# Class 09: Structural Bioinformatics

Jenny Zhou

## PDB statistics

Download a CSV file from the PDB site (accessible from "Analyze" > "PDB Statistics" > "by Experimental Method and Molecular Type". Move this CSV file into your RStudio project and use it to answer the following questions:

```
db <- read.csv("DataExportSummary.csv")
db
```

```
           Molecular.Type   X.ray     EM    NMR Multiple.methods Neutron Other
1          Protein (only) 154,766 10,155 12,187              191      72    32
2 Protein/Oligosaccharide   9,083  1,802     32                7       1     0
3              Protein/NA   8,110  3,176    283                6       0     0
4     Nucleic acid (only)   2,664     94  1,450               12       2     1
5                   Other     163      9     32                0       0     0
6  Oligosaccharide (only)      11      0      6                1       0     4
    Total
1 177,403
2  10,925
3  11,575
4   4,223
5     204
6      22
```

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

```
#use x as input
num <- function(x) {
  #substitute comma, and convert to numeric
  as.numeric(gsub(",", "", x))
```

```
}

#use x as input
total <- function(x) {
  #substitute comma in each element, convert them to numeric, and sum up the vector
  sum(as.numeric(gsub(",", "", x)))
}
```

```
sum(num(db$EM)) / sum(num(db$Total))
```

```
[1] 0.07455763
```

```
sum(num(db$X.ray)) / sum(num(db$Total))
```

```
[1] 0.8553721
```

```
(sum(num(db$X.ray)) + sum(num(db$EM))) / sum(num(db$Total)) *100
```

```
[1] 92.99297
```

```
round(( total(db$X.ray) + total(db$EM) ) / total(db$Total)*100,2)
```

```
[1] 92.99
```

totally 92.99% of structures in the PDB are solved by X-Ray and Electron Microscopy.

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

```
round (total(db$Total[1]) / total(db$Total) *100, 2)
```

```
[1] 86.81
```

86.81% of structures in the PDB are protein.

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?

4926

## Visualizing the HIV-1 protease structure

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

All hydrogen atoms are hidden because they are too small.

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

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.

