# Class 10: Structural Bioinformatics (pt. 1)

Nicole Alfonso, (PID: A16429176)

## The PDB Database

We are examining the size and composition of the main databse of biomolecular structures - PDB.

```
pdbstats <- read.csv('~/BIMM143/class10/Data Export Summary.csv', row.names = 1)
head(pdbstats)
```

```
                        X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)        161,663 12,592 12,337              200      74    32
Protein/Oligosaccharide 9,348  2,167     34                8       2     0
Protein/NA              8,404  3,924    286                7       0     0
Nucleic acid (only)     2,758    125  1,477               14       3     1
Other                     164      9     33                0       0     0
Oligosaccharide (only)     11      0      6                1       0     4
                        Total
Protein (only)        186,898
Protein/Oligosaccharide 11,559
Protein/NA              12,621
Nucleic acid (only)      4,378
Other                      206
Oligosaccharide (only)      22
```

> **Q1.** What percentage of structures in the PDB are solved by X-Ray and Electron Microscopy.
>
> 93.26839 -> 93.27%

How to solve the problem by extracting the data from the data set:

```r
# My function for removing commas from a character vector, then modifying the character ve
character_to_interger <- function(dataset_column) {
  remove_commas <- gsub(",", "", dataset_column)
  interger_data <- as.integer(remove_commas)
  interger_data
}
```

```r
# Isolating the xray column
xray <- pdbstats$X.ray
# Using character_to_interger function
xray_int <- character_to_interger(xray)
xray_sum <- sum(xray_int)
xray_sum
```

```
[1] 182348
```

```r
electron_microscopy <- pdbstats$EM
em_int <- character_to_interger(electron_microscopy)
em_sum <- sum(em_int)
em_sum
```

```
[1] 18817
```

```r
# Combining the xray and electron microscopy methods into one vector
xray_em <- xray_int + em_int
# Calculating the sum of all values in the vector - how many times the xray and em methods
sum_xray_em <- sum(xray_em)
sum_xray_em
```

```
[1] 201165
```

```r
#Calculating the total experimental methods
total <- pdbstats$Total
total_int <- character_to_interger(total)
sum_total <- sum(total_int)
sum_total
```

```
[1] 215684
```

```
# Calculating the percentage
percent_xray_em <- round((sum_xray_em/sum_total)*100, 2)
percent_xray_em
```

```
[1] 93.27
```

**Q2.** What proportion of structures in the PDB are protein?

```
protein_total <- character_to_interger(pdbstats[1,7])
round(protein_total/sum_total*100, 2)
```

```
[1] 86.65
```

Q > Proportion

```
(215684/249751891)*100
```

```
[1] 0.08635931
```

## Visualizing the HIV-1 Protease Structure

We are learning the basics of the Mol*, mol-star, homepage: https://molstar.org/viewer/

