Class 10: Structural Bioinformatics Pt.1

Justin Lu (A16318305)

What is in the PDB database

The main repository of biomolecular strucutre info is the PDB <www.rcsb.org>.

Let's see what this database contains:

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

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

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

Values are characters instead of integers/numbers

```
as.numeric(c(10,"1,000"))
```

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

```
[1] 10 NA
```

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

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

```
[1] 184362
```

I am going to turn this into a function 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
```

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

```
[1] 20191
```

```
apply(stats,2,sumcomma)/ sumcomma(stats$Total)
```

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

84.1 % are solved by X-ray and 9.2% are solved by EM.

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

```
189750/n.total
```

```
[1] 0.8658848
```

86.6% of the structures in the PDB are proteins

> 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?

```
186898 / 248805733 * 100
```

```
[1] 0.07511804
```

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.

Working with the bio3d package

```
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)
```

Figure 1: My first molecular image

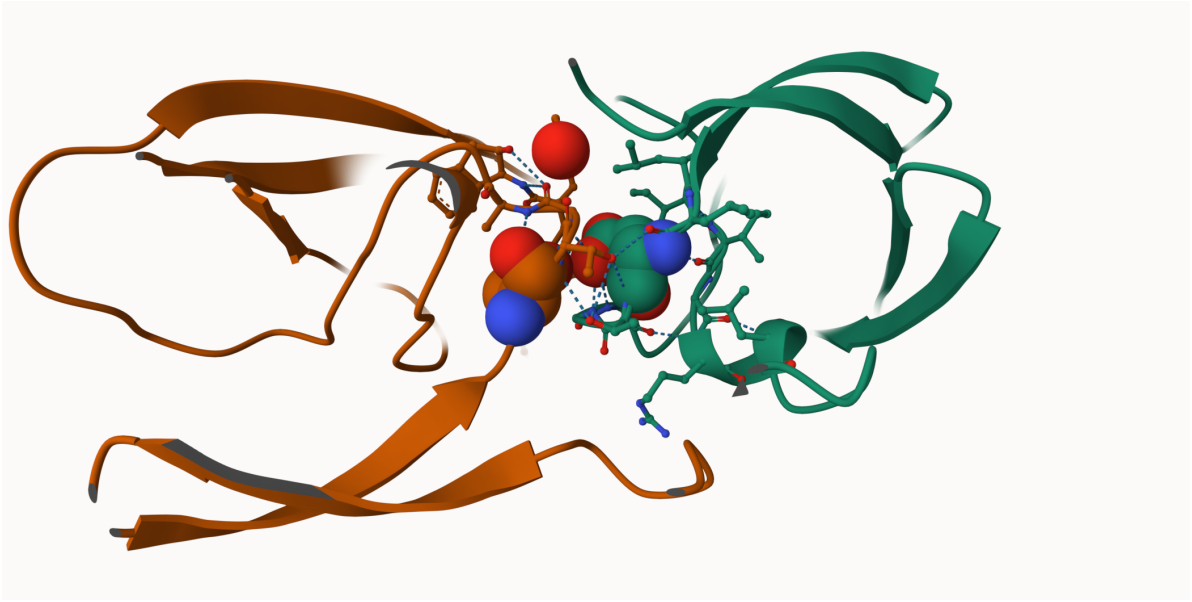


Figure 2: My second molecular image with Aspartic Acid and Water highlighted

```
     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"
```

Predicting functional motions of a single structure

We can do a bioinformatics prediction of functional motions (i.e. flexibility/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
```

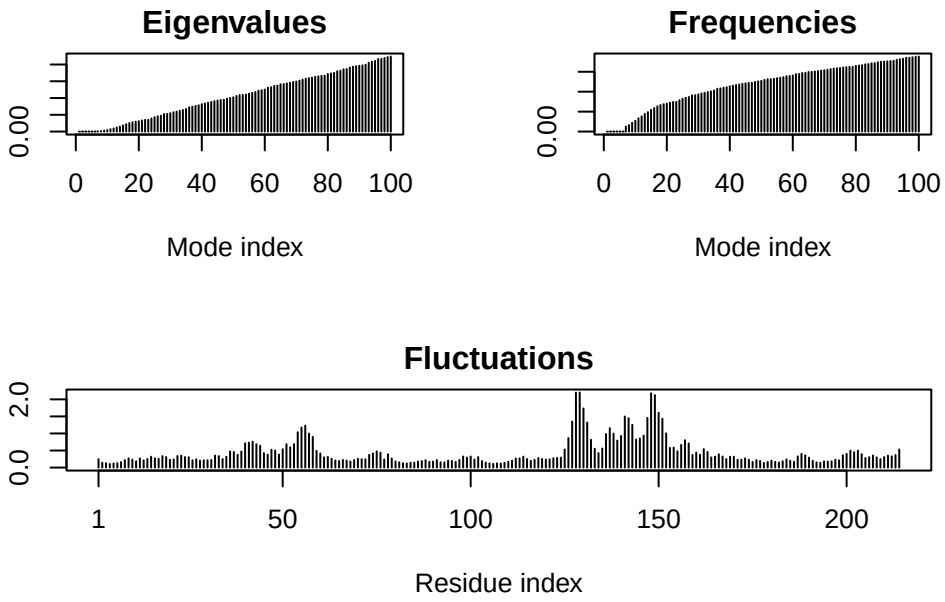
```
      DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDKI
      VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
      YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG

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

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

```
 Building Hessian...        Done in 0.021 seconds.
 Diagonalizing Hessian...   Done in 0.453 seconds.
```

```
plot(m)
```




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