only version of **vim**) to view the file as you go through the various parts of the program.

There are two ways to include comments in Python: Between triple quotes at the top of the text and after the **#** symbol anywhere in the text. Scroll down through the comments to the first actual code,

```
def param():
```

This line indicates that we are going to define a function or subroutine called, **param**. This first subroutine creates a **dictionary**. Let's explore exactly what this means. In a separate terminal window start Python interactively (Just type, **python3**). Then at the **>>>** prompt, type the following commands found in the *lattice.py* program (no indentation),

```
Data = {}
Data['NPART'] = 144
Data['SIGMA'] = 3.4
```

Here you are defining **values** (e.g. 144) associated with **keys** (NPART or number of particles) in the dictionary (**Data**). Now call the dictionary by typing,

```
Data
```

and then type,

```
Data['SIGMA']
```

In this case SIGMA is the **key** to retrieve the **value** (3.4) from the **dictionary**, Data. Here is how you would use the information in the dictionary to define a new variable for the van der Waals radius. Type the following:

```
VDWrad = float(Data['SIGMA']) * 1.12/2.0
print(VDWrad)
```

This relationship of sigma to VDW radius will become clear later in the course.

The second subroutine in *lattice.py* (itself called **lattice**) creates a **list** with the particle system coordinates in it. We use a **list method** called **append** to increment the list. For example type the following after your Python **>>>** prompt in the interactive session,

```

X=[]
X.append(1.0)
X.append(2.0)
X.append(3.0)

```

Now type,

```
print(X)
```

11. What did you get? This is the contents of list X printed as a list.

One of the procedures you will do over and over is to output the items in a list separately and in succession. Type the following to see what I mean (note the ... indentation),

```

for i in range(len(X)):
...     print(X[i])

```

...[press return key]

Note: Don't forget to indent here.

The third subroutine in *lattice.py*, **pdbout**, actually writes the coordinates in PDB file format. Find the line that defines the format,

```
format = '%6s%5d  %-3s%4s  %4d      %8.3f%8.3f%8.3f%6.2f%6.2f\n'
```

Since this syntax is not obvious we will tell you here that it means:

- 6 string characters,
- 5 integer digits,
- 2 spaces,
- 3 string characters (the minus sign means left justified),
- 4 string characters,
- 2 spaces,
- 4 integer digits,
- 4 spaces,
- 3 columns of floating point numbers each 8 spaces with precision of 3 ,
- 2 columns of floating point numbers each 6 spaces with precision of 2,
- carriage return (the \n).

Formatting is one of the necessary evils in computational work. The PDB format is a legacy from the days when most scientific programs were written in FORTRAN where formatting rules.

Look a few lines above the format line in *lattice.py* to see where the variables: *record*, *atmnam* and *resnam* are defined and make a mental note of these string values. The atom name is “O” for oxygen and the residue name is “UNK” for unknown.

The *lattice.py* program writes output to a file rather than displaying it to the screen. **12. What is the name of the output file?** (Hint: Find the comment line in

the `run()` subroutine that says “Open a file...”). Remember this file name.

Notice that the `run` subroutine is actually the main program defined near the bottom of the file. The last line in the file calls the subroutine `run` and, in turn, it simply calls the three other functions (`param`, `lattice`, `pdbout`) in turn.

3. Quit the interactive python session (**ctrl-d**) and run the program with the following command,

```
python3 lattice.py
```

Use the **less** command to inspect the output file that should have been created.

13. What is the residue name used for all residues in this file?

14. What is the atom type used in this file? (Hint: What is the value of the variable, `atmnam` in the code?)

Note that each residue only has one atom in this file (the atom numbers and residue numbers are the same) unlike a protein molecule that has many atoms.

4. Look at a molecular graphic image of your lattice,

```
pymol init.pdb  
S | Spheres
```

You may want to turn off clipping,

```
Display | Clip | None
```

You should be able to use the mouse to rotate the graphic image of the lattice. The spheres are red because the program thinks the atoms are oxygen (see above). Note that the spheres are NOT touching.

Exit **PyMOL** and open the *init.pdb* file using **vim**. Enter the following command (note the white spaces),

```
:1,$ s/ O / C /g
```

This means: from line number 1 to the end of file, substitute C for O globally (in the whole line). [Note the spaces before and after the characters to be changed.] Now write and quit the **vim** session and open the *init.pdb* file again with **PyMOL** and show as spheres.

15. What color are the spheres? (Hint: In PyMOL go to the color menu for the init object: C | by element and view the choices. The default colors used are listed in the the 2nd CHNOS... down.)

Note that now the spheres are touching because the spheres are larger.

16. Why are they of larger radius?

The take home message here is that any 3D coordinate information in PDB format can be viewed using a molecular graphics program such as **PyMOL**. The size of the spheres rendered is dependent on what atom type is designated. We will find this useful to view the results of our first molecular simulations that are of particles in a box. Later we will use **PyMOL** and other applications (**VMD**) to view the protein models we work with.

Appendix.

The wwPDB is transitioning from the PDB file format described above. In the new format the section that contains coordinates may have a variable number of columns. A description of what is in each column is contained in a header section with a list of “_atom.site” variables. See the example below. There may be many more columns than are shown in this example.

```

_atom_site.group_PDB
_atom_site.id
_atom_site.type_symbol
_atom_site.label_atom_id
_atom_site.label_alt_id
_atom_site.label_comp_id
_atom_site.label_asym_id
_atom_site.label_entity_id
_atom_site.label_seq_id
_atom_site.pdbx_PDB_ins_code
_atom_site.Cartn_x
_atom_site.Cartn_y
_atom_site.Cartn_z
_atom_site.occupancy
_atom_site.B_iso_or_equiv
_atom_site.pdbx_formal_charge
_atom_site.auth_seq_id
_atom_site.auth_comp_id
_atom_site.auth_asym_id
_atom_site.auth_atom_id
_atom_site.pdbx_PDB_model_num

```

ATOM	1	N	N	.	MET	A	1	1	?	27.340	24.430	2.614	1.00	9.67	?	1	MET	A	N	1
ATOM	2	C	CA	.	MET	A	1	1	?	26.266	25.413	2.842	1.00	10.38	?	1	MET	A	CA	1
ATOM	3	C	C	.	MET	A	1	1	?	26.913	26.639	3.531	1.00	9.62	?	1	MET	A	C	1
ATOM	4	O	O	.	MET	A	1	1	?	27.886	26.463	4.263	1.00	9.62	?	1	MET	A	O	1
ATOM	5	C	CB	.	MET	A	1	1	?	25.112	24.880	3.649	1.00	13.77	?	1	MET	A	CB	1
ATOM	6	C	CG	.	MET	A	1	1	?	25.353	24.860	5.134	1.00	16.29	?	1	MET	A	CG	1
ATOM	7	S	SD	.	MET	A	1	1	?	23.930	23.959	5.904	1.00	17.17	?	1	MET	A	SD	1
ATOM	8	C	CE	.	MET	A	1	1	?	24.447	23.984	7.620	1.00	16.11	?	1	MET	A	CE	1
ATOM	9	N	N	.	GLN	A	1	2	?	26.335	27.770	3.258	1.00	9.27	?	2	GLN	A	N	1
ATOM	10	C	CA	.	GLN	A	1	2	?	26.850	29.021	3.898	1.00	9.07	?	2	GLN	A	CA	1