Class10: Structural Bioinformatics

Andres Vasquez (PID: 16278181)

What is in the PDB database

The repository of biomolecular structure info is the PDB < www.rscb.org >.

Let's see what this database contains:

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

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

```
#sum(stats$X.ray)
```

```
as.numeric(stats$X.ray)
```

```
Warning: NAs introduced by coercion
```

```
[1]  NA  NA  NA  NA 164  11
```

The commas are affecting the result.

We got to get rid of the commas. Can you find a function to get rid of the commas?

```
x <- stats$X.ray
sum(as.numeric(gsub(",","",x)))
```

```
[1] 184362
```

I am going to turn this into a function and then use `apply()` to work on the entire table of data

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

sumcomma(stats$X.ray)
```

```
[1] 184362
```

```
n.total <- sumcomma(stats$Total)
n.total
```

```
[1] 219140
```

```
apply(stats, 2, sumcomma)
```

```
           X.ray               EM              NMR Multiple.methods 
          184362            20191            14237              234 
         Neutron            Other            Total 
              79               37           219140 
```

```
apply(stats, 2, sumcomma)/n.total
```

```
           X.ray               EM              NMR Multiple.methods 
    0.8412978005     0.0921374464     0.0649676006     0.0010678105 
         Neutron            Other            Total 
    0.0003605001     0.0001688418     1.0000000000 
```

```
sumcomma(stats$EM)
```

```
[1] 20191
```

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

```
head(stats)
```

```
                          X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)          163,468 13,582 12,390              204      74    32
Protein/Oligosaccharide   9,437  2,287     34                8       2     0
Protein/NA                8,482  4,181    286                7       0     0
Nucleic acid (only)       2,800    132  1,488               14       3     1
Other                       164      9     33                0       0     0
Oligosaccharide (only)       11      0      6                1       0     4
                          Total
Protein (only)          189,750
Protein/Oligosaccharide  11,768
Protein/NA               12,956
Nucleic acid (only)       4,438
Other                       206
Oligosaccharide (only)       22
```

```
as.numeric(gsub(",","", stats[1, "Total"]))
```

```
[1] 189750
```

> 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 248,805,733 entries which compared to PDB protein entries (189,750) means there are only ~7% of known sequences with a known structure.

248,805,733 - 189,750

```
189750/248805733 *100
```

```
[1] 0.07626432
```

Visualizing the HIV-1 protease structure

Mol* ("mol-star") viewer is now everywhere. The homepage is here: https://molstar.org/viewer/

I want to insert my image from Mol* here.

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

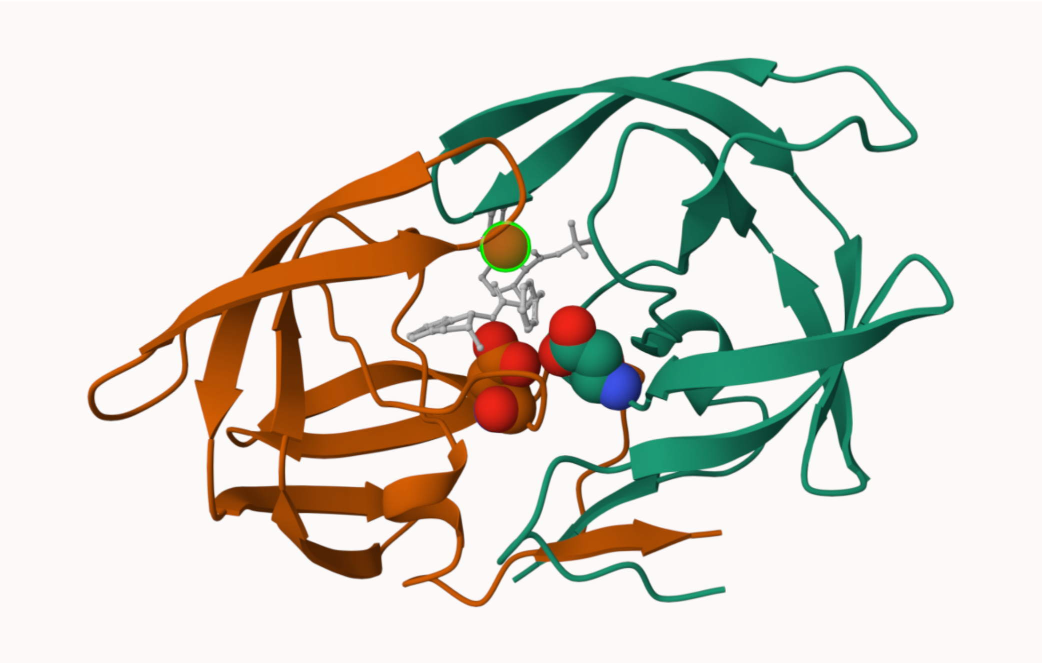
We only see one atom per water molecule because Hydrogen cannot be detected

> 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 critical "conserved" water molecule is identified as "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.



Introduction to Bio3D in R

