Class 9: Structural Bioinformatics 1

Gaeun Jun (PID: A16814573)

The main database for structural data is called the PDB (Protein Data Bank). Let's see what it contains:

Data from: https://www.rcsb.org/stats URL:https://bioboot.github.io/bimm143_F24/class-material/pdb_stats.csv

Read this into R

```
pdb_statistics <- read.csv("pdb_stats.csv", row.names=1)
pdb_statistics
```

```
                         X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)         167,192 15,572 12,529              208      77    32
Protein/Oligosaccharide  9,639  2,635     34                8       2     0
Protein/NA               8,730  4,697    286                7       0     0
Nucleic acid (only)      2,869    137  1,507               14       3     1
Other                      170     10     33                0       0     0
Oligosaccharide (only)      11      0      6                1       0     4
                         Total
Protein (only)         195,610
Protein/Oligosaccharide  12,318
Protein/NA               13,720
Nucleic acid (only)       4,531
Other                       213
Oligosaccharide (only)       22
```

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

```
pdb_statistics$Total
```

```
[1] "195,610" "12,318"  "13,720"  "4,531"   "213"     "22"
```

I need to remove the comma and convert it to numeric in order to do math:

```
# get rid of the commas to make it integers
as.numeric(sub(",", "", pdb_statistics$Total))
```

```
[1] 195610  12318  13720   4531    213     22
```

I could turn this into a function to fix the whole table or any future table I read like this:

```
x <- pdb_statistics$Total
as.numeric(sub(",", "", x))
```

```
[1] 195610  12318  13720   4531    213     22
```

```
comma2numeric <- function(x) {
 as.numeric(sub(",", "", x))
}
```

Test it

```
comma2numeric(pdb_statistics$X.ray)
```

```
[1] 167192   9639   8730   2869    170     11
```

```
apply(pdb_statistics, 2, comma2numeric)
```

```
      X.ray    EM   NMR Multiple.methods Neutron Other  Total
[1,] 167192 15572 12529              208      77    32 195610
[2,]   9639  2635    34                8       2     0  12318
[3,]   8730  4697   286                7       0     0  13720
[4,]   2869   137  1507               14       3     1   4531
[5,]    170    10    33                0       0     0    213
[6,]     11     0     6                1       0     4     22
```

Or try a different read/import function:

```
library(readr)
pdb_statistics <- read_csv("pdb_stats.csv")
```

```
sum(pdb_statistics$Total)
```

```
[1] 226414
```

```
sum(pdb_statistics$`X-ray`)/sum(pdb_statistics$Total)*100
```

```
[1] 83.30359
```

```
sum(pdb_statistics$EM)/sum(pdb_statistics$Total)*100
```

```
[1] 10.18091
```

X-ray: 83.30% Electron Microscopy: 10.18%

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

```
pdb_statistics[1, "Total"]
```

```
# A tibble: 1 x 1
   Total
   <dbl>
1 195610
```

Calculating the total amount of structures.

```
sum(pdb_statistics[,"Total"])
```

```
[1] 226414
```

Dividing the total proteins by the total amount.

```
pdb_statistics[1, "Total"]/sum(pdb_statistics[,"Total"])*100
```

```
     Total
1 86.39483
```

Or

```
pdb_statistics$Total[1]/sum(pdb_statistics$Total)*100
```

```
[1] 86.39483
```

Protein: 86.39%

> 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?

There are 4563 structures

Mol*

Mol* (pronounced "molstar") is a new web-based molecular viewer that we will need to learn the basics of here.

https://molstar.org/viewer/

We will use PDB code: 1HSG

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

We see only one atom per water molecule, representing oxygen. This is most likely because the structure is alread complex and they want to minimize/simplify the representations of the water molecules.

> 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?

This water molecule has the residue number 308.

