Class 10: Structural Bioinformatics Pt.1

Cynthia Perez (A16393492)

PDB Statistics

The main repository of biomolecular structure info is the PDB <www.rcsb.org>. Let's see what the database contains. Go to "Analyze" > "PDB Statistics" > "by Experimental Method and Molecular Type". Read the csv file:

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

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

	Total
Protein (only)	186,898
Protein/Oligosaccharide	11,559
Protein/NA	12,621
Nucleic acid (only)	4,378
Other	206
Oligosaccharide (only)	22

We have to get rid of the commas in order to convert the vector from characters to numeric.

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

```
[1] 182348
```

We have a working snippet of code that can be turned into a function to work on more variables from the table of data.

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

Use the `apply()` function to use the `sumcomma()` on the entire dataset.

```
#apply the sumcomm() function to the stats dataset over columns
apply(stats, 2, sumcomma)
```

```
          X.ray              EM             NMR Multiple.methods
         182348           18817           14173             230
        Neutron           Other           Total
             79              37          215684
```

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

```
#take the sum of the total to divide the sum of X.ray and EM by
n.total <- sumcomma(stats$Total)
#percentage of x.ray
(sumcomma(stats$X.ray)/n.total)*100
```

```
[1] 84.54406
```

```
#percentage of EM
(sumcomma(stats$EM)/n.total)*100
```

```
[1] 8.724337
```

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

```
sumcomma(stats["Protein (only)",])/n.total
```

```
[1] 1.733072
```

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?

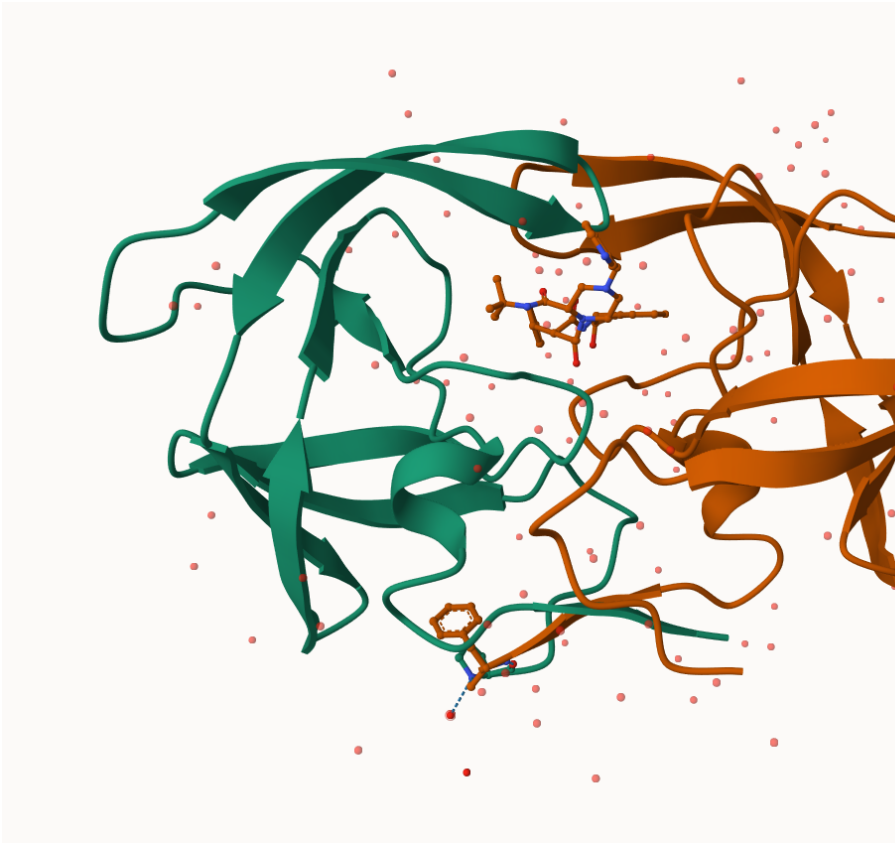
In UniProt there are 248,805,744 entries which compared to PDB protein entries 186,898 means there are only ~7% of known sequences with a known structure

```
186898/248805744 *100
```

```
[1] 0.07511804
```

Visualizing the HIV-1 protease Structure

Mol*("mol-star") viewer is now everywhere. The homepage is 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 need a better resolution. Hydrogens are too small to be seen so we only see the oxygen atom

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

Residue number 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.

