Class 10: Structural Bioinformatics pt.1

Jessica Gao (PID:A16939806)

Table of contents

1. The PDB database

The main repository of biomolecular structure data is called the PDB found at: https://www.rcsb.org/

Let's see what this database contains.I went to PDB > ANalyze > PDB statistics > By Exp method and molecular type.

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

```
           Molecular.Type   X.ray     EM    NMR Multiple.methods Neutron Other
1          Protein (only) 169,563 16,774 12,578              208      81    32
2 Protein/Oligosaccharide   9,939  2,839     34                8       2     0
3              Protein/NA   8,801  5,062    286                7       0     0
4     Nucleic acid (only)   2,890    151  1,521               14       3     1
5                   Other     170     10     33                0       0     0
6  Oligosaccharide (only)      11      0      6                1       0     4
    Total
1 199,236
2  12,822
3  14,156
4   4,580
```

```
5     213
6      22
```

```
pdb_file$X.ray
```

```
[1] "169,563" "9,939"   "8,801"   "2,890"   "170"     "11"
```

means character, can't do math with character it's underneath each column name

Get rid of the commas and change things to numeric. OR Do a different read csv function.

The comma in these numbers is causing them to be read as character rather than numeric. I can fix this by "," for nothing " " with the sub() function:

```
x <- pdb_file$X.ray
sum(as.numeric(sub(",","",x)))
```

```
[1] 191374
```

Or I can use the **readr** package and the 'read_csv()' function.

```
library(readr)
pdbstats <- read_csv("Data Export Summary.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.
```

```
pdbstats
```

```
# A tibble: 6 x 8
  `Molecular Type`   `X-ray`    EM   NMR `Multiple methods` Neutron Other  Total
  <chr>                <dbl> <dbl> <dbl>              <dbl>   <dbl> <dbl>  <dbl>
1 Protein (only)      169563 16774 12578                208      81    32 199236
```

```
2 Protein/Oligosacc~    9939  2839    34                 8       2     0  12822
3 Protein/NA            8801  5062   286                 7       0     0  14156
4 Nucleic acid (onl~    2890   151  1521                14       3     1   4580
5 Other                  170    10    33                 0       0     0    213
6 Oligosaccharide (~      11     0     6                 1       0     4     22
```

I want to clean the column names so they are all lower case and don't have spaces in them.

```
colnames(pdbstats)
```

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

```
library(janitor)
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
df <- clean_names(pdbstats)
df
```

```
# A tibble: 6 x 8
  molecular_type         x_ray    em   nmr multiple_methods neutron other  total
  <chr>                  <dbl> <dbl> <dbl>            <dbl>   <dbl> <dbl>  <dbl>
1 Protein (only)        169563 16774 12578              208      81    32 199236
2 Protein/Oligosacchar~   9939  2839    34                8       2     0  12822
3 Protein/NA              8801  5062   286                7       0     0  14156
4 Nucleic acid (only)     2890   151  1521               14       3     1   4580
5 Other                    170    10    33                0       0     0    213
6 Oligosaccharide (onl~     11     0     6                1       0     4     22
```

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

Total number of X-ray structures

```
x_ray_sum <- sum(df$x_ray)
```

Total number of structures

```
total_struc <- sum(df$total)
```

Percentage solved by X-Ray

```
percen.x_ray <- x_ray_sum/total_struc*100
percen.x_ray
```

```
[1] 82.83549
```

82.8% of the structures are solved by X-ray

Total number of EM structures

```
em_sum <-sum(df$em)
```

Percentage solved by EM

```
percen.em <- em_sum/total_struc*100
percen.em
```

```
[1] 10.75017
```

10.8% of the structures are solved through Electron Microscopy.

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

```
protein_only <- df[1, "total"]
protein_only
```

```
# A tibble: 1 x 1
   total
   <dbl>
1 199236
```

```
total_str <- sum(df$total)
total_str
```

```
[1] 231029
```

```
prop <- protein_only/total_str*100
prop
```

```
     total
1 86.23852
```

86.24 percent of the structures are protein only.

> 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?

There are 231,029 HIV-1 protease structure in the PDB website.

2. Using Mol* Visualizing the HIV-1 protease structure

The main Mol* homepage: We can input our pdb files or just give it a PDB database accession code (4 letter PDB code).

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

It is showing each individual water molecule as a whole instead of individual atoms, each water molecule is represented by a sphere.

