Class 09: Structural Bioinformatics I

Neha Deshpande (PID: A17567541)

1. The PDB Database

Here we examine the size and composition of the main database of proteins, PDB.

Get a CSV file and read it into R.

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

```
           Molecular.Type   X.ray     EM    NMR Multiple.methods Neutron Other
1          Protein (only) 161,663 12,592 12,337              200      74    32
2 Protein/Oligosaccharide   9,348  2,167     34                8       2     0
3              Protein/NA   8,404  3,924    286                7       0     0
4     Nucleic acid (only)   2,758    125  1,477               14       3     1
5                   Other     164      9     33                0       0     0
6  Oligosaccharide (only)      11      0      6                1       0     4
    Total
1 186,898
2  11,559
3  12,621
4   4,378
5     206
6      22
```

```
#View(pdb_data)
```

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

My pdb_data data frame has numbers with commas in them. This may cause us problems. Let's see.

```
pdb_data$X.ray
```

```
[1] "161,663" "9,348"   "8,404"   "2,758"   "164"     "11"
```

These are showing up as character vectors. We can susbtitute the comma with an empty string using the `gsub()` function.

```
x<-"22,200"
sum(as.numeric(gsub(",","",x)))
```

```
[1] 22200
```

```
commasum<- function(x){
  sum(as.numeric(gsub(",","",x)))
}

commasum(pdb_data$X.ray)
```

```
[1] 182348
```

```
totals<-apply(pdb_data,2,commasum)
```

```
Warning in FUN(newX[, i], ...): NAs introduced by coercion
```

```
round(totals/totals["Total"]*100,2)
```

```
  Molecular.Type            X.ray               EM              NMR
              NA            84.54             8.72             6.57
Multiple.methods          Neutron            Other            Total
            0.11             0.04             0.02           100.00
```

Ans: Percent of X-ray is 84.54, and percent of EM is 8.72.

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

This was giving me an error before, because it was reading the numbers as characters because of the comma, even if I use the `as.numeric()` function. Instead, I could just use the `commasum()` function that I made because losing a comma is built into the function.

```
protein<-commasum(pdb_data[1,8])
protein/commasum(pdb_data$Total)
```

```
[1] 0.8665362
```

2. Visualizing Protein Structure

We will learn the basics of Mol* (mol-star) homepage: https://molstar.org/viewer/

We will play with PDB code 1HSG

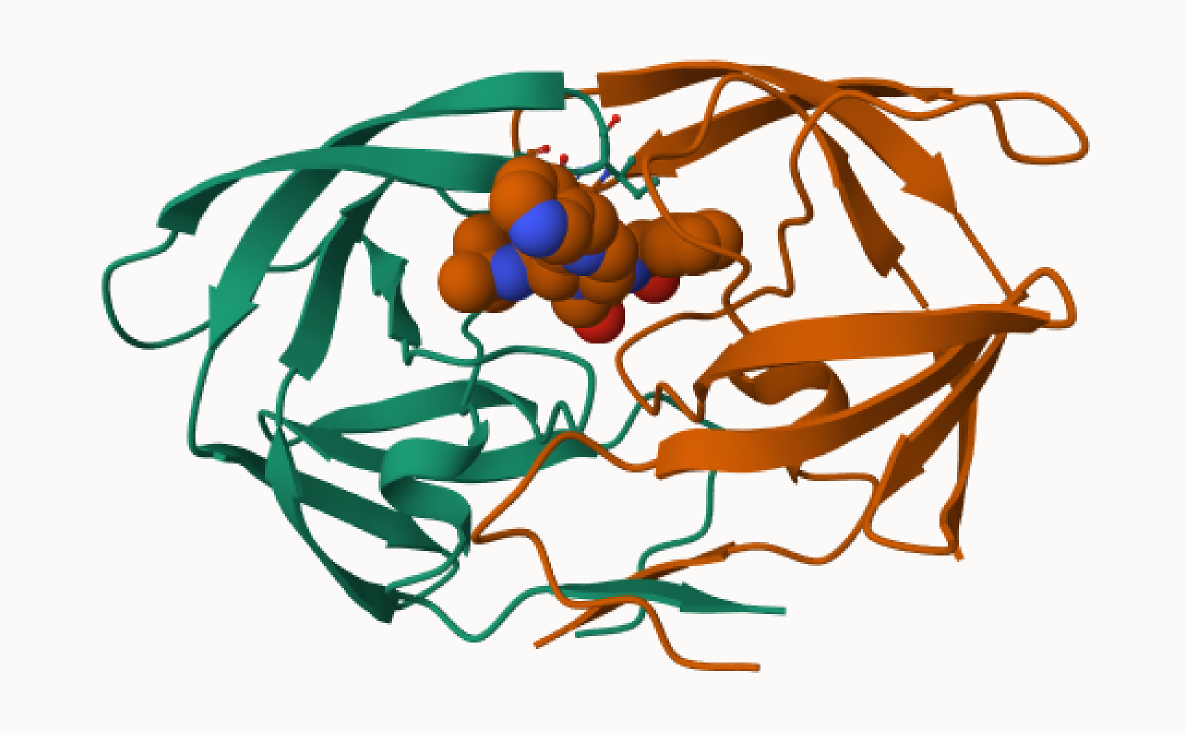


Figure 1: Picture of binding of the small molecule to the HIV protease

3. Back to R and working with PDB structures:

Predict the dynamics (flexibility) of an important protein:

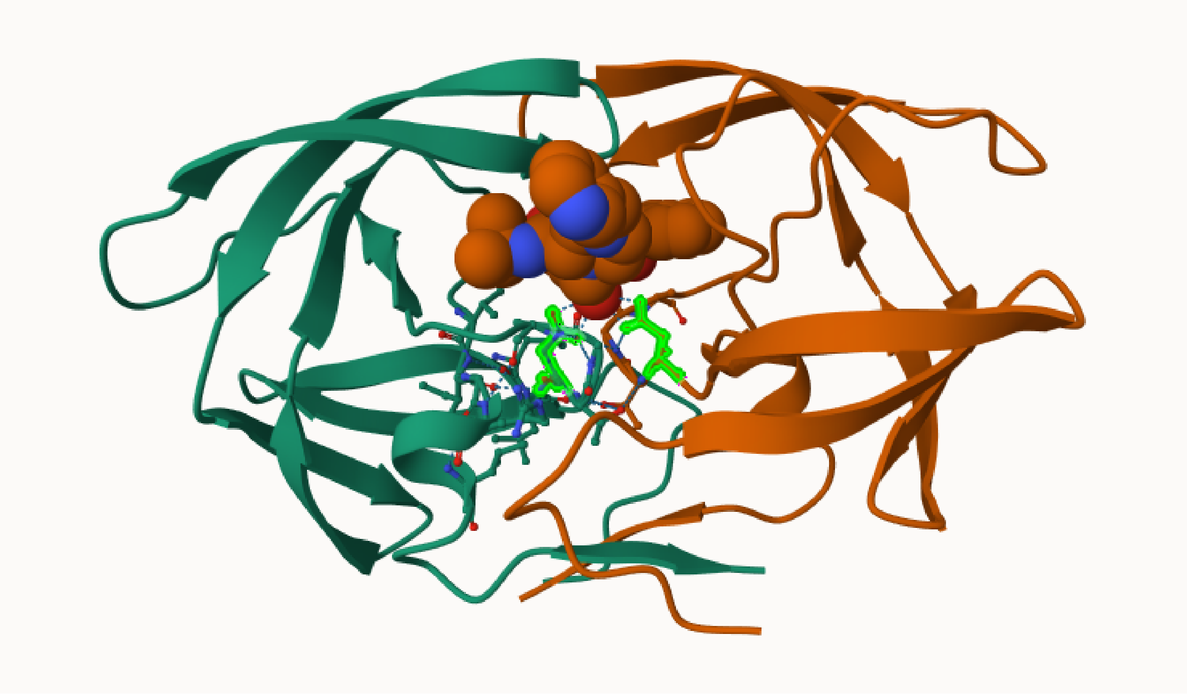


Figure 2: Picture of the two aspartate residues binding to the small molecule

```
library(bio3d)

hiv<-read.pdb("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
```

```
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" 
```

Here we will do a Normal Mode Analysis (NMA) to predict functional 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
```

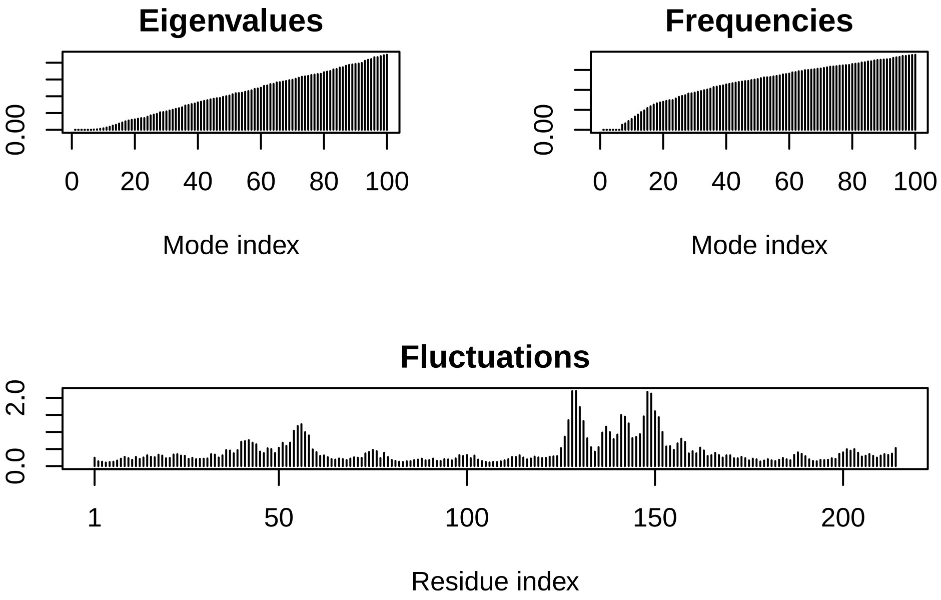
```
adk
```

```

 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
```

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

```
 Building Hessian...        Done in 0.02 seconds.
 Diagonalizing Hessian...   Done in 0.4 seconds.
```

```
plot(m)
```



Make a “movie” called a trajectory of the predicted motions:

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

Then I can open this file in Mol*