

Class 10: Structural Bioinformatics (pt1)

Yoonjin Lim (PID: A16850635)

Table of contents

The PDB Database	1
2. Visualizing with Mol-star	4
3. Using the bio3d package in R	8
Molecular visualization in R	9
Predicting functional motions of a single structure	10

The PDB Database

The main repository of biomolecular structure data is called the [Protein Data Bank](#) (PDB for short). It is the second oldest database (after GenBank).

What is currently in the PDB? We can access current composition stats [here](#).

```
stats <- read.csv ("Data Export Summary.csv", row.names=1)
head(stats)
```

	X.ray	EM	NMR	Multiple.methods	Neutron	Other
Protein (only)	171,959	18,083	12,622	210	84	32
Protein/Oligosaccharide	10,018	2,968	34	10	2	0
Protein/NA	8,847	5,376	286	7	0	0
Nucleic acid (only)	2,947	185	1,535	14	3	1
Other	170	10	33	0	0	0
Oligosaccharide (only)	11	0	6	1	0	4
Total						
Protein (only)	202,990					
Protein/Oligosaccharide	13,032					
Protein/NA	14,516					
Nucleic acid (only)	4,685					

Other	213
Oligosaccharide (only)	22

```
202990/252188522 * 100
```

```
[1] 0.08049137
```

Notes: the structure analysis in bioinformatics is mainly to predict the function of protein. However, we are likely to be given huge coverage of sequences and tiny coverage/proportion of structures to predict the function. This number represents the proportion of structure...

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

```
stats$X.ray
```

```
[1] "171,959" "10,018" "8,847" "2,947" "170" "11"
```

```
as.numeric(stats$X.ray)
```

Warning: NAs introduced by coercion

```
[1] NA NA NA NA 170 11
```

```
x <- stats$X.ray
# Substitute comma for nothing
y <- gsub(",", "", x)
# Convert to numeric
sum(as.numeric(y))
```

```
[1] 193952
```

```
sum(stats$Neutron)
```

```
[1] 89
```

Turn this snippet into a function so I can use it any time I have this comma problem (i.e. the other columns of this `stats` table).

```
comma.sum <- function(x) {
  # Substitute comma for nothing
  y <- gsub(",", "", x)
  # Convert to numeric and sum
  return (sum(as.numeric(y)))
}
```

```
xray.sum <- comma.sum(stats$X.ray)
em.sum <- comma.sum(stats$EM)
total.sum <- comma.sum(stats$Total)
```

```
xray.sum/total.sum *100
```

```
[1] 82.37223
```

```
em.sum/total.sum *100
```

```
[1] 11.30648
```

X-Ray: 82.37223% Electron Microscopy: 11.30648%

Notes: comma.sum is the function I made for base R to remove the comma from the numeric values and to recognize the X-Ray or Total which is set as the character into numeric value (character => numeric).

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

```
total.protein <- comma.sum (stats["Protein (only)", "Total"])
(total.protein/total.sum) *100
```

```
[1] 86.2107
```

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?

SKIPPED.

2. Visualizing with Mol-star

Explore the HIV-1 protease structure with PDB cod: 1HSG. Mol-star homepage at: <https://molstar.org/viewer/>.