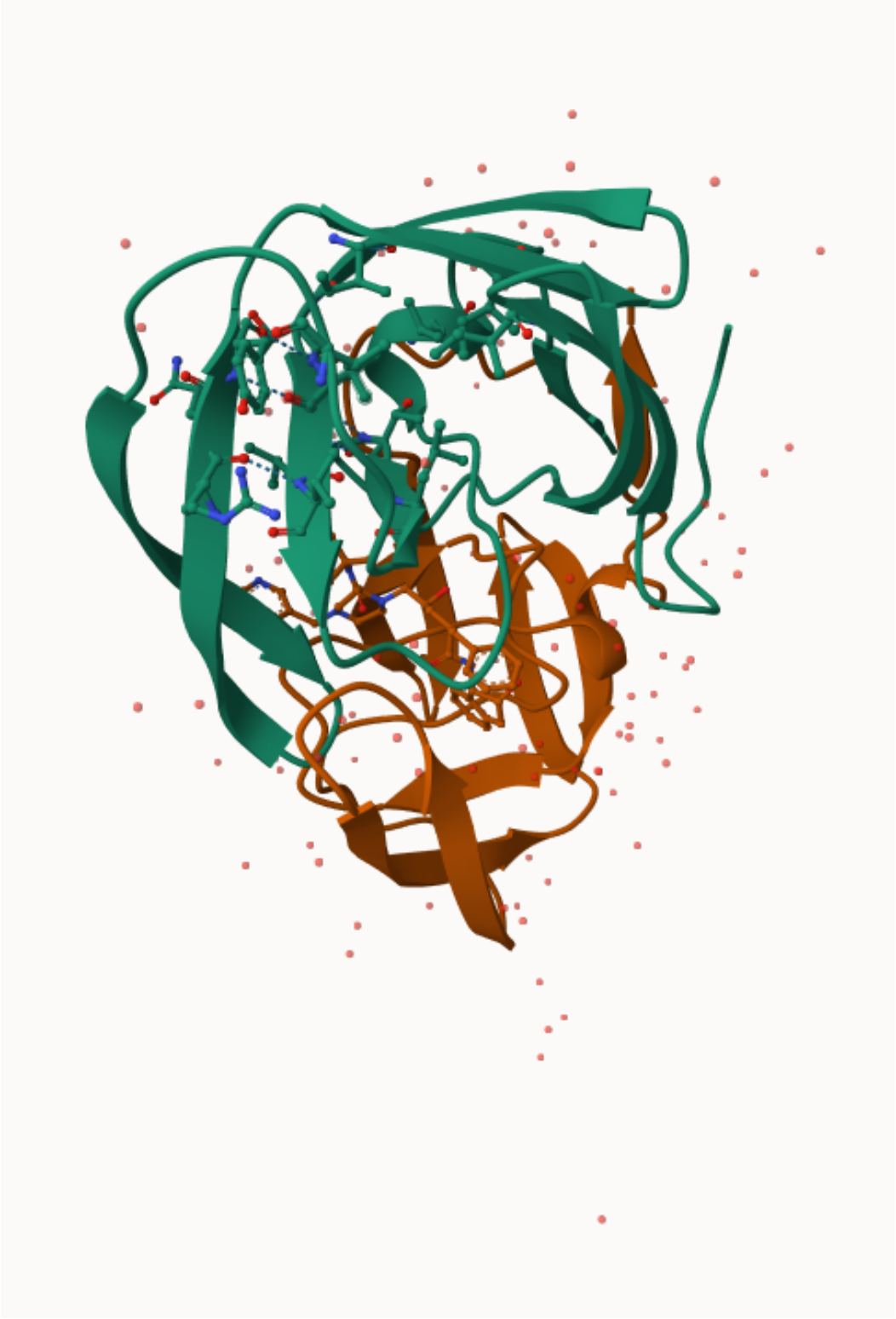


Figure 1: Molecular view of 1HSG

