Class 10 Structural Bioinformatics

Job Rocha PID 59023124

1: Introduction to the RCSB Protein Data Bank (PDB)

The PDB archive is the major repository of information about the 3D structures of large biological molecules, including proteins and nucleic acids. Understanding the shape of these molecules helps to understand how they work. This knowledge can be used to help deduce a structure's role in human health and disease, and in drug development. The structures in the PDB range from tiny proteins and bits of DNA or RNA to complex molecular machines like the ribosome composed of many chains of protein and RNA.

In the first section of this lab we will interact with the main US based PDB website (note there are also sites in Europe and Japan).

Visit: http://www.rcsb.org/ and answer the following questions

NOTE: The "Analyze" > "PDB Statistics" > "by Experimental Method and Molecular Type" on the PDB home page should allow you to determine most of these answers.

PDB statistics

Open RStudio and begin a new class09 project. If we have covered GitHub in a previous class then you should create this within your GitHub tacked directory/folder from that class. Make sure "Create a git repository" option is NOT ticked. This is because we want to use the same git repository as we used last day and not start a new one - if you are not sure what this means ask Barry now!

Next, open a new Quarto document (File > New File > Quarto Document…). As always, we will aim to have a rendered PDF report with working code by the end of this class!

Download a CSV file from the PDB site (accessible from "Analyze" > "PDB Statistics" > "by Experimental Method and Molecular Type". Move this CSV file into your RStudio project and use it to answer the following questions:

```
data.from.string.to.numeric <- function(file){
  raw.string <- read.csv("Data Export Summary.csv", row.names = 1)
  data <- as.data.frame(lapply(raw.string, function(x){ as.integer(gsub(",", "", x)) }))
  rownames(data) <- rownames(raw.string)
  return(data)
}

csv <- data.from.string.to.numeric("Data Export Summary.csv")
csv
```

```
                         X.ray    EM   NMR Multiple.methods Neutron Other
Protein (only)          158844 11759 12296               197      73    32
Protein/Oligosaccharide   9260  2054    34                 8       1     0
Protein/NA                8307  3667   284                 7       0     0
Nucleic acid (only)       2730   113  1467                13       3     1
Other                      164     9    32                 0       0     0
Oligosaccharide (only)      11     0     6                 1       0     4
                         Total
Protein (only)          183201
Protein/Oligosaccharide  11357
Protein/NA               12265
Nucleic acid (only)       4327
Other                      205
Oligosaccharide (only)      22
```

Q1: What percentage of structures in the PDB are solved by X-Ray and Electron Microscopy.

```
sums <- apply(csv, 2, sum)
print(paste0("Percentage of structures solved by X-ray: ",sums["X.ray"] * 100 / sums["Tota
```

```
[1] "Percentage of structures solved by X-ray: 84.8323138278999%"
```

```
print(paste0("Percentage of structures solved by EM: ",sums["EM"] * 100 / sums["Total"], "
```

```
[1] "Percentage of structures solved by EM: 8.32730145663909%"
```

Q2: What proportion of structures in the PDB are protein?

```
print(paste0("Proportion of structures that are proteins: ", csv["Protein (only)", "Total"
```

```
[1] "Proportion of structures that are proteins: 86.6702621382648%"
```

Q3: Type HIV in the PDB website search box on the home page and determine how many HIV-1 protease structures are in the current PDB? *14,220 Structures*

The PDB format

Now download the "PDB File" for the HIV-1 protease structure with the PDB identifier 1HSG. On the website you can "Display" the contents of this "PDB format" file.

Alternatively, you can examine the contents of your downloaded file in a suitable text editor or use the Terminal tab from within RStudio (or your favorite Terminal/Shell) and try the following command:

```
#less ~/Downloads/1hsg.pdb         ## (use 'q' to quit)
```

NOTE: When viewing the file stop when you come the lines beginning with the word "ATOM". We will discuss this ubiquitous PDB file format when you have got this far.

