Class 9: Structural Bioinformatics

Shubhayan Manjrekar: A17128282

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 or from alternate link: https://tinyurl.com/pdbstats24

Read this into R

```
pdbdb<- read.csv("pdb_stats.csv")
pdbdb
```

```
           Molecular.Type   X.ray     EM    NMR Multiple.methods Neutron Other
1          Protein (only) 167,192 15,572 12,529              208      77    32
2 Protein/Oligosaccharide   9,639  2,635     34                8       2     0
3              Protein/NA   8,730  4,697    286                7       0     0
4     Nucleic acid (only)   2,869    137  1,507               14       3     1
5                   Other     170     10     33                0       0     0
6  Oligosaccharide (only)      11      0      6                1       0     4
    Total
1 195,610
2  12,318
3  13,720
4   4,531
5     213
6      22
```

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

```
                          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
```

and answer the following questions:

```
pdbdb$Total
```

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

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

```
as.numeric(sub(",","", pdbdb$Total ) )
```

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

```
#as.numeric(pdbdb$Total)
```

```
x<- pdbdb$Total
as.numeric
```

```
function (x, ...)  .Primitive("as.double")
```

```
comma2numeric<- function(x) {

 as.numeric(sub(",","", pdbdb$Total ) )
}
```

Test it

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

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

```
apply(pdbdb, 2, comma2numeric)
```

```
      X.ray     EM    NMR Multiple.methods Neutron  Other  Total
[1,] 195610 195610 195610           195610  195610 195610 195610
[2,]  12318  12318  12318            12318   12318  12318  12318
[3,]  13720  13720  13720            13720   13720  13720  13720
[4,]   4531   4531   4531             4531    4531   4531   4531
[5,]    213    213    213              213     213    213    213
[6,]     22     22     22               22      22     22     22
```

Or try a different read/import function”

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

```
Rows: 6 Columns: 8
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): Molecular Type
dbl (3): Multiple methods, Neutron, Other
num (4): X-ray, EM, NMR, Total

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pdbdb$Total
```

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

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

```
#message: false
library(readr)
```

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

```
[1] 83.30359
```

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

```
[1] 10.18091
```

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

```
colnames(pdbdb)
```

```
[1] "Molecular Type"   "X-ray"            "EM"               "NMR"
[5] "Multiple methods" "Neutron"          "Other"            "Total"
```

```
total_structures <- sum(pdbdb$Total, na.rm = TRUE)
protein_structures <- sum(pdbdb$Total[pdbdb$`Molecular Type` %in%
  c("Protein (only)", "Protein/Oligosaccharide", "Protein/NA")], na.rm = TRUE)
proportion_protein <- protein_structures / total_structures
proportion_protein
```

```
[1] 0.9789501
```

> 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?