Figure 1: Figure 1. A first view of HIV-Pr

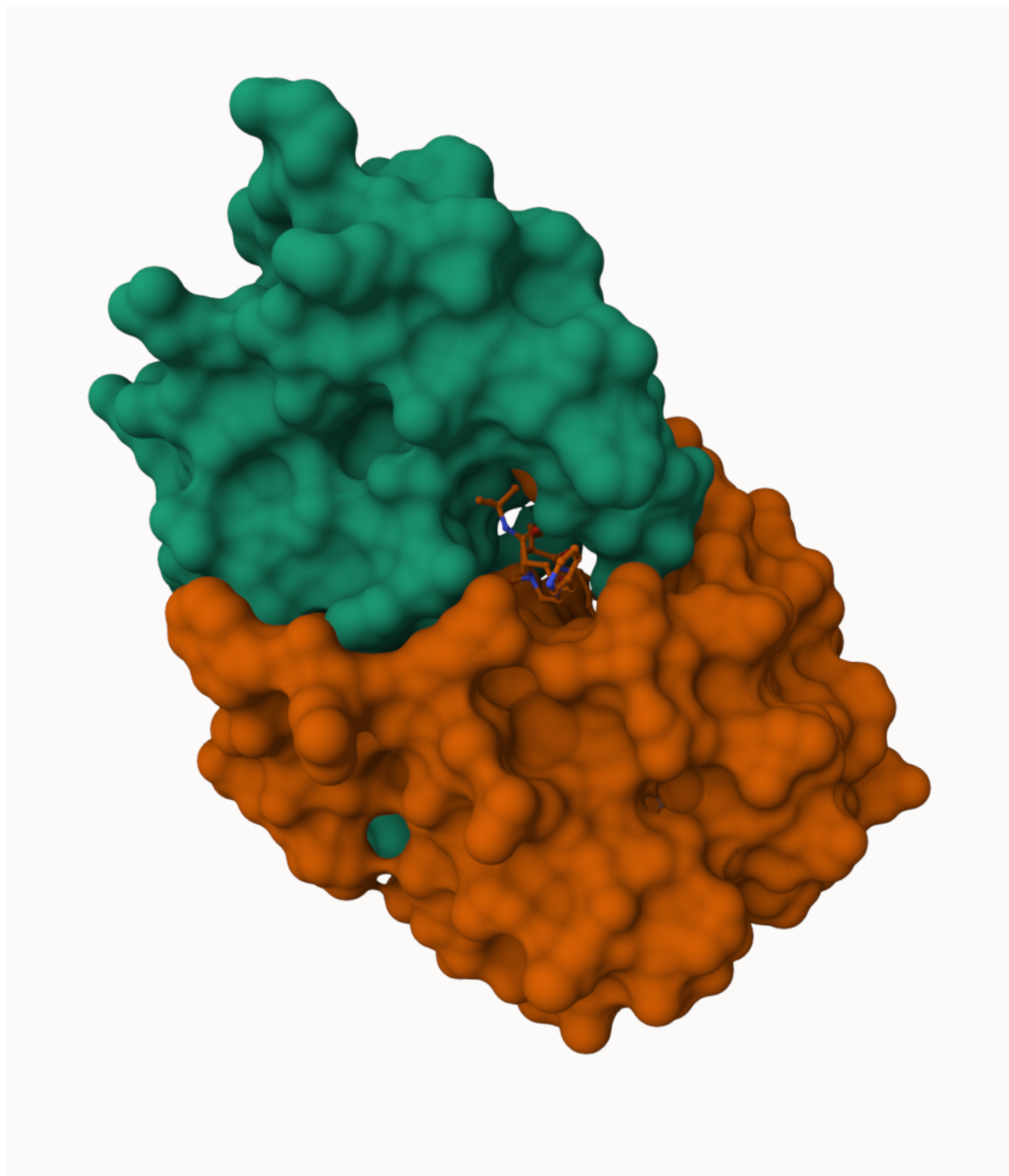


Figure 2: Figure 2. Molecular surface showing binding cavity

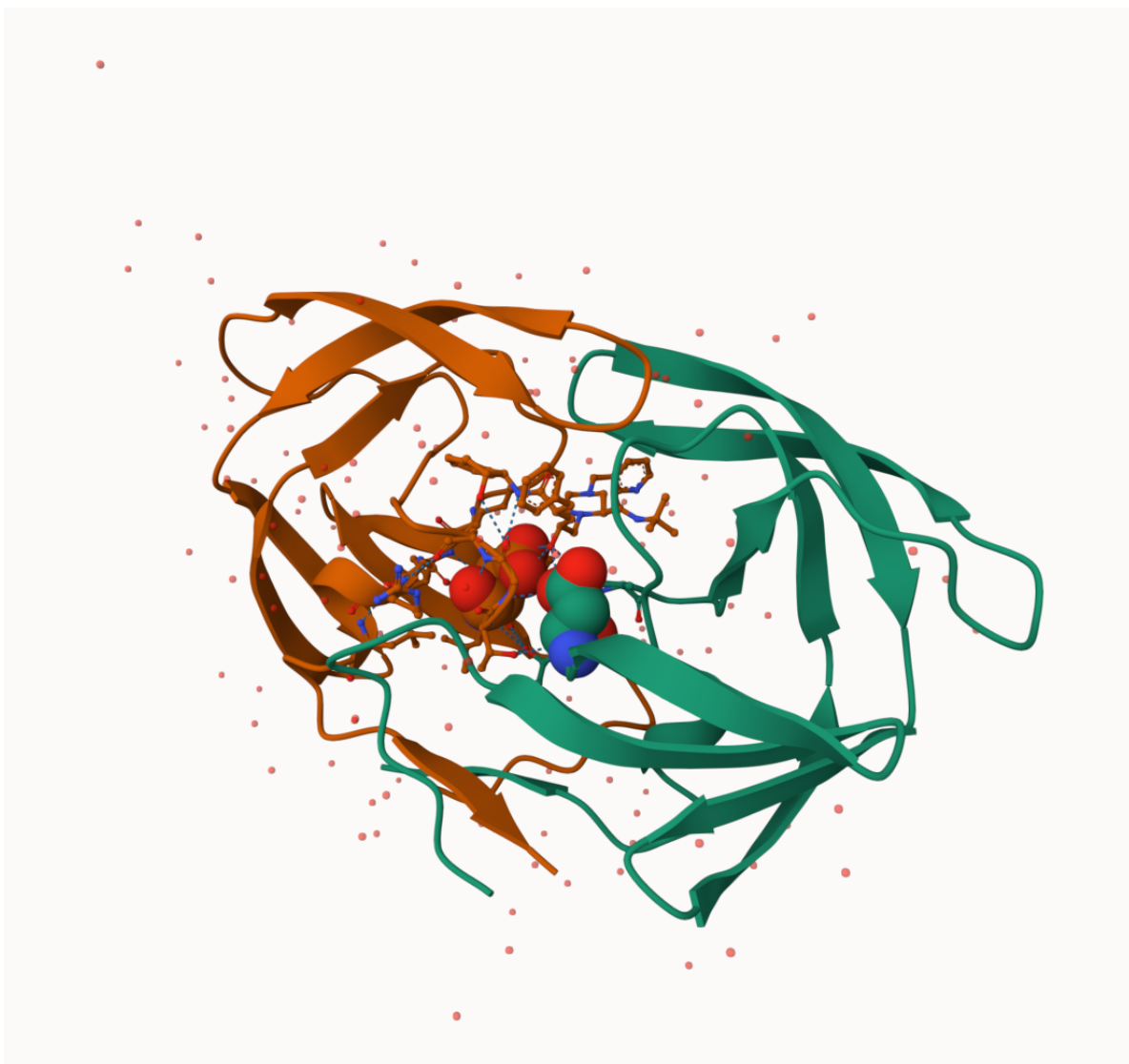


Figure 3: Figure 3. The catalytically important ASP 25 amino acids and drug interacting HOH 308 water molecule

Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure?

Yes, it is true that water molecules have normally 3 atoms (2 hydrogen and 1 oxygen). However, we only see just one atom, oxygen, per water molecule in this structure because of camera resolution we use. The camera resolution used to observe this structure is 2.0Å. However, the size of hydrogen molecule is way smaller than the resolution, which makes just one atom per water molecule be visible.

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.

Yes, I could identify the “conserved” water molecule, HOH 308.

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.

Yes.

