nov_1_class10

xiaowen xu

what is in the database anyway?

I grabed summary data from https://www.rcsb.org/stats/summary

```
fna.data <- "/Users/xiaowen/Downloads/Data Export Summary.csv"
pdbstats <- read.csv(fna.data, row.names=1)
pdbstats
```

```
                          X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)          167,317 15,698 12,534               208      77    32
Protein/Oligosaccharide   9,645  2,639     34                 8       2     0
Protein/NA                8,735  4,718    286                 7       0     0
Nucleic acid (only)       2,869    138  1,507                14       3     1
Other                       170     10     33                 0       0     0
Oligosaccharide (only)       11      0      6                 1       0     4
                          Total
Protein (only)          195,866
Protein/Oligosaccharide  12,328
Protein/NA               13,746
Nucleic acid (only)       4,532
Other                       213
Oligosaccharide (only)       22
```

-

```
x <- pdbstats$Total
x
```

```
[1] "195,866" "12,328"  "13,746"  "4,532"   "213"     "22"
```

```
as.numeric(gsub(",", "", x))
```

```
[1] 195866  12328  13746   4532    213     22
```

```
convert_comma_numbers <-function(x){
  x <-gsub(',','',x)
  x<-as.numeric(x)
  return(x)}
```

```
n.total <- sum(convert_comma_numbers(pdbstats$Total))
n.total
```

```
[1] 226707
```

- **Q1:** What percentage of structures in the PDB are solved by X-Ray and Electron Microscopy. X-Ray: 83.26%, EM:10.23%

  the apply() function is very useful as it can take any function and apply it over either the rows or cols of a data.frame

```
colSums(apply(pdbstats,2,convert_comma_numbers))/n.total
```

```
          X.ray              EM             NMR Multiple.methods
   0.8325592064    0.1023479646    0.0635181093    0.0010498132
        Neutron           Other           Total
   0.0003617003    0.0001632063    1.0000000000
```

```
library(readr)
fna.data <- "/Users/xiaowen/Downloads/Data Export Summary.csv"
pdb <- read_csv(fna.data)
```

```
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.
```

```
n.xray <-sum(convert_comma_numbers(pdbstats$X.ray))
n.em<-sum(convert_comma_numbers(pdbstats$EM))
```

```
n.xray/n.total *100
```

```
[1] 83.25592
```

```
n.em/n.total *100
```

```
[1] 10.2348
```

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

  ```
  n.protein <-convert_comma_numbers(pdbstats["Protein (only)", "Total"])
  n.protein/n.total *100
  ```

  ```
  [1] 86.3961
  ```

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

It is only showing the oxygen atom. because H is too small to be seen at this resolution.

- **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

yes. H308

- **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.

Discussion Topic: Can you think of a way in which indinavir, or even larger ligands and substrates, could enter the binding site?

Q7: [Optional] As you have hopefully observed HIV protease is a homodimer (i.e. it is composed of two identical chains). With the aid of the graphic display can you identify secondary structure elements that are likely to only form in the dimer rather than the monomer?

Bio3D package for structural bioinformatics

```
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? 198 residues

Q8: Name one of the two non-protein residues? HOH, MK1

Q9: How many protein chains are in this structure? 2 chains

```
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>
```

```
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
```

functional dynamics prediction

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

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

```
summary(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) ]

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

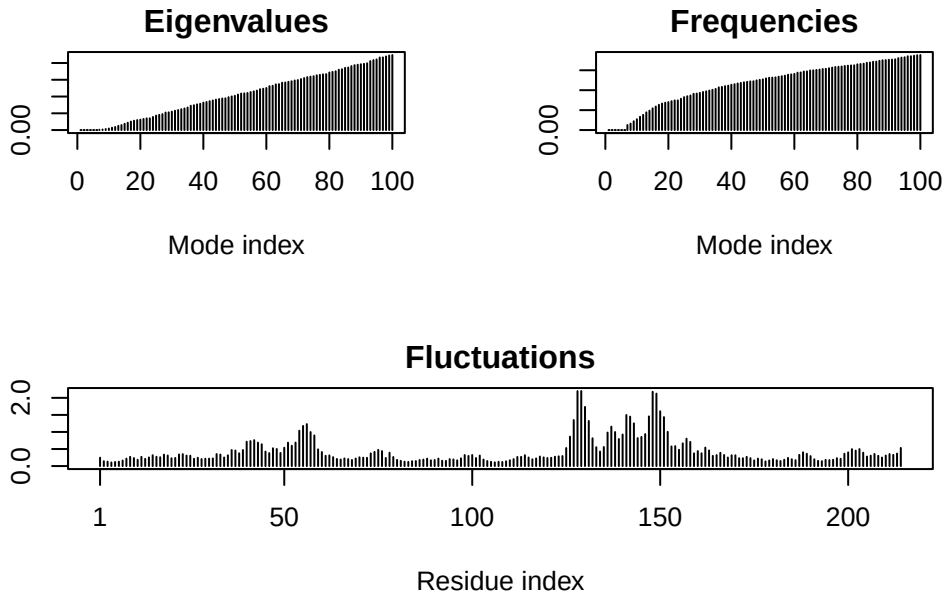
```
source("https://tinyurl.com/viewpdb")
library(r3dmol)

view.pdb(pdb,backgroundColor="pink")
```

```
view.pdb(adk)
```

```
modes <- nma(adk)
```

```
 Building Hessian...        Done in 0.024 seconds.
 Diagonalizing Hessian...   Done in 0.452 seconds.
```

```
plot(modes)
```




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

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

```
Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/sd/cf1692xd5vqdq51k8c_38cxh0000gn/T//RtmpIwUQ6O/6s36.pdb exists.
Skipping download
```

```
   PDB has ALT records, taking A only, rm.alt=TRUE
```

```
modes <- nma(adk)
```

```
 Building Hessian...         Done in 0.02 seconds.
 Diagonalizing Hessian...    Done in 0.455 seconds.
```

```
mktrj(modes,pdb=adk, file="adk.pdb")
```