Five structures of HIV-1

MOl *

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

https://molstar.org

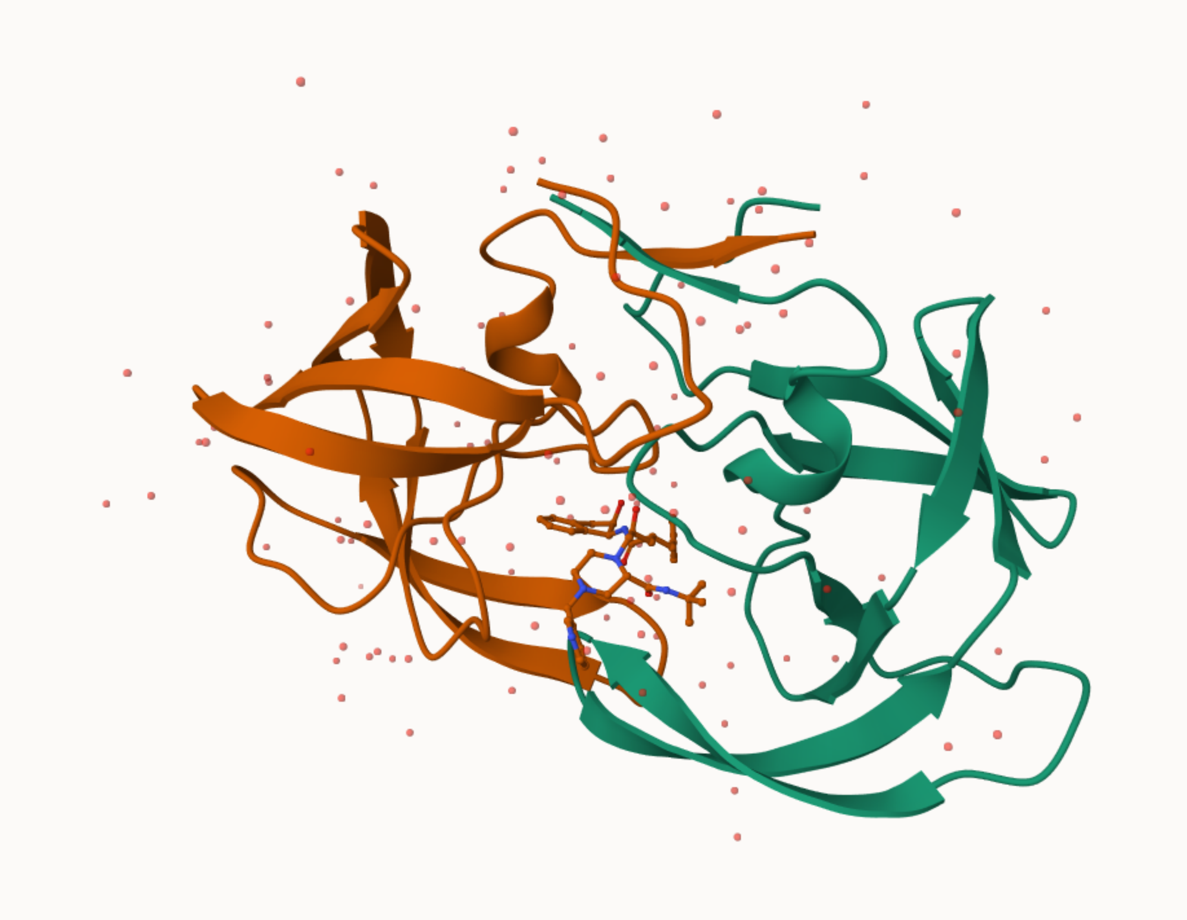


Figure 1: A first image from molstar

some more custom images:

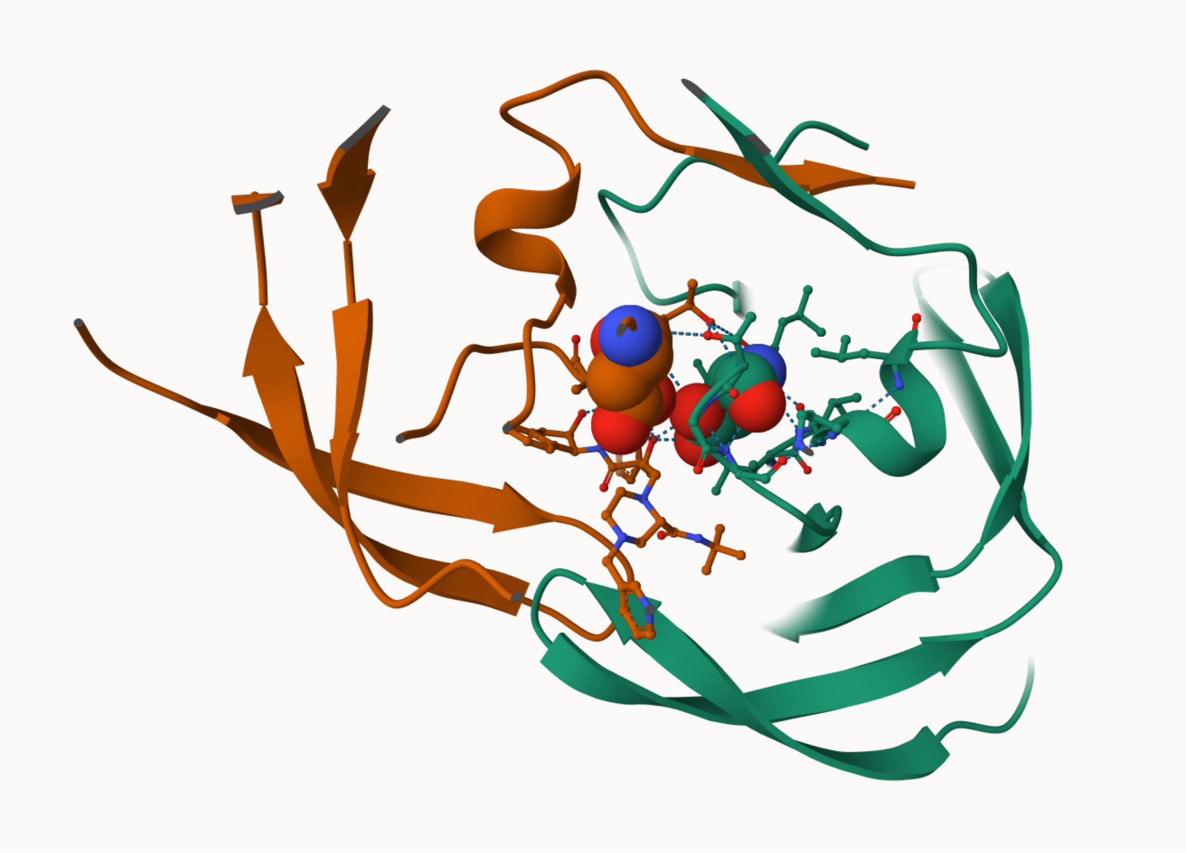


Figure 2: The all important catalytic ASP25 amino acids

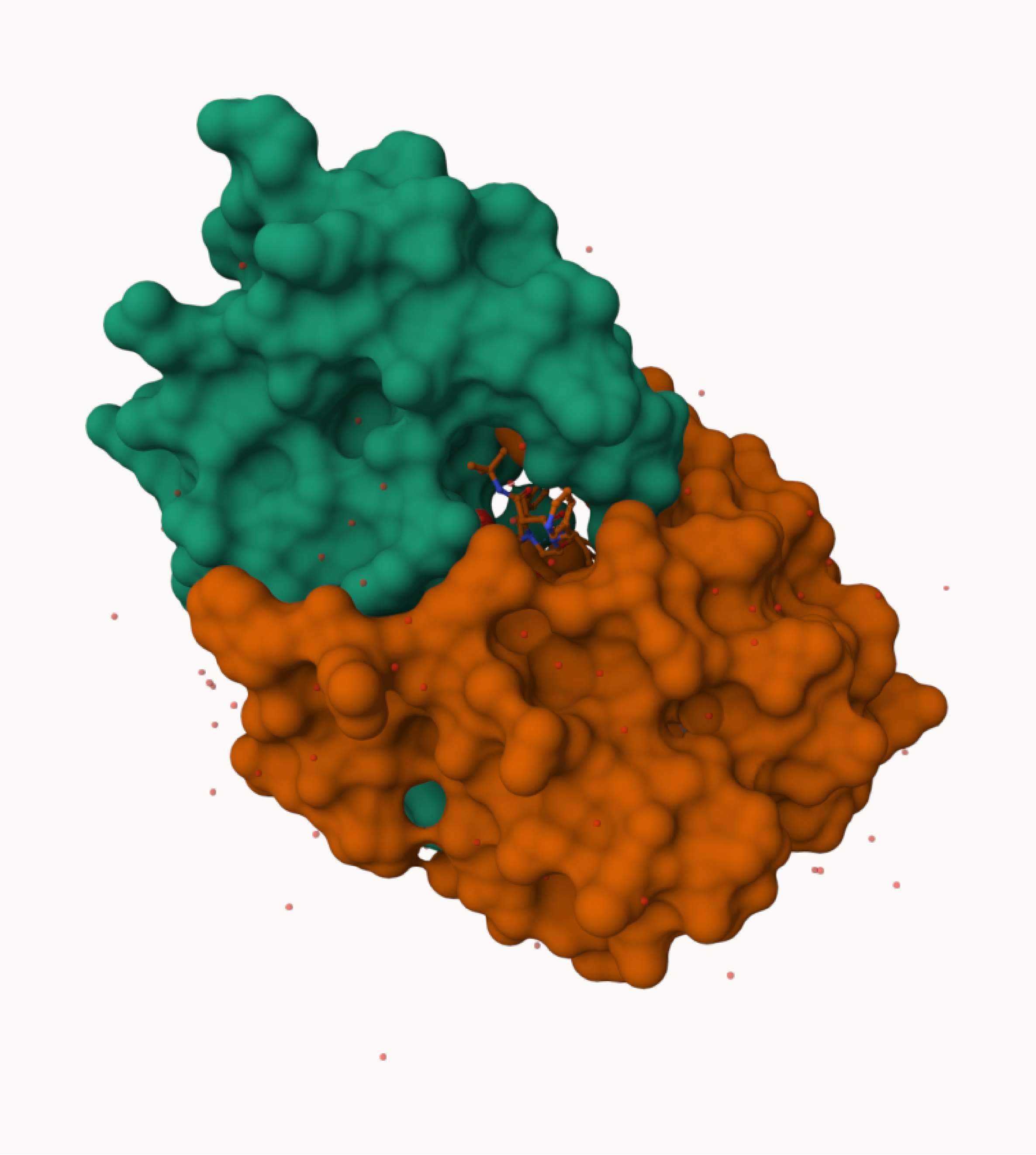


Figure 3: Surface display showing Merk compound in the peptide binding pocket

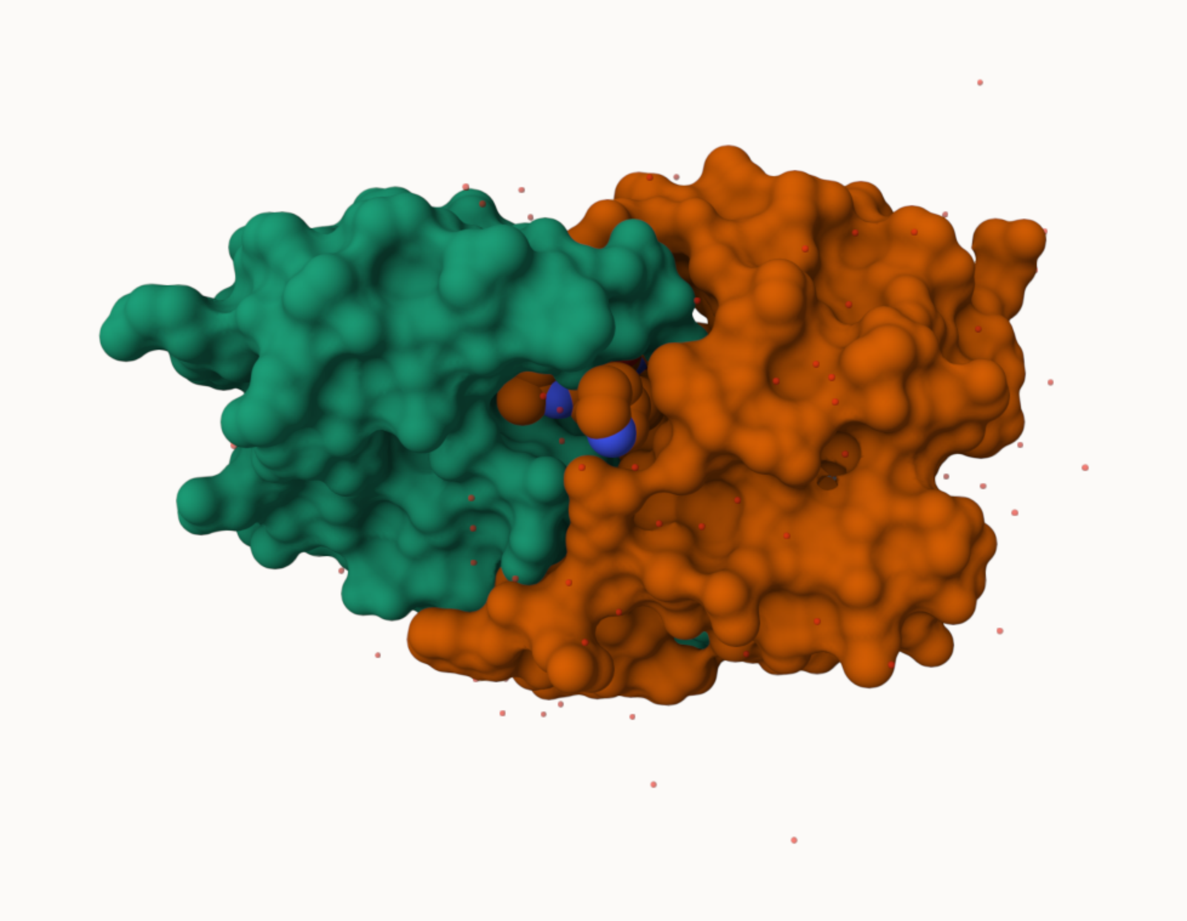


Figure 4: Close up view of bindng site with drug and HOH 308

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 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
```

```r
attributes(pdb)
```

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

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