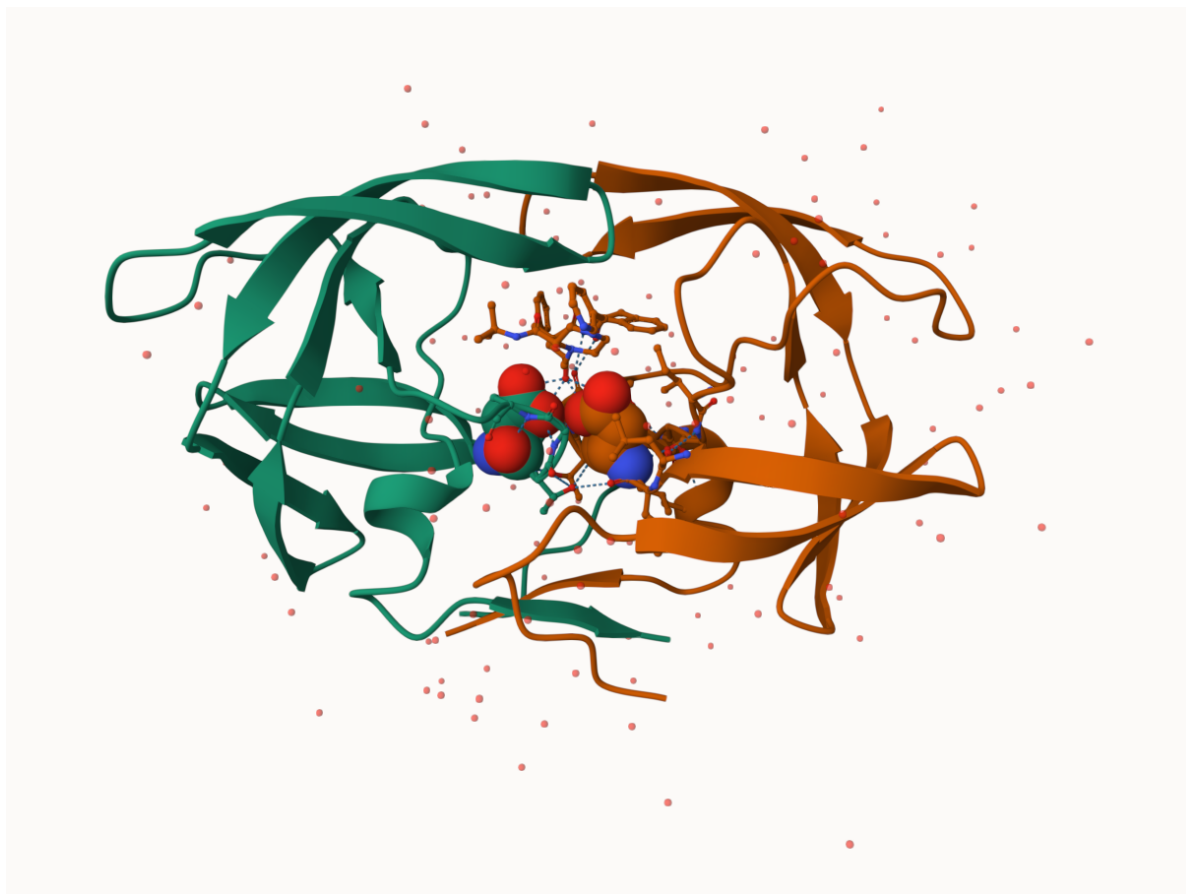


Figure 4: Figure 4. More clear image on catalytic residues ASP 25 in “each” chain and critical water molecule HOH 308

3. Using the bio3d package in R

The Bio3D package is focused on structural bioinformatics analysis and allows us to read and analyze PDB (and related) data.

```
library(bio3d)
```

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

Note: Accessing on-line PDB file

```
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?

There are 198 amino acid residues in this pdb object.

Q8: Name one of the two non-protein residues?

HOH and MK1

Q9: How many protein chains are in this structure?

There are 2 protein chains in this structure: (alpha and beta chain).

```
attributes(pdb)
```

```
$names
```

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

```
$class
```

```
[1] "pdb" "sse"
```

We can see atom data with `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>

```
head(pdbseq(pdb))
```

```
  1    2    3    4    5    6  
"P" "Q" "I" "T" "L" "W"
```

Molecular visualization in R

We can make quick 3D viz with the `view.pdb()` function:

```
library(bio3dview)
library(NGLViewerR)

#view.pdb(pdb, backgroundColor = "pink", colorScheme = "sse")
```

```
sel <- atom.select(pdb, resno=25)

#view.pdb(pdb, cols=c("green", "orange"),
#         highlight=sel,
#         highlight.style="spacefill") |>
# setRock()
```

Predicting functional motions of a single structure

We can finish off today with a bioinformatics prediction of the functional motions of a protein.

We will run a Normal Mode Analysis (NMA)

```
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:
  MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMRLRAAVKSGSELGKQAKDIMDAGKLV
```

```

DELVIALVKERIAQEDCRNGFLLDGFPR TIPQADAMKEAGINVDYVLEFDVPDELIVDKI
VGRRVHAPSGRVYHV KFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
YYSKAEAEAGNTKYAKVDGTPVAEVRADLEKILG

```

```

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

```

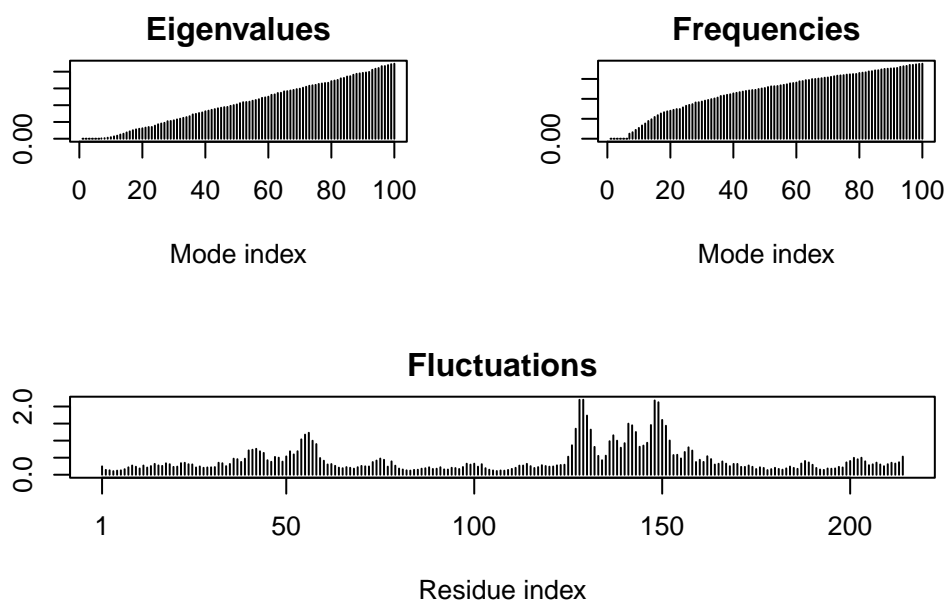
```
m <- nma(adk)
```

```

Building Hessian...      Done in 0.269 seconds.
Diagonalizing Hessian... Done in 1.216 seconds.

```

```
plot(m)
```



```
#view.nma(m)
```

We can write out a trajectory of the predicted dynamics and view this in Mol-star.

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