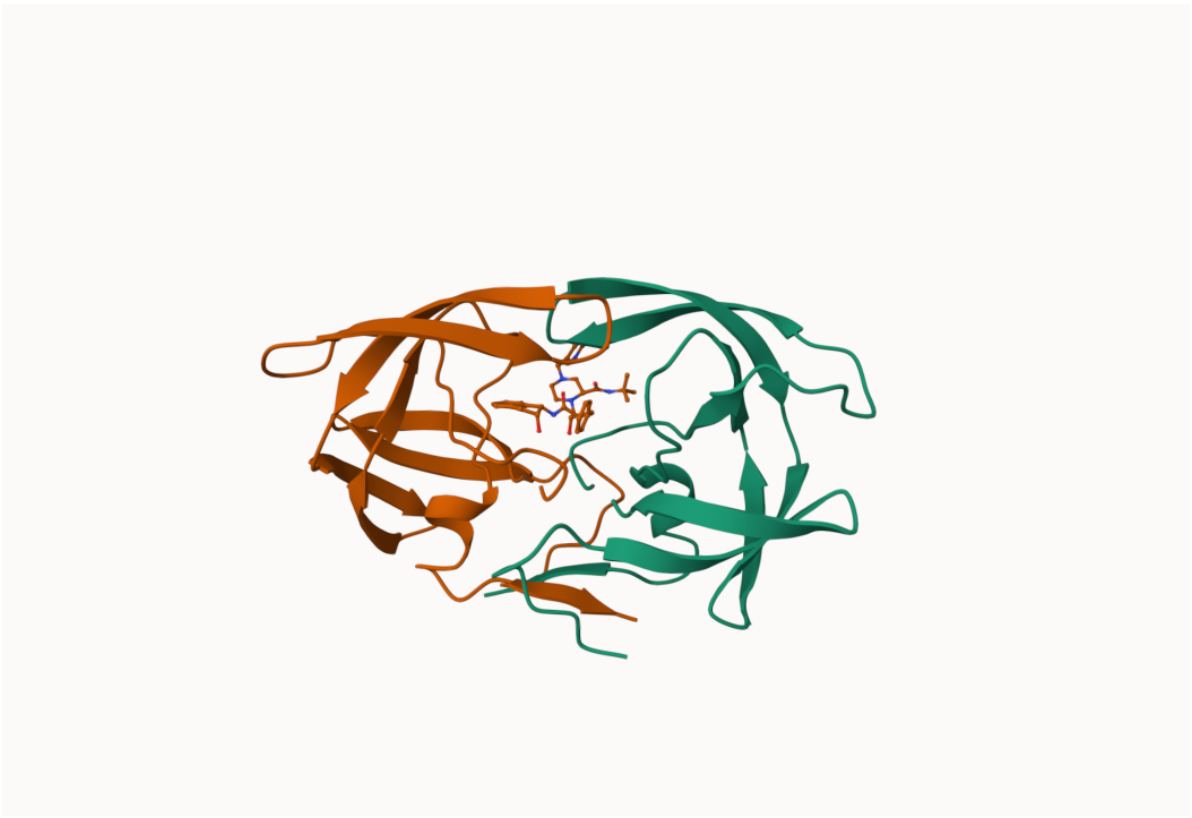
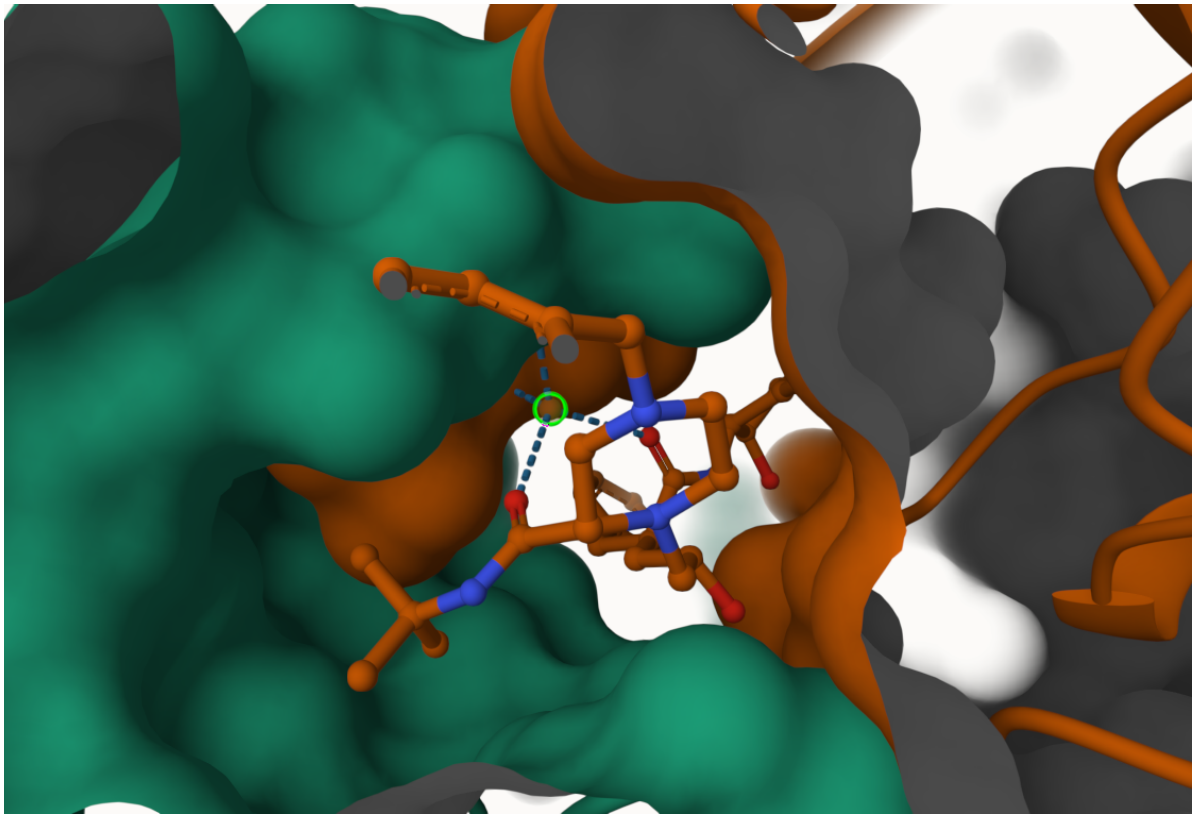


