class10

Eli Sobel A69027989

download csv from pdb website-> "Analyze" > "PDB Statistics" > "by Experimental Method and Molecular Type"

```
pdbstats <- read.csv("C:/Users/eliso/Downloads/Data_Export_Summary.csv", 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
```

```
pdbstats$Total
```

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

Remove commas

```
convert_comma_numbers <- function(x) {
  x <- gsub(",","",x)
  x <- as.numeric(x)

  return(x)
}

convert_comma_numbers(pdbstats$Total)
```

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

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

In PDB, 83% of structures are solved by X-ray, and 10% of the structures are solved by EM.

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.

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

```
[1] 226707
```

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

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

How to do the same thing without `apply()`

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

n.xray/n.tot * 100
```

```
[1] 83.25592
```

```
n.em/n.tot * 100
```

```
[1] 10.2348
```

Another way to read in the csv without having to remove commas is to download and use the readr package

```
# library(readr)
# read_csv("C:/Users/eliso/Downloads/Data_Export_Summary.csv")
```

make new dataframe with no commas

```
pdb_nc <- apply(pdbstats, c(1,2), convert_comma_numbers)
pdb_nc
```

```
                        X.ray    EM   NMR Multiple.methods Neutron Other
Protein (only)         167317 15698 12534               208      77    32
Protein/Oligosaccharide  9645  2639    34                 8       2     0
Protein/NA               8735  4718   286                 7       0     0
Nucleic acid (only)      2869   138  1507                14       3     1
Other                     170    10    33                 0       0     0
Oligosaccharide (only)     11     0     6                 1       0     4
                        Total
Protein (only)         195866
Protein/Oligosaccharide  12328
Protein/NA               13746
Nucleic acid (only)       4532
Other                      213
Oligosaccharide (only)      22
```

```
# pdb$Total/ sum(pdb$Total)
```

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?

Using Mol*

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

HOH308 is the conserved water molecule

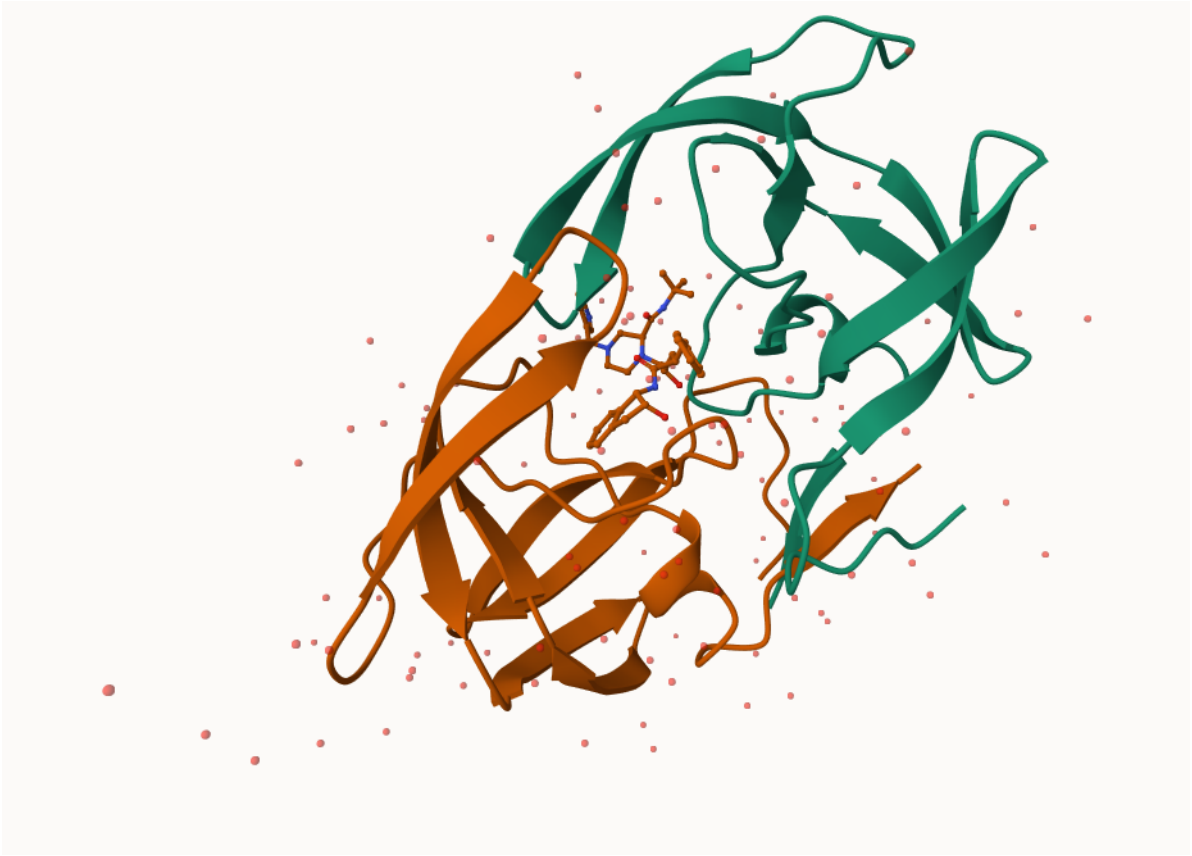

Figure 1: My first image from Mol-star

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 2: The very important aspartic acid residue!

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
```

```
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)[25]
```

```
 25
"D"
```

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

198

```
length( pdbseq(pdb))
```

```
[1] 198
```

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

Q9: How many protein chains are in this structure?

There are two chains

Functional dynamics prediction

```
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
```

```
# install r3dmol and shiny
# source("https://tinyurl.com/viewpdb")
# library(r3dmol)
# view.pdb(pdb)
```

```
# install r3dmol and shiny
# source("https://tinyurl.com/viewpdb")
# library(r3dmol)
# view.pdb(adk)
```

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

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

```
#adk <- read.pdb("6s36")
#modes <- nma(adk)
#mktrj(modes, pdb=adk, file="adk.pdb")
```