Class 10: Structural Bioinformatics (pt1)

Cienna Santos (PID: A17581026)

Table of contents

1. The PDB Database

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

What is currently in the PDB?

```
stats <- read.csv("Data_Export_Summary.csv")
stats
```

```
           Molecular.Type   X.ray     EM    NMR Multiple.methods Neutron Other
1          Protein (only) 171,959 18,083 12,622              210      84    32
2 Protein/Oligosaccharide  10,018  2,968     34               10       2     0
3              Protein/NA   8,847  5,376    286                7       0     0
4     Nucleic acid (only)   2,947    185  1,535               14       3     1
5                   Other     170     10     33                0       0     0
6  Oligosaccharide (only)      11      0      6                1       0     4
    Total
1 202,990
2  13,032
3  14,516
4   4,685
5     213
6      22
```

> 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"
```

This column is all strings! Because there are commas.

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

```
[1] 193952
```

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
  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
```

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

```r
rownames(stats) <- stats$Molecular.Type

proteins <- comma.sum(stats["Protein (only)", "Total"])
proteins/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?

1,149

2. Visualizing with Mol-Star

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

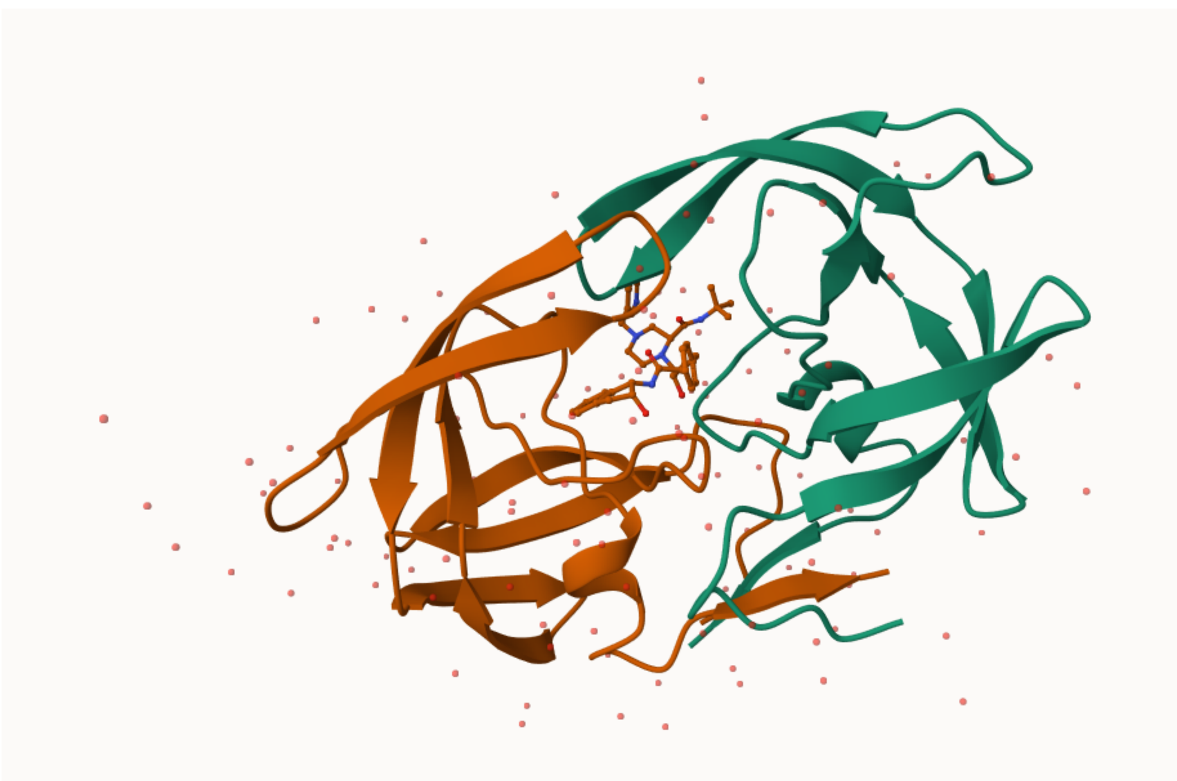


Figure 1: Figure 1. A first view of HIV-Pr

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

Because the resolution of these models are 2 Angstroms, and hydrogen is smaller than that.

> 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

308 water molecule. It's between the two dimers.

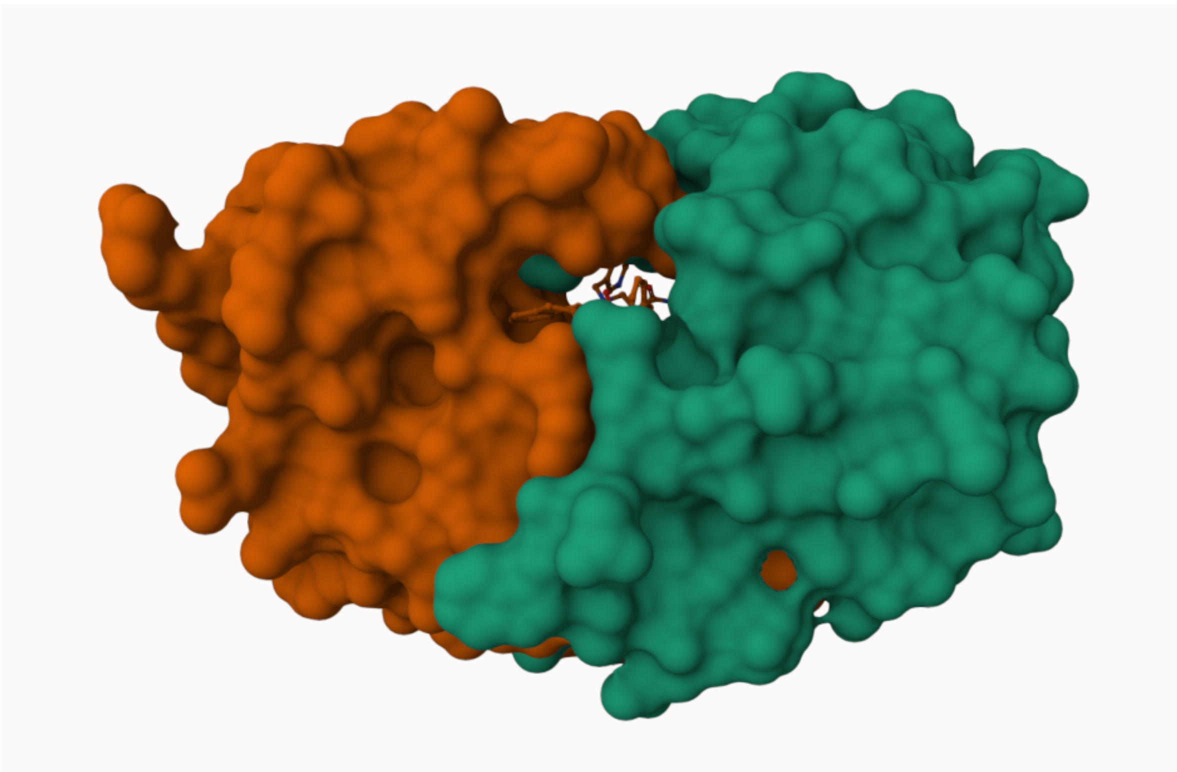


Figure 2: Figure 2. Molecular surface showing binding cavity

> 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.

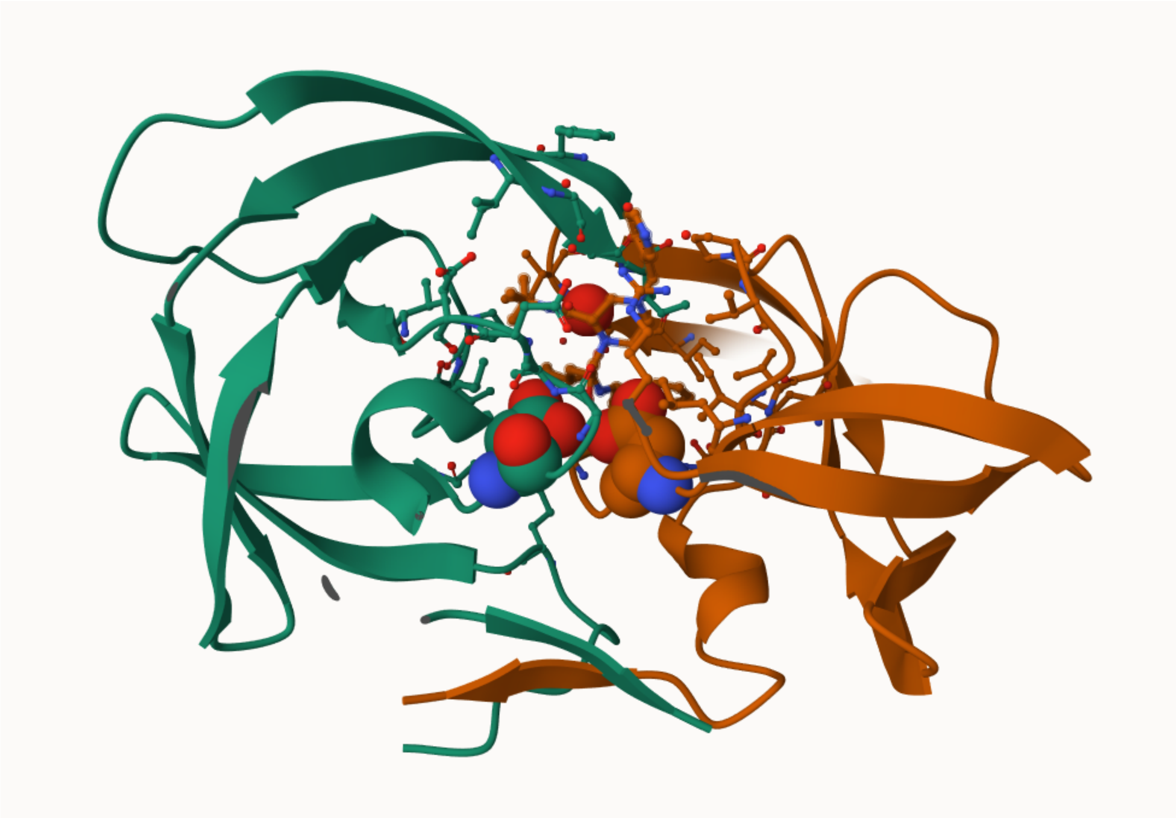


Figure 3: Figure 3. The catatilically important ASP 25 amino acids and drug interacting HOH 308 water molecule

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
```

```r
attributes(pdb)
```