Protein Data Bank files (or PDB files) are the most common format for the distribution and storage of high-resolution biomolecular coordinate data. At their most basic, PDB coordinate files contain a list of all the atoms of one or more molecular structures. Each atom position is defined by its x, y, z coordinates in a conventional orthogonal coordinate system. Additional data, including listings of observed secondary structure elements, are also commonly (but not always) detailed in PDB files.

Molecular graphics programs such as Mol*, VMD, PyMol and Chimera take these files and plot them in 3D with the ability to make simplified and stylized representations such as the one shown below:

Figure 1. HIV-1 protease structure (PDB code: 1HSG) in complex with the small molecule indinavir.

2. Visualizing the HIV-1 protease structure.

The HIV-1 protease is an enzyme that is vital for the replication of HIV. It cleaves newly formed polypeptide chains at appropriate locations so that they form functional proteins. Hence, drugs that target this protein could be vital for suppressing viral replication. A handful of drugs - called HIV-1 protease inhibitors (saquinavir, ritonavir, indinavir, nelfinavir, etc.) - are currently commercially available that inhibit the function of this protein, by binding in the catalytic site that typically binds the polypeptide.

In this section we will use the 2Å resolution X-ray crystal structure of HIV-1 protease with a bound drug molecule indinavir (PDB ID: 1HSG). We will use the Mol* molecular viewer to visually inspect the protein, the binding site and the drug molecule. After exploring features of the complex we will move on to perform bioinformatics analysis of single and multiple crystallographic stuctures to explore the conformational dynamics and flexibility of the protein - important for it's function and for considering during drug design.

Using Mol *Mol* (pronounced "molstar") is a new web-based molecular viewer that is rapidly gaining in popularity and utility. At the time of writing it is still a long way from having the full feature set of stand-alone molecular viewer programs like VMD, PyMol or Chimera. However, it is gaining new features all the time and does not require any download or complicated installation.

You can use Mol* directly at the PDB website (as well as UniProt and elsewhere). However, for the latest and greatest version we will visit the Mol* homepage at: https://molstar.org/viewer/.

To load a structure from the PDB we can enter the PDB code and click "Apply" in the "Download Structure" menu (see figure below)

Once loaded the sidebar should change to the so-called hierarchical "State Tree" menu. Of particular note there are entries for Polymer, Ligand and Water. You can turn the display of any of these entries OFF/ON by clicking on the eye icon or delete them by clicking the "trash" bin icon (but we will not do that just yet). We can turn this left-side control panel off to save screen space. Especially as we will not need it again until we come to close the molecule or read a new molecule later.

Key-point: You can access and change all visual representations on the opposite right side control panel under the "Components" drop-down menu (see figure below). Try togling ON/OFF the display of Ligand and Water with the "eye" icon.

Getting to know HIV-Pr.

Let's temporally toggle OFF/ON the display of water molecules and change the display representation of the Ligand to Spacefill (a.k.a VdW spheres). To do this:

Click on three dots for the "Ligand" components entry in the right side control panel (blue box in above image), Then from the drop-down select Add Representation > Spacefill. Note that there are now two "reps" listed for the ligand component that you can control independently in the expandable menu accessible from clicking the three dots (see red box below).