```r
library(bio3d)
```

```r
pdb <- read.pdb("1hsg")
```

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

```r
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?

There are 198 amino acid residues

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

HOH (127) and MK1 (1)

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

There are 2 protein chains

Note that the attributes `(+ attr:)` of this object are listed on the last couple of lines. To find the attributes of any such object you can use:

```
attributes(pdb)
```

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

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

To access these individual attributes we use the `dollar-attribute` name convention that is common with R list objects. For example, to access the `atom` attribute or component use `pdb$atom`:

```
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>
```

Predicting functional motions of a single structure

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

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

```r
adk
```

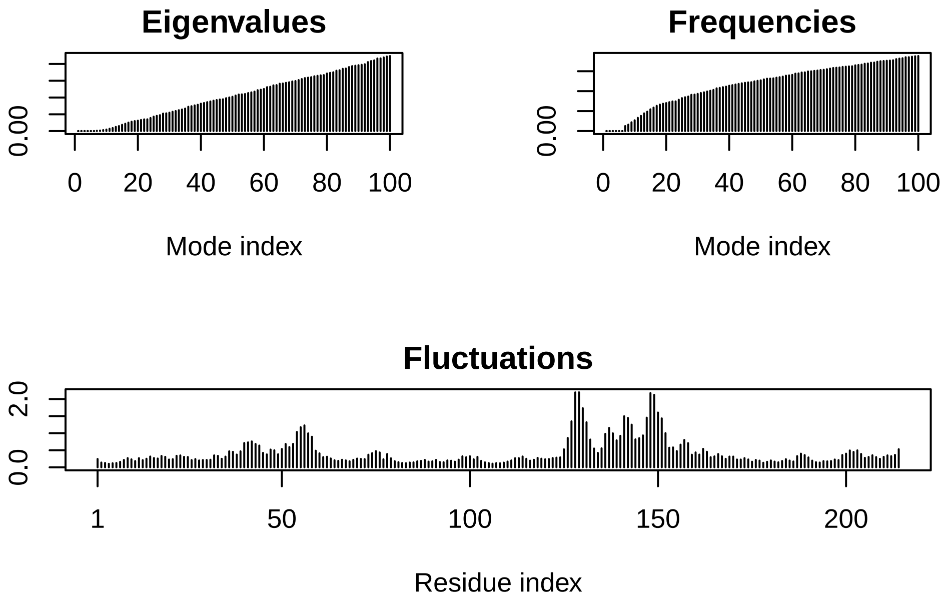
```

 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
```

Normal mode analysis (NMA) is a structural bioinformatics method to predict protein flexibility and potential functional motions (a.k.a. conformational changes).

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

```
 Building Hessian...        Done in 0.042 seconds.
 Diagonalizing Hessian...   Done in 0.529 seconds.
```

```
plot(m)
```



To view a “movie” of these predicted motions we can generate a molecular “trajectory” with the mktrj() function.

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