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

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

We can see atom data with `pdb$atom`:

```r
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>
```

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

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

```r
# install.packages("pak")
# pak::pak("bioboot/bio3dview")
# install.packages("NGLVieweR")
```

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

```r
library(bio3dview)
library(NGLVieweR)

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

```r
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 funcitonal motions of a protein.

We will run a Normal Mode Analysis

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

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

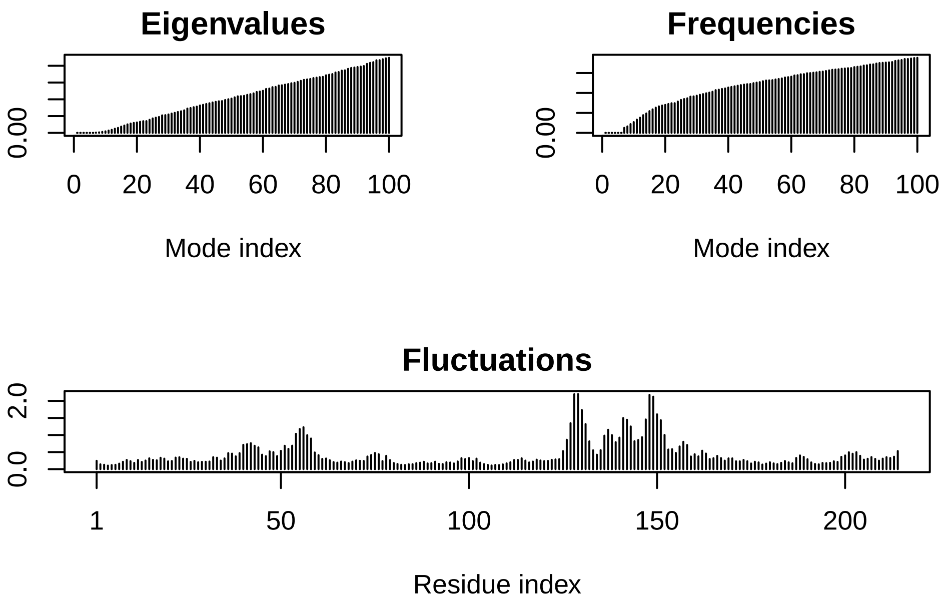
```r
adk
```

```
 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
```

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

```
 Building Hessian...        Done in 0.041 seconds.
 Diagonalizing Hessian...   Done in 0.669 seconds.
```

```r
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")
```