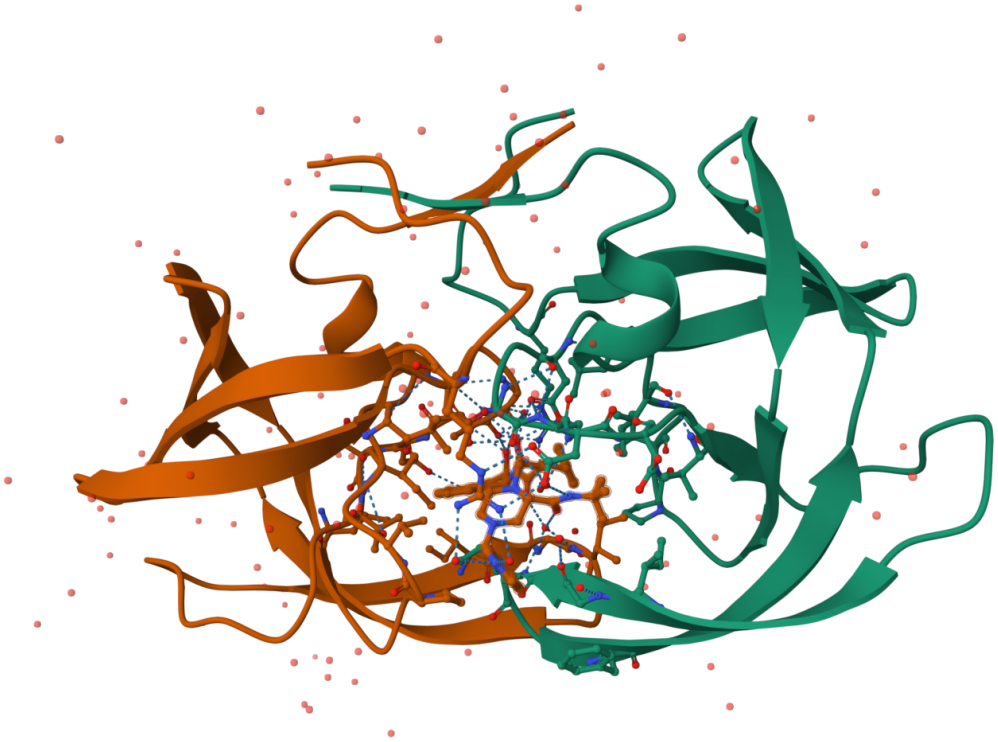
Let's also change the protein "Polymer" > "Set Coloring" > "Residue Property" > "Secondary Structure".

Key-point: All these expanding drop-down menus can quickly become overwhelming. I find that closing them by clicking the 3 dots again can help keep things tidy and avoid menu items disappearing off small screens.

Saving an image.

Once you are happy with your display you can save a high-resolution image to your computer for including in your Quarto document. To do this find the "iris-like" screenshot icon on the right side of the display region and select your resolution and click download (see figure below)

Here is a rubish pic of HIV-Pr that is not very useful yet.



Delving deeper.

To help highlight important amino acid residues that interact with the ligand you can click on the ligand itself. This will lead to a new "Focus Surroundings (5A)" display component to appear. Mousing over this will highlight the corresponding amino acids in the Sequence display panel.

Note: Zoom in and rotate to examine these ligand interactions. Of these positions Asp 25 (D25) in both chains is critical for protease activity. Can you find this amino acid in both chains? Note the residue information displayed in the bottom right of the viewing window as you mouse over different amino acids.

Cleaning up the display.

Most viewers will find that displaying all ligand surrounding amino acids is too busy for a single display. Turn off the display of these positions by clicking the eye" icon for the "Focus

Surroundings (5A)" Components entry in the right side control panel.

Now we can highlight a subset of the most important positions:

Using the top Sequence display select position Asp 25 (D25) in one of the chains. Now activate so-called "Selection Mode" by clicking the Arrow icon (red box to the right side of the 3D viewer panel in the figure below).

