BIMM143class10

The PDB Database

First let's see what is in the PDB database- the main repository of protein structures

Downloaded composition stats from:

https://tinyurl.com/statspdb

```
##remmeber you gotta put the downloaded file
##in the same location as the R
stats<-read.csv("PDBStats.csv", row.names=1)
stats
```

```
                         X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)         158,844 11,759 12,296              197      73    32
Protein/Oligosaccharide  9,260  2,054     34                8       1     0
Protein/NA               8,307  3,667    284                7       0     0
Nucleic acid (only)      2,730    113  1,467               13       3     1
Other                      164      9     32                0       0     0
Oligosaccharide (only)      11      0      6                1       0     4
                         Total
Protein (only)         183,201
Protein/Oligosaccharide 11,357
Protein/NA              12,265
Nucleic acid (only)      4,327
Other                      205
Oligosaccharide (only)      22
```

```
##There is a problem here due to the commas n #
##This causes R to treat them as characters
```

```
x<-stats$X.ray
##gsub to replace
##first is the comma then none to replace
##and of course x
gsub(",", "", x)
```

```
[1] "158844" "9260"   "8307"   "2730"   "164"    "11"
```

```
##as.numeric to change to numeric #
as.numeric(gsub(",", "", x))
```

```
[1] 158844   9260   8307   2730    164     11
```

```
rm.comma<-function(x){
  as.numeric(gsub(",", "", x))
}

rm.comma(stats$EM)
```

```
[1] 11759  2054  3667   113     9     0
```

```
##I can use 'apply()' to fix the whole table
##2 for apply to column, second
##rm.comma for function as argument third
##stats for the x or array matrix first
pdbstats<-apply(stats, 2, rm.comma)
rownames(pdbstats)<-rownames(stats)
head(pdbstats)
```

```
                        X.ray    EM   NMR Multiple.methods Neutron Other
Protein (only)         158844 11759 12296               197      73    32
Protein/Oligosaccharide  9260  2054    34                 8       1     0
Protein/NA               8307  3667   284                 7       0     0
Nucleic acid (only)      2730   113  1467                13       3     1
Other                     164     9    32                 0       0     0
Oligosaccharide (only)     11     0     6                 1       0     4
                        Total
```

```
Protein (only)          183201
Protein/Oligosaccharide  11357
Protein/NA               12265
Nucleic acid (only)       4327
Other                      205
Oligosaccharide (only)      22
```

```
##now we want to find the total
## so we do sum
totals<-apply(pdbstats, 2, sum)
##since totals alone just give you numbers
##you need to make it into percentage
##use round to do so
##totals/totals alone would be wrong
##totals/totals["Total] shows
##you are interacting with the total number
##of occurance so they give you the right
##% value. 2 is the decimal place
round(totals/totals["Total"]*100, 2)
```

```
        X.ray             EM            NMR Multiple.methods
        84.83           8.33           6.68           0.11
      Neutron          Other          Total
         0.04           0.02         100.00
```

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

84.83 and 8.33 respectively

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

```
round(pdbstats[,"Total"]/sum(pdbstats[,"Total"])*100,2)
```

```
      Protein (only) Protein/Oligosaccharide             Protein/NA
               86.67                    5.37                   5.80
  Nucleic acid (only)                   Other Oligosaccharide (only)
               2.05                    0.10                   0.01
```

```
##Q.3 skipped
```

```
library(readr)
read_csv("PDBstats.csv")
```

