

Class 10: Structural Bioinformatics pt.1

Sion Kang (PID: A17052234)

2025-02-06

Table of contents

1. The PDB database	1
2. Using Mol*	5
3. Introduction to Bio3D in R	9
4. Predicting functional dynamics	11

1. The PDB database

The main repository of biomolecular structure data is called the PDB found at: <http://www.rcsb.org>

Let's see what this database contains. I went to PDB > Analyze > PDB Statistics > By Exp method and molecular type.

```
pdbstats <- read.csv("Data Export Summary.csv")
pdbstats
```

	Molecular.Type	X.ray	EM	NMR	Multiple.methods	Neutron	Other
1	Protein (only)	169,563	16,774	12,578	208	81	32
2	Protein/Oligosaccharide	9,939	2,839	34	8	2	0
3	Protein/NA	8,801	5,062	286	7	0	0
4	Nucleic acid (only)	2,890	151	1,521	14	3	1
5	Other	170	10	33	0	0	0
6	Oligosaccharide (only)	11	0	6	1	0	4
	Total						
1		199,236					
2		12,822					
3		14,156					

```
4 4,580
5 213
6 22
```

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

```
pdbstats$X.ray
```

```
[1] "169,563" "9,939" "8,801" "2,890" "170" "11"
```

The comma in these numbers is causing them to be read as character rather than numeric.

I can fix this by replacing “,” for nothing “” with the `sub()` function:

```
x <- pdbstats$X.ray
sum (as.numeric( sub(",", "", x) ))
```

```
[1] 191374
```

Or I can use the **readr** package and the `read_csv()` function.

```
library(readr)
pdbstats2 <- read_csv("Data Export Summary.csv")
```

```
Rows: 6 Columns: 8
```

```
-- Column specification -----
```

```
Delimiter: ","
```

```
chr (1): Molecular Type
```

```
dbl (3): Multiple methods, Neutron, Other
```

```
num (4): X-ray, EM, NMR, Total
```

```
i Use `spec()` to retrieve the full column specification for this data.
```

```
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pdbstats2
```

```
# A tibble: 6 x 8
  `Molecular Type`   `X-ray`      EM    NMR `Multiple methods` Neutron Other  Total
  <chr>             <dbl> <dbl> <dbl>          <dbl>   <dbl> <dbl> <dbl>
1 Protein (only)    169563 16774 12578          208     81    32 199236
2 Protein/Oligosacc~ 9939 2839 34           8       2     0 12822
3 Protein/NA        8801 5062 286           7       0     0 14156
4 Nucleic acid (onl~ 2890 151 1521          14       3     1 4580
5 Other             170 10 33            0       0     0 213
6 Oligosaccharide (~ 11 0 6            1       0     4 22
```

I want to clean the column names so they are all lower case and don't have spaces in them

```
colnames(pdbstats2)
```

```
[1] "Molecular Type"  "X-ray"          "EM"             "NMR"
[5] "Multiple methods" "Neutron"        "Other"          "Total"
```

```
library(janitor)
```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

```
chisq.test, fisher.test
```

```
df<- clean_names(pdbstats2)
df
```

```
# A tibble: 6 x 8
  molecular_type      x_ray    em    nmr multiple_methods neutron other  total
  <chr>             <dbl> <dbl> <dbl>          <dbl>   <dbl> <dbl> <dbl>
1 Protein (only)    169563 16774 12578          208     81    32 199236
2 Protein/Oligosacchar~ 9939 2839 34           8       2     0 12822
3 Protein/NA        8801 5062 286           7       0     0 14156
4 Nucleic acid (only) 2890 151 1521          14       3     1 4580
5 Other             170 10 33            0       0     0 213
6 Oligosaccharide (onl~ 11 0 6            1       0     4 22
```

Total number of X-ray structures

```
sum(df$x_ray)
```

```
[1] 191374
```

Total number of structures

```
sum(df$total)
```

```
[1] 231029
```

Percentage of X-ray structures

```
sum(df$x_ray)/sum(df$total) *100
```

```
[1] 82.83549
```

Percentage of EM structures

```
sum(df$em)/sum(df$total) *100
```

```
[1] 10.75017
```

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

```
tot.prot <- df[1,]$total  
tot.prot/sum(df$total)
```

```
[1] 0.8623852
```

2. Using Mol*

The main Mol* homepage at: <https://molstar.org/viewer> We can input our own PDB files or just give it a PDB database accession code (4 letter PDB code).