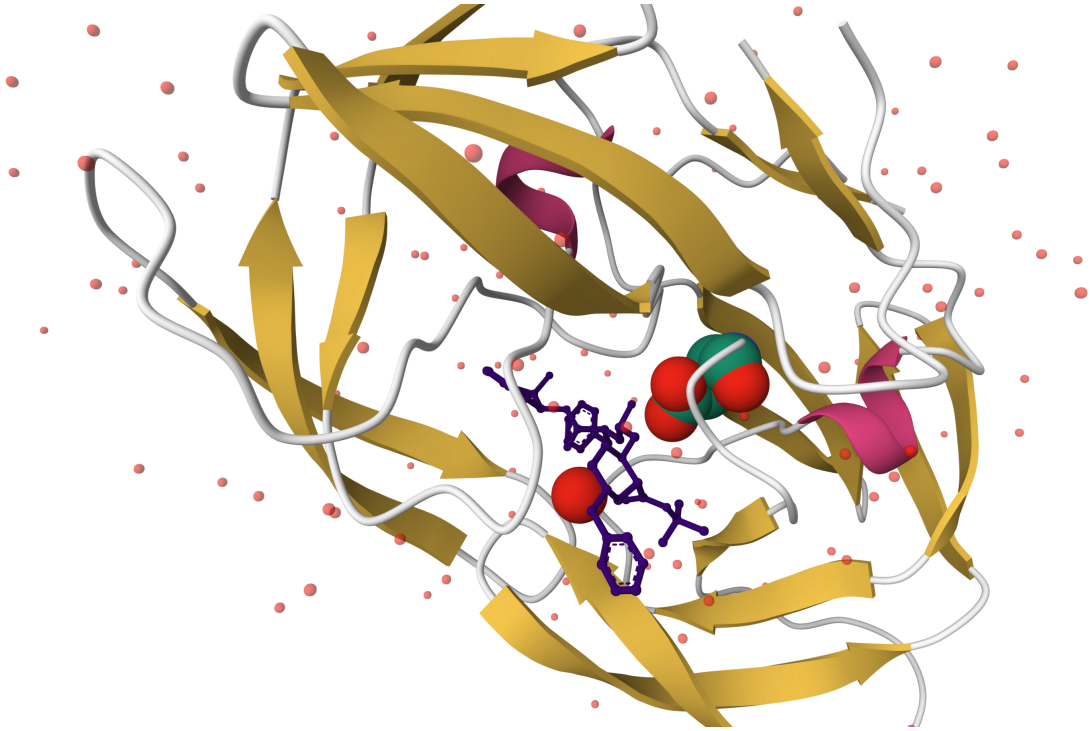
Then select the two Asp 25 positions in the 3D structure.

Finally click the cube icon (blue box in below figure) and from the drop-down menu that appears select Representation Spacefill or Ball & Stick (whatever you prefer), then click +Create Component.

Note that a new "Custom Selection" component has appeared in the right side control panel. This will contain your two D25 positions. You can again delete the "Focus Surroundings (5A)" and Focus Target Components to clean up the display.

At this point you should consider saving an image as discussed above.

And a nicer pic colored by secondary structure with catalytic active site ASP 25 shown in each chain along with MK1 drug and all important water molecules…



The important role of water

Toggle on the display of all water molecules again.

Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure? ***The resoulution of the structure (2A) is not high enough to identify the small atoms of Hydrogen, but can resolve Oxygen atoms.***

Q5: There is a critical "conserved" water molecule in the binding site. Can you identify this water molecule? What residue number does this water molecule have. ***Residue number 308.***

Now you should be able to produce an image similar or even superior to Figure 2 and save it to an image file.

Q6: Generate and save a figure clearly showing the two distinct chains of HIV-protease along with the ligand. You might also consider showing the catalytic residues ASP 25 in each chain and the critical water (we recommend "Ball & Stick" for these side-chains). Add this figure to your Quarto document.

Discussion Topic: Can you think of a way in which indinavir, or even larger ligands and substrates, could enter the binding site?

Q7: [Optional] As you have hopefully observed HIV protease is a homodimer (i.e. it is composed of two identical chains). With the aid of the graphic display can you identify secondary structure elements that are likely to only form in the dimer rather than the monomer?

3. Introduction to Bio3D in R.

Bio3D is an R package for structural bioinformatics. Features include the ability to read, write and analyze biomolecular structure, sequence and dynamic trajectory data.

In your existing Rmarkdown document load the Bio3D package by typing in a new code chunk:

```
library(bio3d)
```

Side-Note: If you see an error message reported then you will first need to install the package with the command: install.packages("bio3d") in your R Console (i.e. don't put this in your

Rmarkdown document or it will be re-installed every time you knit/render your document). This is only required once whereas the library(bio3d) command is required at the start of every new R session where you want to use Bio3D.

Reading PDB file data into R.