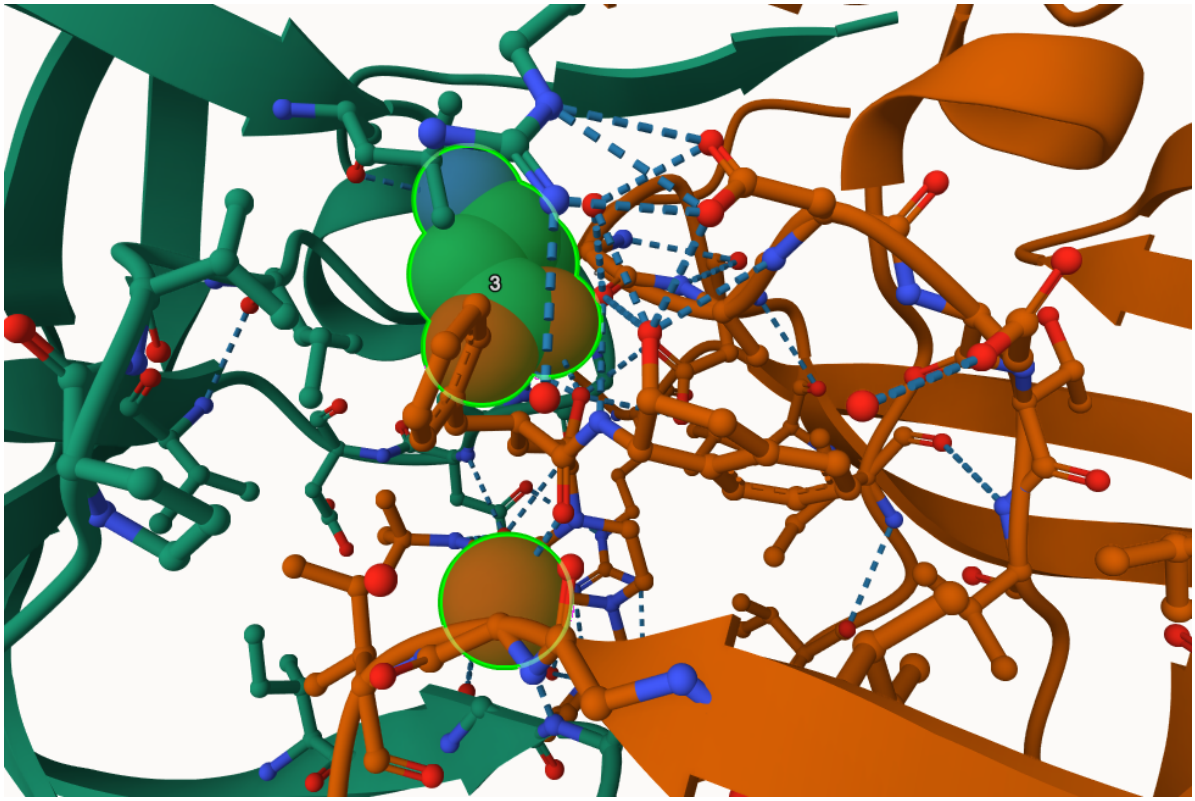
```
Rows: 6 Columns: 8
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): Molecular Type
dbl (3): Multiple methods, Neutron, Other
num (4): X-ray, EM, NMR, Total

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.
```

```
# 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)      158844 11759 12296                197      73    32 183201
2 Protein/Oligosacc~    9260  2054    34                  8       1     0  11357
3 Protein/NA            8307  3667   284                  7       0     0  12265
4 Nucleic acid (onl~    2730   113  1467                 13       3     1   4327
5 Other                  164     9    32                  0       0     0    205
6 Oligosaccharide (~      11     0     6                  1       0     4     22
```

```
##Fraction of Uniprot

##Protein structures in PDB as a fraction of
##Uniprot sequences

round((pdbstats[1,"Total"]/251600768)*100,2)
```

```
[1] 0.07
```

Here is a lovely figure of HIP-Pr with the catalytic ASP residues, the MK1 compounds and the all important water 308. QUestion 6



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

Too small resolution of 2 armstrong is bigger than the hydrogen of water. You need 1 armstrom or better to see such small atoms.

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?

Water HOH 308

##The bio3d package for structural bioinformatics

```
library(bio3d)
pdb<-read.pdb("1HSG")
```

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

```
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>
```

```
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
```

##Predicting functional motions of a single structure

Let's finish today with a bioinformatics calculation to predict the functional motion of a PDB structure.

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

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

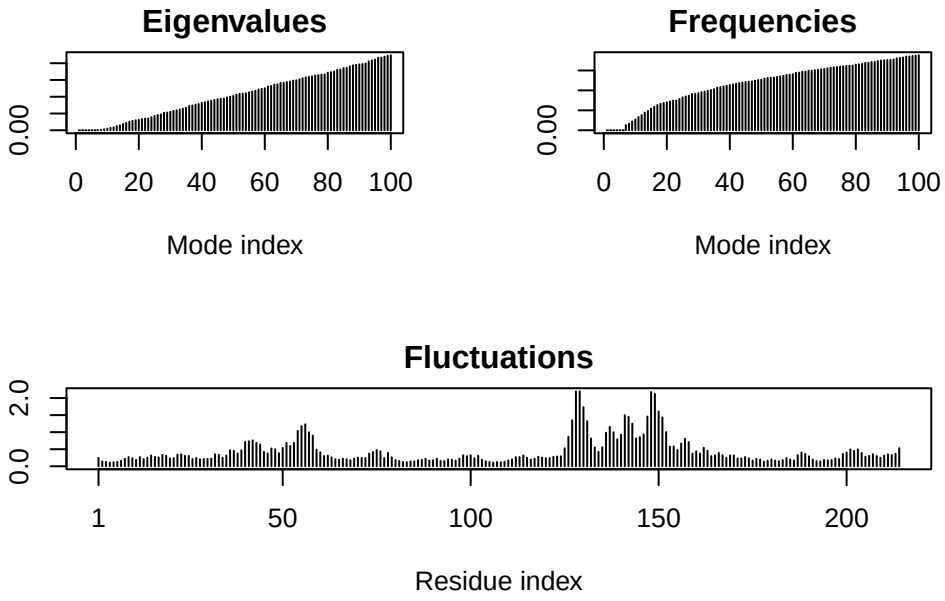
```
adk
```

```
 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
```

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

```
Building Hessian...         Done in 0.04 seconds.
Diagonalizing Hessian...    Done in 0.63 seconds.
```

```
plot(m)
```




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