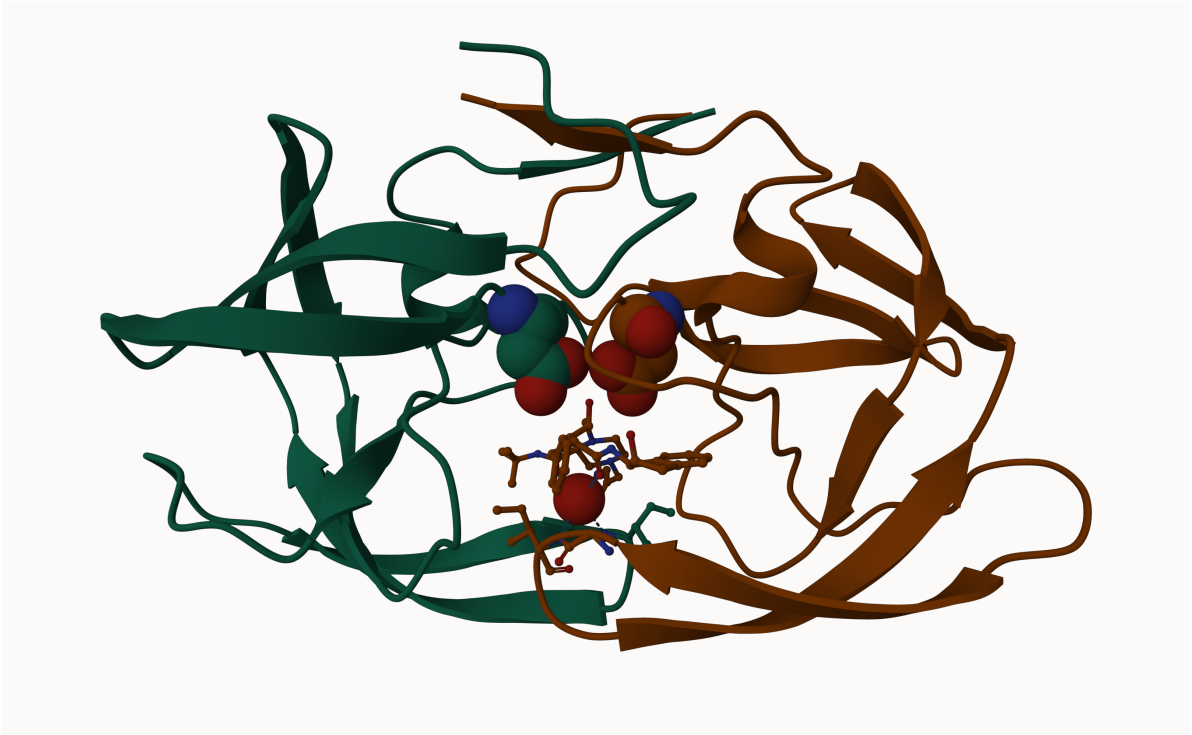


Figure 1: HIV protease structure from MERK with a bound drug

Discussion Topic: Can you think of a way in which indinavir, or even larger ligands and substrates, could enter the binding site?

After the 'flaps' or 'arms' of protease open, larger substrate will be able to enter the binding site.

## Introduction to Bio3D in R

We can use the `bio3d` package to read and perform bioninformatics calculations on PDB structures.

```
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"
```

```
pdb$helix
```

```
$start

87 87

$end

90 90

$chain
[1] "A" "B"

$type
[1] "1" "1"
```

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

198 residues (99 residues in each chain)

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

HOH (127), MK1 (1)

Q9: How many protein chains are in this structure?

2

Read an ADK structure

```
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
```

### Predicting functional motions of a single structure

Perform a prediction of flexibility with a technique called NMA (normal mode analysis)

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

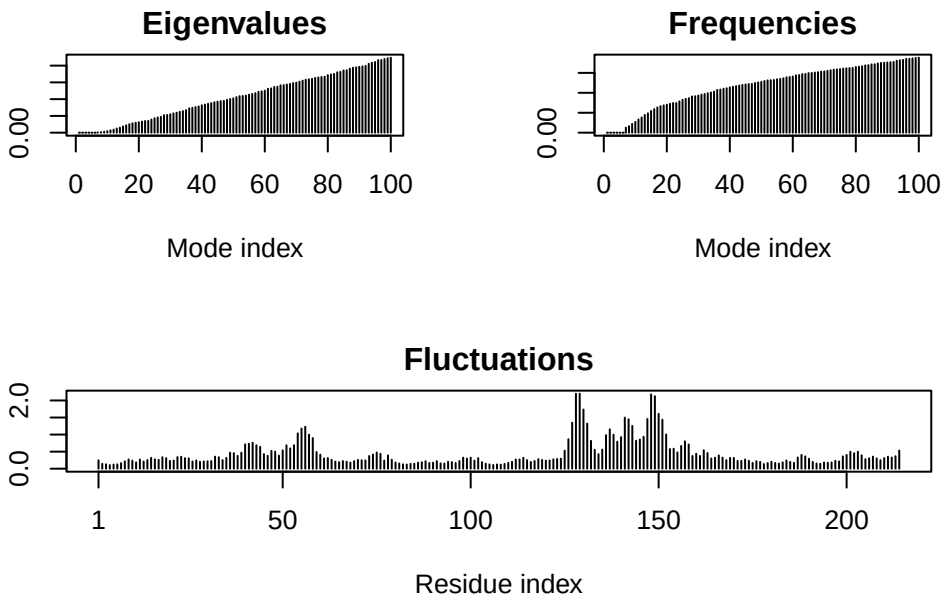
```
 Building Hessian...        Done in 0.06 seconds.
 Diagonalizing Hessian...   Done in 0.48 seconds.
```

```
m
```

```

Call:
  nma.pdb(pdb = adk)

Class:
  VibrationalModes (nma)
```

```
Number of modes:
  642 (6 trivial)

Frequencies:
  Mode 7:   0.005
  Mode 8:   0.007
  Mode 9:   0.009
  Mode 10:  0.011
  Mode 11:  0.013
  Mode 12:  0.015

+ attr: modes, frequencies, force.constants, fluctuations,
        U, L, xyz, mass, temp, triv.modes, natoms, call
```

```
plot(m)
```




Write out (create) a "movie" ("trajectory") of the motion for viewing in Molstar

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

the created file (in our own computer) can be read in Molstar