We will be analyzing the PDB code 1 HSG

**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.

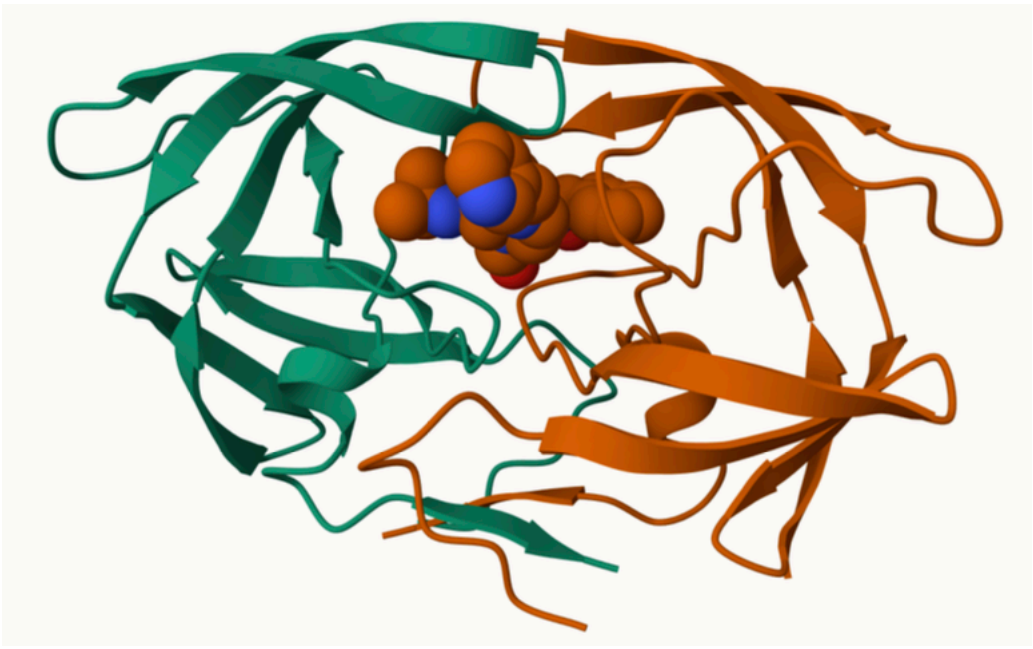


Figure 1: HIV-Protease with a bound inhibitor, including two APS 25 amino acids

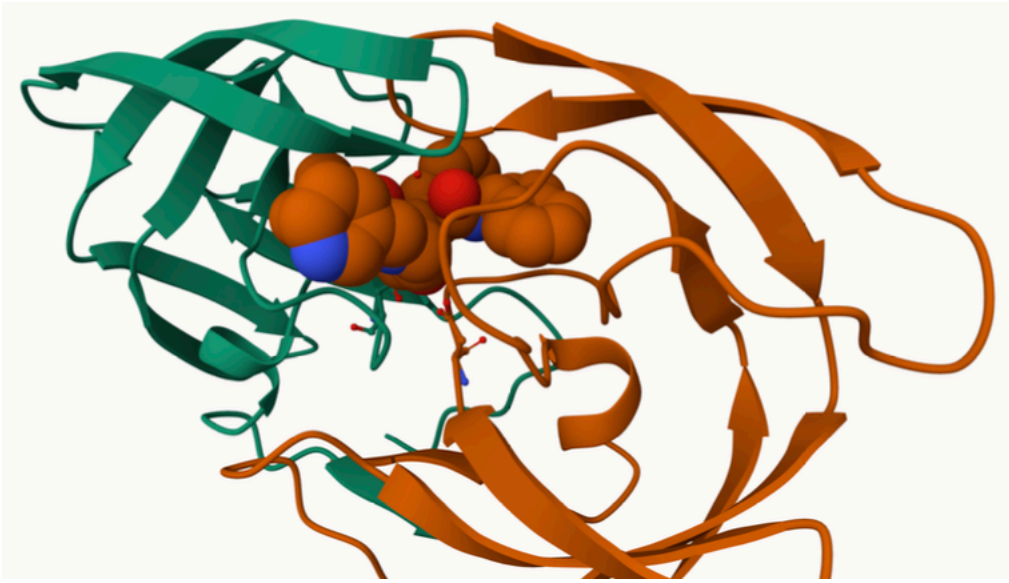
**Q4**. Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure?

Water is composed of three atoms, two Hydrogen and one Oxygen. However, because Hydrogen is too small, it is omitted from the sequence, making only the

one Oxygen atom 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

The conserved water molecule: HOH 308



## R and Working with PDB Structures

Predicting the dynamics/flexibility of an important protein

```
library(bio3d)
hiv <- read.pdb(file = '1hsg')
```

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

```
hiv
```

```
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

**Q8.** Name one of the two non-protein residues.

MK1

**Q9.** How many protein chains are in this protein structure?

2 protein chains