```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
pdbseq(pdb)
```

```
  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" 
```

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

```
 25 
"D" 
```

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

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

```
[1] 198
```

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

HOH and MK1

> Q9: How many protein chains are in this structure?

2

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

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

Predicing functional motions of a single structure

Let's do bioinfromatics 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
```

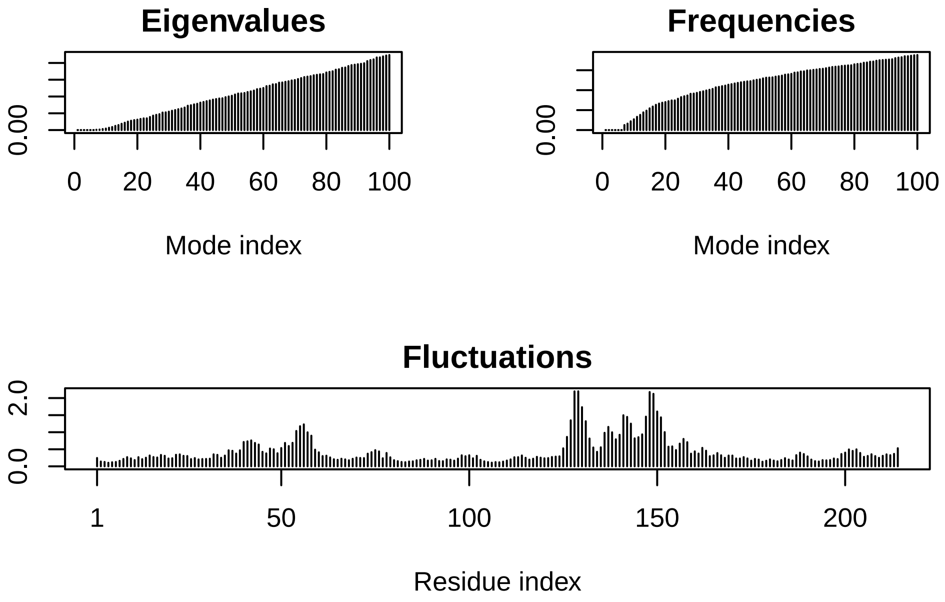
```
adk
```

```

 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
```

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

```
 Building Hessian...        Done in 0.031 seconds.
 Diagonalizing Hessian...   Done in 0.314 seconds.
```

```r
plot(m)
```



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

```r
mktrj(m, file="adk.pdb")
```

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