Figure 1: Molecular view of 1HSG

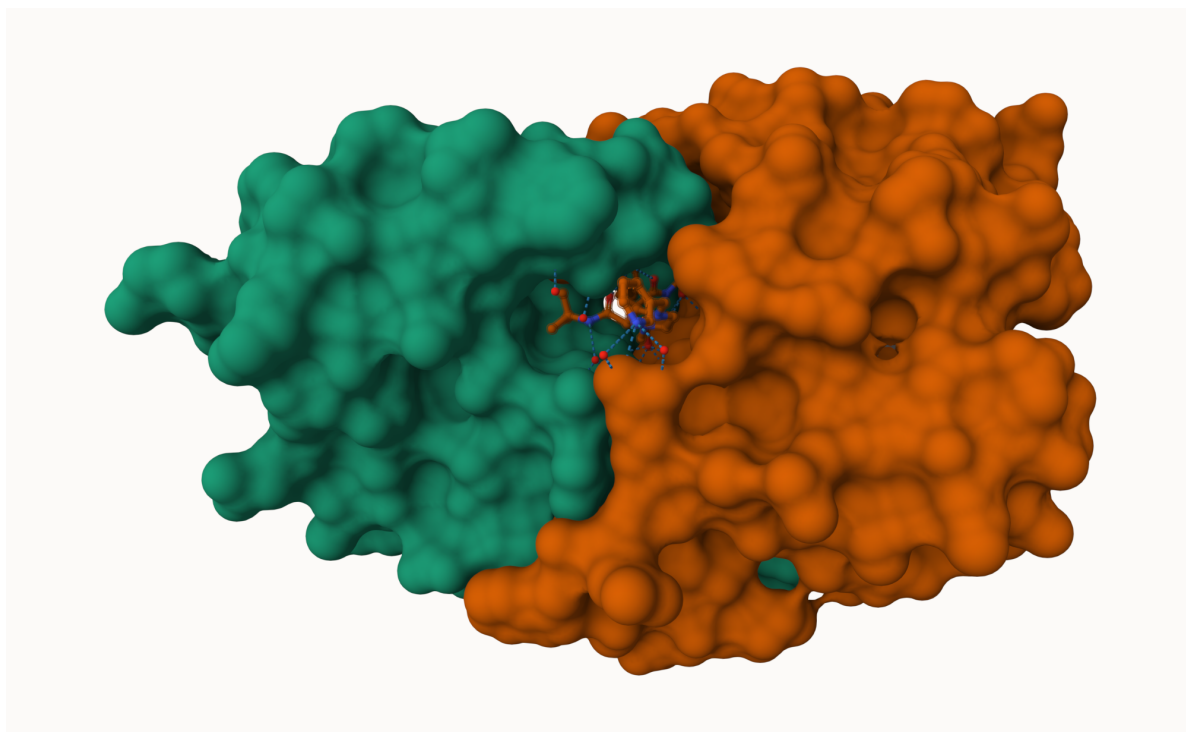


Figure 2: Surface representation showing binding cavity

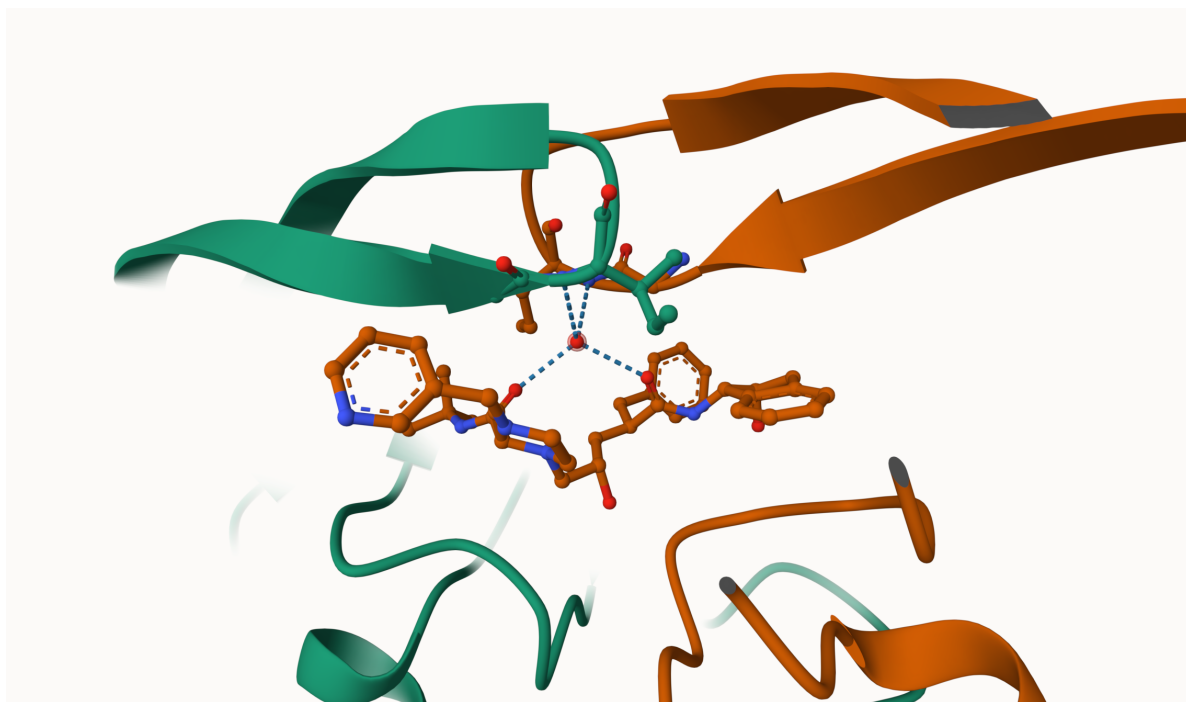


Figure 3: Water 308 in the binding site

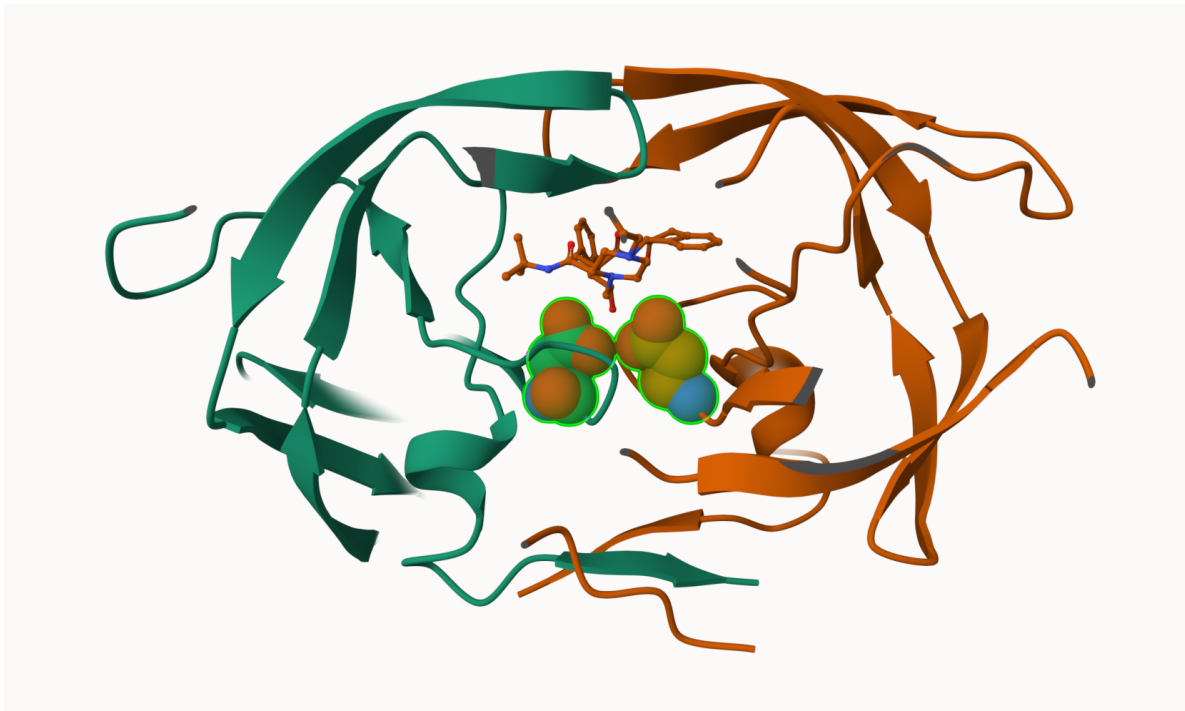


Figure 4: The important ASP25 amino-acids

3. Introduction to Bio3D in R

We can use the **bio3d** package for structural bioinformatics to read PDB data into R

```
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

```
length( pdbseq(pdb) )
```

```
[1] 198
```

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

MK1

Q9: How many protein chains are in this structure?

2 chains: A and B