To read a single PDB file with Bio3D we can use the read.pdb() function. The minimal input required for this function is a specification of the file to be read. This can be either the file name of a local file on disc, or the RCSB PDB identifier of a file to read directly from the on-line PDB repository. For example to read and inspect the on-line file with PDB ID 1HSG:

```
pdb <- read.pdb("1hsg")
```

```
  Note: Accessing on-line PDB file
```

To get a quick summary of the contents of the pdb object you just created you can issue the command print(pdb) or simply type pdb (which is equivalent in this case):

```
pdb
```

```
 Call:  read.pdb(file = "1hsg")

   Total Models#: 1
     Total Atoms#: 1686,  XYZs#: 5058  Chains#: 2  (values: A B)

     Protein Atoms#: 1514  (residues/Calpha atoms#: 198)
     Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)

     Non-protein/nucleic Atoms#: 172  (residues: 128)
     Non-protein/nucleic resid values: [ HOH (127), MK1 (1) ]

   Protein sequence:
      PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYD
      QILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFPQITLWQRPLVTIKIGGQLKE
      ALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTP
      VNIIGRNLLTQIGCTLNF

+ attr: atom, xyz, seqres, helix, sheet,
         calpha, remark, call
```

Q7: How many amino acid residues are there in this pdb object? ***198 aa***

```
aa321(pdb$atom$resid[pdb$calpha])
```

```
  [1] "P" "Q" "I" "T" "L" "W" "Q" "R" "P" "L" "V" "T" "I" "K" "I" "G" "G" "Q"
 [19] "L" "K" "E" "A" "L" "L" "D" "T" "G" "A" "D" "D" "T" "V" "L" "E" "E" "M"
 [37] "S" "L" "P" "G" "R" "W" "K" "P" "K" "M" "I" "G" "G" "I" "G" "G" "F" "I"
 [55] "K" "V" "R" "Q" "Y" "D" "Q" "I" "L" "I" "E" "I" "C" "G" "H" "K" "A" "I"
 [73] "G" "T" "V" "L" "V" "G" "P" "T" "P" "V" "N" "I" "I" "G" "R" "N" "L" "L"
 [91] "T" "Q" "I" "G" "C" "T" "L" "N" "F" "P" "Q" "I" "T" "L" "W" "Q" "R" "P"
[109] "L" "V" "T" "I" "K" "I" "G" "G" "Q" "L" "K" "E" "A" "L" "L" "D" "T" "G"
[127] "A" "D" "D" "T" "V" "L" "E" "E" "M" "S" "L" "P" "G" "R" "W" "K" "P" "K"
[145] "M" "I" "G" "G" "I" "G" "G" "F" "I" "K" "V" "R" "Q" "Y" "D" "Q" "I" "L"
[163] "I" "E" "I" "C" "G" "H" "K" "A" "I" "G" "T" "V" "L" "V" "G" "P" "T" "P"
[181] "V" "N" "I" "I" "G" "R" "N" "L" "L" "T" "Q" "I" "G" "C" "T" "L" "N" "F"
```

Q8: Name one of the two non-protein residues? ***H2O and MK1***

Q9: How many protein chains are in this structure? ***Two chains (A and B).***

Note that the attributes (+ attr:) of this object are listed on the last couple of lines. To find the attributes of any such object you can use:

```
attributes(pdb)
```

```
$names
[1] "atom"   "xyz"    "seqres" "helix"  "sheet"  "calpha" "remark" "call"

$class
[1] "pdb" "sse"
```

To access these individual attributes we use the dollar-attribute name convention that is common with R list objects. For example, to access the atom attribute or component use pdb$atom:

```
head(pdb$atom)
```

```
  type eleno elety  alt resid chain resno insert      x      y     z o     b
1 ATOM     1     N <NA>   PRO     A     1   <NA> 29.361 39.686 5.862 1 38.10
2 ATOM     2    CA <NA>   PRO     A     1   <NA> 30.307 38.663 5.319 1 40.62
3 ATOM     3     C <NA>   PRO     A     1   <NA> 29.760 38.071 4.022 1 42.64
4 ATOM     4     O <NA>   PRO     A     1   <NA> 28.600 38.302 3.676 1 43.40
5 ATOM     5    CB <NA>   PRO     A     1   <NA> 30.508 37.541 6.342 1 37.87
6 ATOM     6    CG <NA>   PRO     A     1   <NA> 29.296 37.591 7.162 1 38.40
  segid elesy charge
1  <NA>     N   <NA>
2  <NA>     C   <NA>
3  <NA>     C   <NA>
4  <NA>     O   <NA>
5  <NA>     C   <NA>
6  <NA>     C   <NA>
```

Predicting functional motions of a single structure.