Figure 1: A first image from molstar

Some more custom images:

Figure 2: The all important catalytic ASP25 amino acids

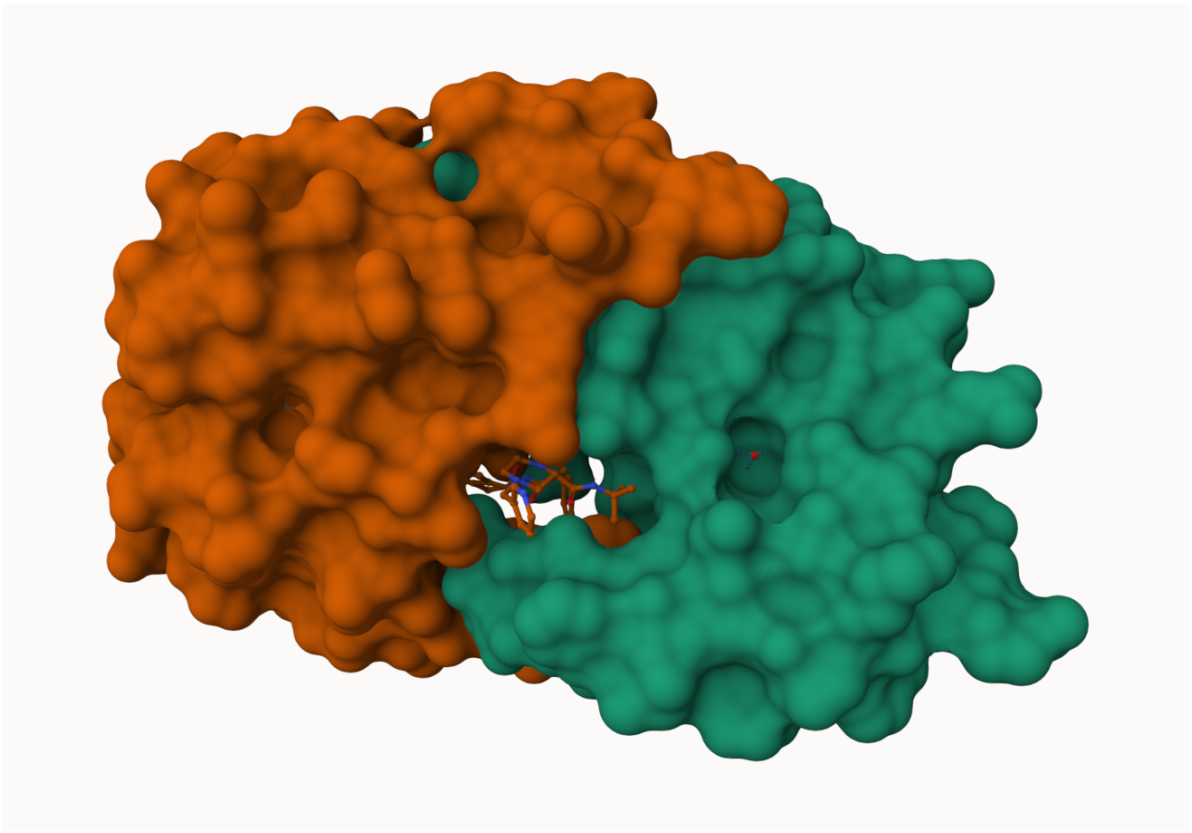


Figure 3: Surface display

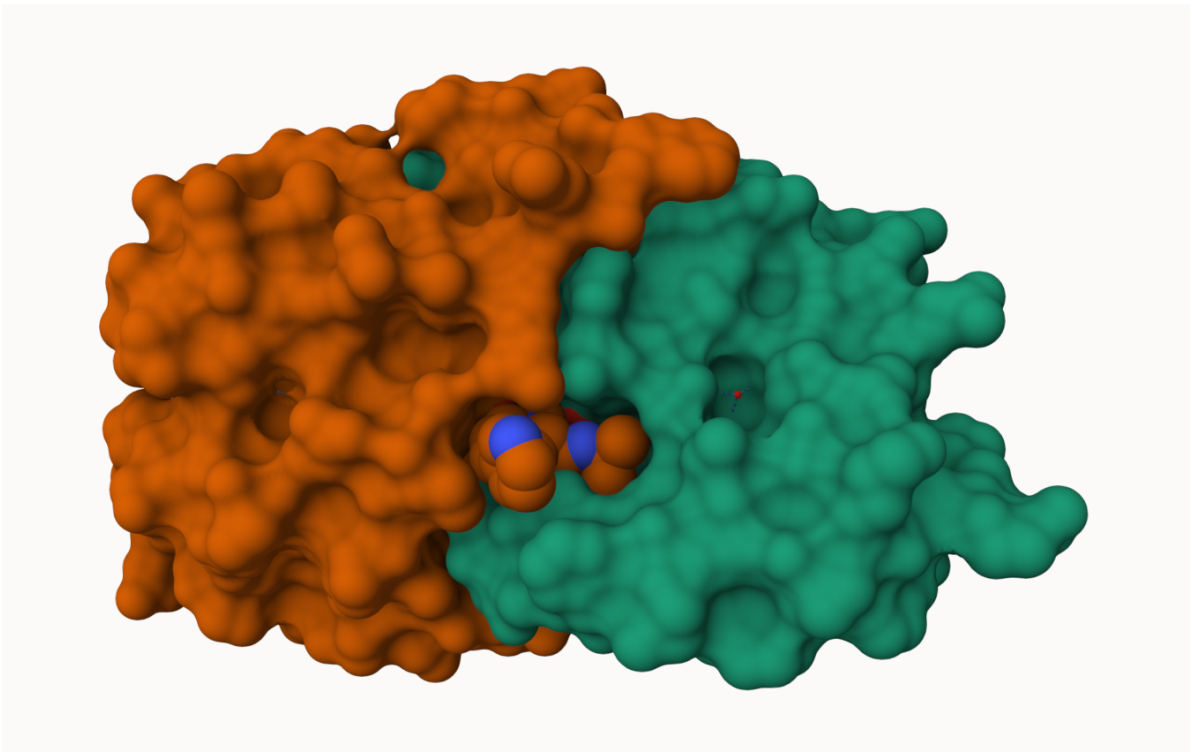
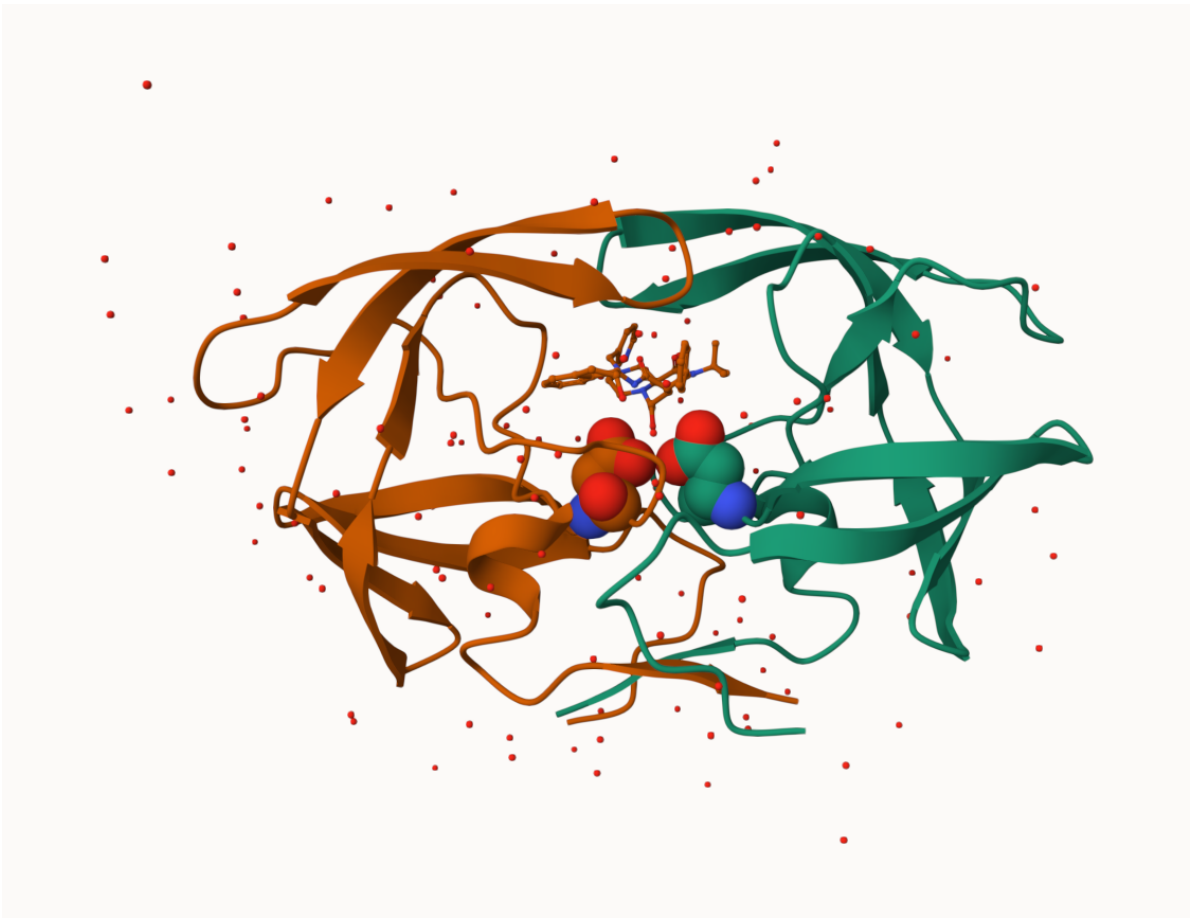