```
head(hiv$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(hiv)
```

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
"P" "Q" "I" "T" "L" "W" "Q" "R" "P" "L" "V" "T" "I" "K" "I" "G" "G" "Q" "L" "K"
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
"E" "A" "L" "L" "D" "T" "G" "A" "D" "D" "T" "V" "L" "E" "E" "M" "S" "L" "P" "G"
 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
"R" "W" "K" "P" "K" "M" "I" "G" "G" "I" "G" "G" "F" "I" "K" "V" "R" "Q" "Y" "D"
 61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
"Q" "I" "L" "I" "E" "I" "C" "G" "H" "K" "A" "I" "G" "T" "V" "L" "V" "G" "P" "T"
 81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99   1
"P" "V" "N" "I" "I" "G" "R" "N" "L" "L" "T" "Q" "I" "G" "C" "T" "L" "N" "F" "P"
  2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
"Q" "I" "T" "L" "W" "Q" "R" "P" "L" "V" "T" "I" "K" "I" "G" "G" "Q" "L" "K" "E"
 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
"A" "L" "L" "D" "T" "G" "A" "D" "D" "T" "V" "L" "E" "E" "M" "S" "L" "P" "G" "R"
 42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61
"W" "K" "P" "K" "M" "I" "G" "G" "I" "G" "G" "F" "I" "K" "V" "R" "Q" "Y" "D" "Q"
 62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81
"I" "L" "I" "E" "I" "C" "G" "H" "K" "A" "I" "G" "T" "V" "L" "V" "G" "P" "T" "P"
 82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99
"V" "N" "I" "I" "G" "R" "N" "L" "L" "T" "Q" "I" "G" "C" "T" "L" "N" "F"
```

We will be doing Normal Mode Analysis (NMA) to predict the function motions of a kinase protein.

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

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

```
adk
```

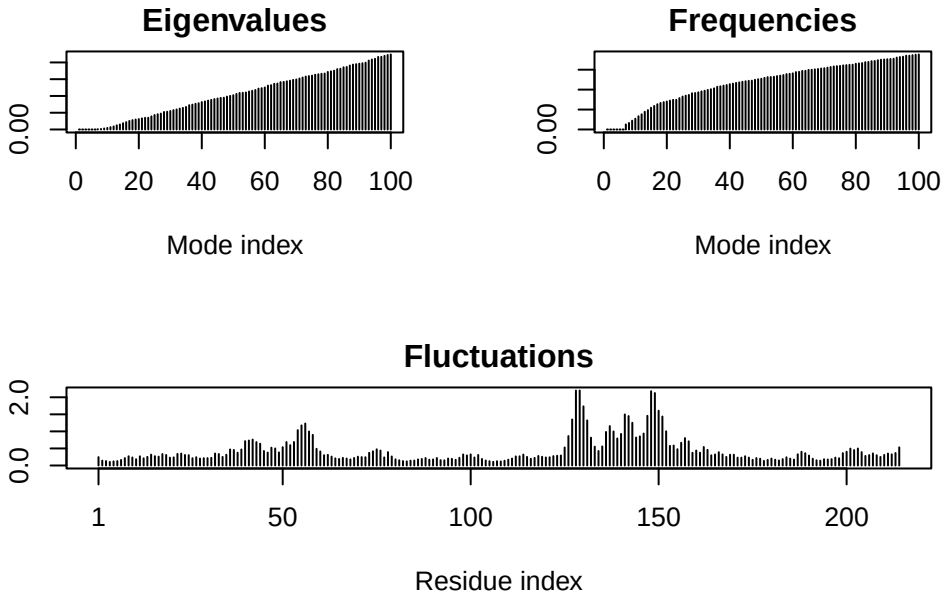
```
Call:  read.pdb(file = "6s36")

  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
```

```
modes <- nma(adk)
```

```
 Building Hessian...        Done in 0.021 seconds.
 Diagonalizing Hessian...   Done in 0.456 seconds.
```

```
plot(modes)
```



Now, make a “movie” called a trajectory of the predicted motion:

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

This file can be opened in Mol*