Looking at the pdb object in more detail

```
attributes(pdb)
```

```
$names
```

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

```
$class
```

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

```
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>

Let's try a new function not yet in the bio3d package. It requires the **r3dmol** package that we need to install with `install.packages("r3dmol")` and `install.packages("shiny")`

```
source("https://tinyurl.com/viewpdb")  
#view.pdb(pdb, backgroundColor="pink")
```

4. Predicting functional dynamics

We can use the `nma()` function in bio3d to predict the large-scale functional motions of biomolecules.

```
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:

```
MRIILLGAPGAGKGTQAFIMEKYGIPQISTGDMRLRAAVKSGSELGKQAKDIMDAGKLV  
TDELVIALVKERIAQEDCRNGFLDGFPRITPQADAMKEAGINVDYVLEFDVPDELIVDKI  
VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQM  
TAPLIGYYSKEAEAGNTKYAKVDGTPVAEVRADLEKILG
```

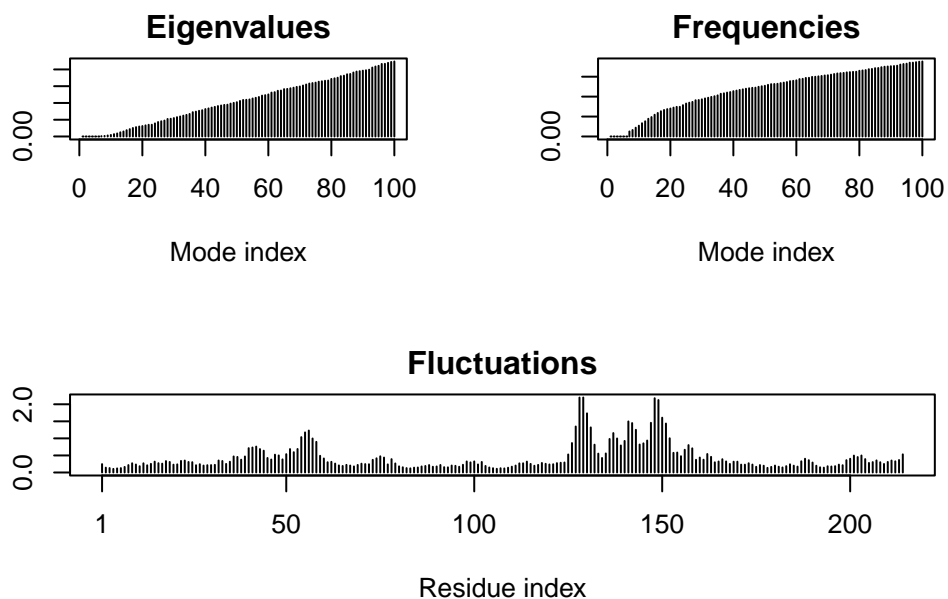
+ attr: atom, xyz, seqres, helix, sheet,
calpha, remark, call

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

Building Hessian... Done in 0.018 seconds.

Diagonalizing Hessian... Done in 0.417 seconds.

```
plot(m)
```



Write out a trajectory of the predicted molecular motion:

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