Figure 4: Surface display showing the ligand fitting into the compound

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.



The Bio3D package

The bio3d package allows us to do all sorts of structural bioinformatics work in R.

Let's start with how it can read these PDB files:

```
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
```

```
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>
```

```
pdbseq(pdb)[25]
```

```
 25
"D"
```

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

```
sum(pdb$calpha)
```

```
[1] 198
```

or

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

```
[1] 198
```

198 calpha residues

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

HOH and MK1

Q9: How many protein chains are in this structure?

```
unique(pdb$atom$chain)
```

```
[1] "A" "B"
```

2 protein chains

Predicting functional motions of a single structure

Let's do a bioinformatics prediction of functional motions - i.e. the movements that one of these molecules needs to make to do its stuff.

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

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

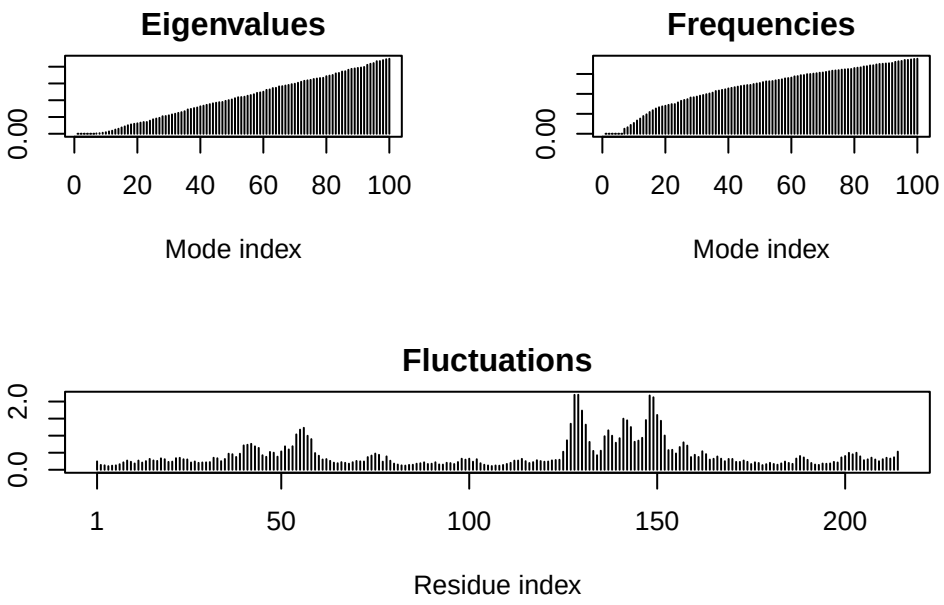
```
adk
```

```
 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
```

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

```
 Building Hessian...        Done in 0.015 seconds.
 Diagonalizing Hessian...   Done in 0.277 seconds.
```

```
plot(m)
```



Write out multi-model PDB file (trajectory) that we can use to make an animation of the predicted motions.

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

I can open this in Mol* to play the trajectory…