Let's read a new PDB structure of Adenylate Kinase and perform Normal mode analysis.

```
adk <- read.pdb("6s36")
```

```
Note: Accessing on-line PDB file
 PDB has ALT records, taking A only, rm.alt=TRUE
```

```
adk
```

```
Call:  read.pdb(file = "6s36")

  Total Models#: 1
    Total Atoms#: 1898,  XYZs#: 5694  Chains#: 1  (values: A)

    Protein Atoms#: 1654  (residues/Calpha atoms#: 214)
    Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)

    Non-protein/nucleic Atoms#: 244  (residues: 244)
    Non-protein/nucleic resid values: [ CL (3), HOH (238), MG (2), NA (1) ]

  Protein sequence:
     MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
```

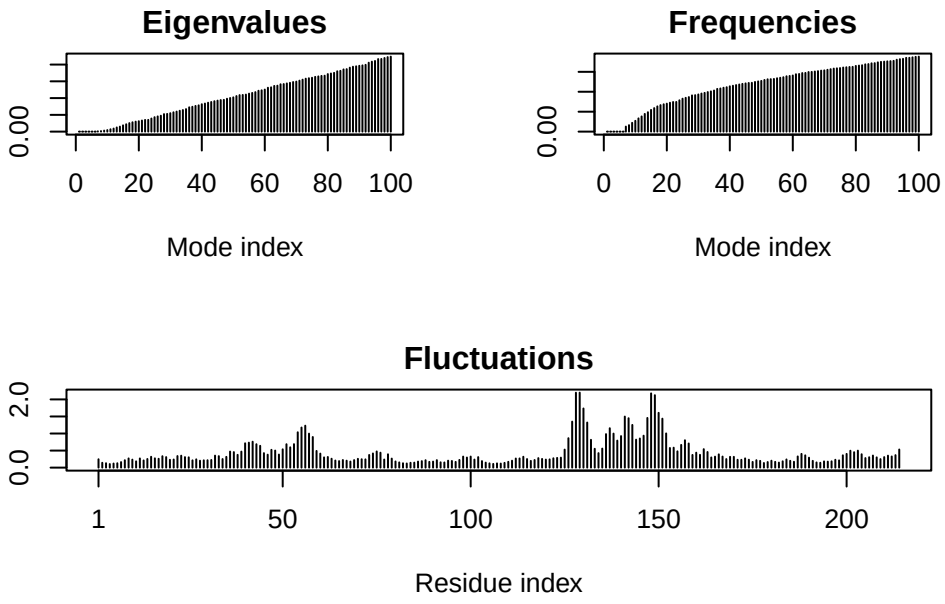
```
      DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDKI
      VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
      YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG

+ attr: atom, xyz, seqres, helix, sheet,
        calpha, remark, call
```

Normal mode analysis (NMA) is a structural bioinformatics method to predict protein flexibility and potential functional motions (a.k.a. conformational changes).

```
# Perform flexiblity prediction
m <- nma(adk)
```

```
 Building Hessian...        Done in 0.028 seconds.
 Diagonalizing Hessian...   Done in 0.312 seconds.
```

```
plot(m)
```



To view a "movie" of these predicted motions we can generate a molecular "trajectory" with the mktrj() function

```
mktrj(m, file="adk_m7.pdb")
```

Now we can load the resulting "adk_m7.pdb" PDB into Mol* with the "Open Files" option on the right side control panel. Once loaded click the "play" button to see a movie (see image below). We will discuss how this method works at the end of this lab when we apply it across a large set of homologous structures.