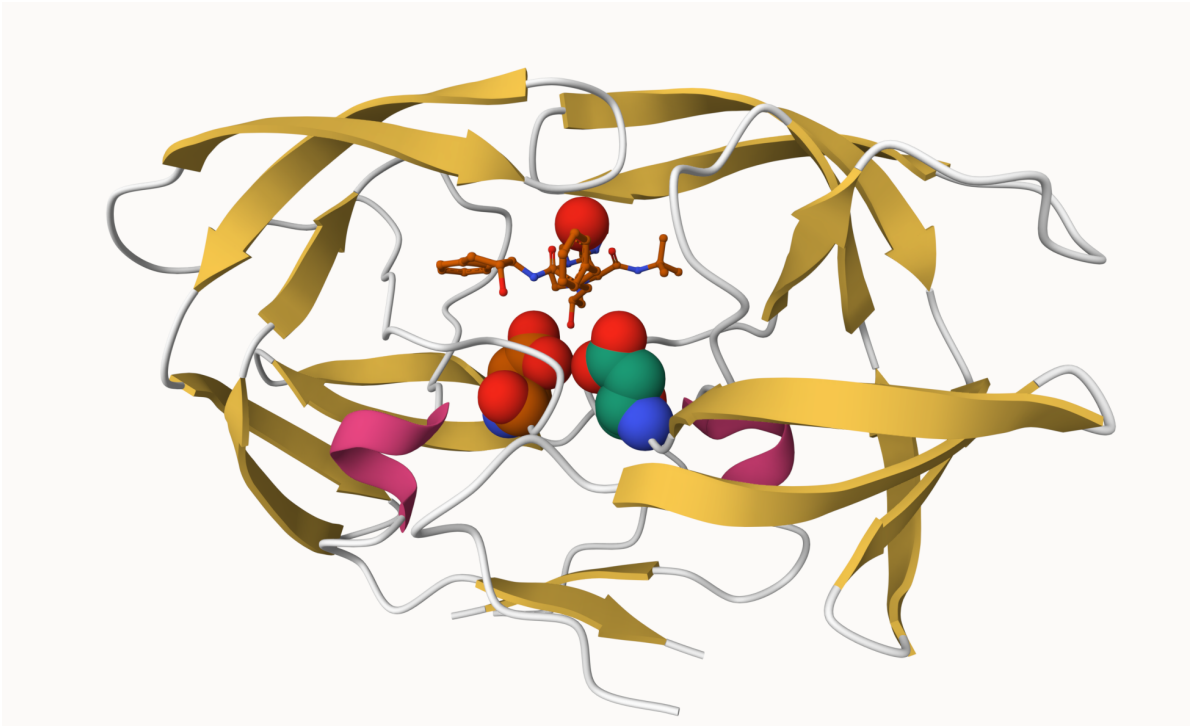


Figure 1: 1HSG structure with highlighted water and ASP 25 residues

Introduction to Bio3D in 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
```

```
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>
```

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

```
 25
"D"
```

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

198 amino acid residues

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

HOH, MK1 > Q9: How many protein chains are in this structure?

2 A and B

Predicting Funcitonal Motions of a single strucutre

We can do a bioinformatics prediction of functional motions (felxibility/dynamics)

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

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

```
pdb
```

```

 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:
```

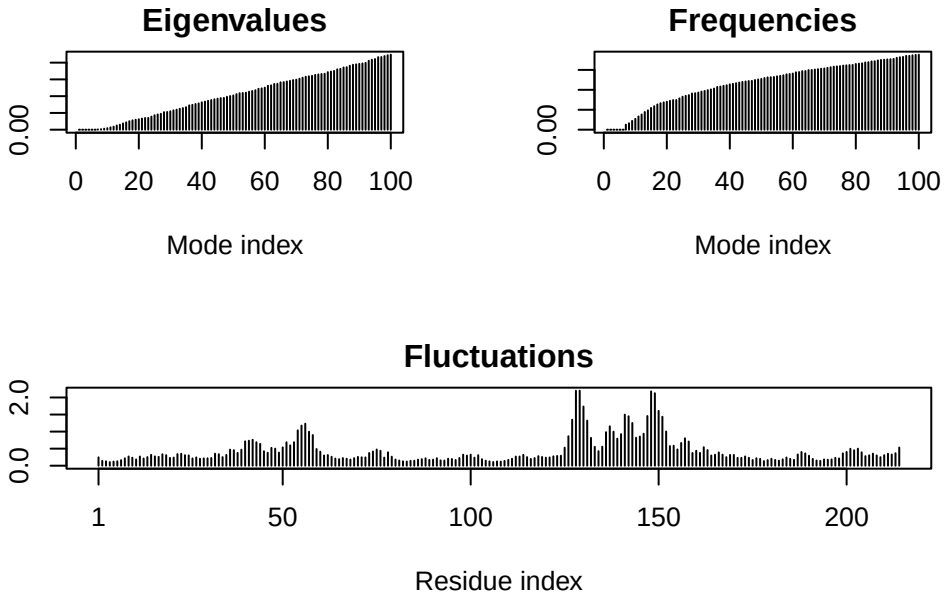
```
      MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
      DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDKI
      VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
      YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG

+ attr: atom, xyz, seqres, helix, sheet,
        calpha, remark, call
```

```
m <- nma(pdb)
```

```
 Building Hessian...        Done in 0.044 seconds.
 Diagonalizing Hessian...   Done in 0.487 seconds.
```

```
plot(m)
```




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

We download this file into Mol* to create an animation of the structure.