Figure 2: 1HSP cartoon structure

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



find residue number: 308 for water molecule

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 3: 1HSP Chain A and B Aspartate

Figure 4: 1HSP Chain A and B Aspartate and Critical Water

3. Introduction to Bio3D in R

We can use the **bio3d** package for structural bioinformatics to read PDB data into R

```
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
```

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

```
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"
```

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

```
[1] 198
```

There are 198 amino acid residues.

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

HOH [127], MK1 [1]

Q9:How many protein chains are in this structure?

2 chains A and B

Looking at the 'pdb' object in more detail

```
attributes(pdb)
```

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

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

```
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>
```

Let's try a new function not yeat in the bio3d package. It requires **r3dmol** package that we need to install with 'install.packages("r3dmol")' and 'install.pachakges("shiny").

```
library(r3dmol)
source("https://tinyurl.com/viewpdb")
#view.pdb(pdb, backgroundColor="pink")
```

4. Predicting functional motions dynamics

We can use the 'nma()' function in bio3d to predict the large-scale functional motions of biomolecules.

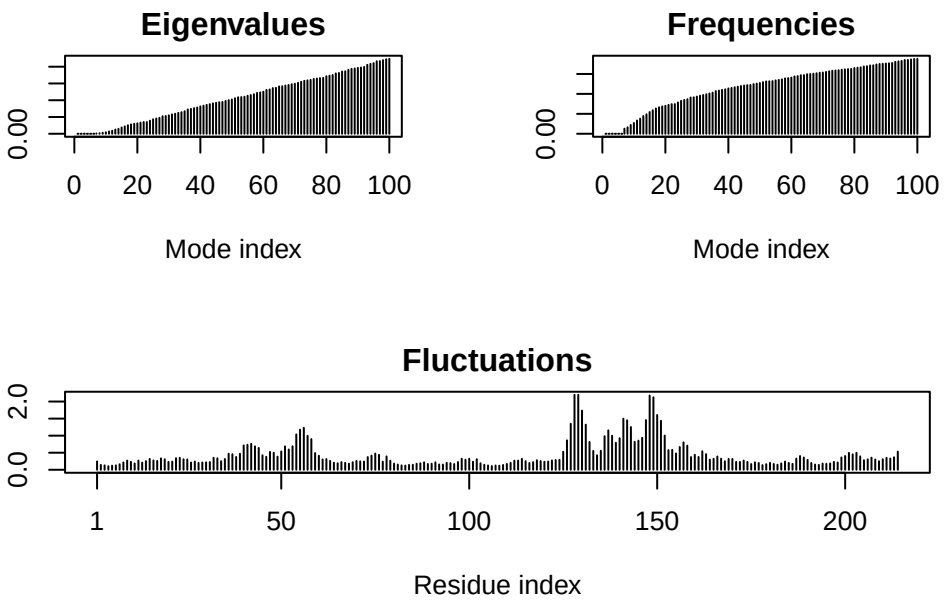
```
adk <- read.pdb("6s36")
```

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

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

```
 Building Hessian...        Done in 0.02 seconds.
 Diagonalizing Hessian...   Done in 0.3 seconds.
```

```
plot(m)
```



Write out a trajectory of the predicted molecular motion:

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

load file adk_m7.pdb into Mol*