Class 10: Structural Bioinformatics 1

Emily Ignatoff (A16732102)

Table of contents

PDB Statistics

The main repository for biomolecular structural data is the PDB database: < http://www.rcsb.org/ >

Let's see what this database contains. I went to PDB > Analyze > PDB Statistics > by Exp method and molecular type.

```
pdbstats <- read.csv("Data Export Summary.csv", row.names=1)
pdbstats
```

```
                        X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)        169,563 16,774 12,578              208      81    32
Protein/Oligosaccharide 9,939  2,839     34                8       2     0
Protein/NA              8,801  5,062    286                7       0     0
Nucleic acid (only)     2,890    151  1,521               14       3     1
Other                     170     10     33                0       0     0
Oligosaccharide (only)     11      0      6                1       0     4
                        Total
Protein (only)        199,236
```

```
Protein/Oligosaccharide  12,822
Protein/NA               14,156
Nucleic acid (only)       4,580
Other                       213
Oligosaccharide (only)       22
```

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

```
x <- pdbstats$X.ray
as.numeric(sub(",","",x))
```

```
[1] 169563   9939   8801   2890    170     11
```

The comma in these numbers prevents them from being integers. I can fix this using a different read function from the **readr** package (as opposed to the as.numeric function shown here which substitutes the comma for nothing)

```
library(readr)
pdbstats <- read_csv("Data Export Summary.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.
```

```
pdbstats
```

```
# 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)      169563 16774 12578                208      81    32 199236
2 Protein/Oligosacc~    9939  2839    34                  8       2     0  12822
3 Protein/NA            8801  5062   286                  7       0     0  14156
4 Nucleic acid (onl~    2890   151  1521                 14       3     1   4580
5 Other                  170    10    33                  0       0     0    213
6 Oligosaccharide (~      11     0     6                  1       0     4     22
```

I want the row names to all have consistent formatting with lowercase letters and no spaces:

```
library(janitor)
```

```
Warning: package 'janitor' was built under R version 4.3.3
```

```

Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
df <- clean_names(pdbstats)
df
```

```
# 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)        169563 16774 12578              208      81    32 199236
2 Protein/Oligosacchar~   9939  2839    34                8       2     0  12822
3 Protein/NA              8801  5062   286                7       0     0  14156
4 Nucleic acid (only)     2890   151  1521               14       3     1   4580
5 Other                    170    10    33                0       0     0    213
6 Oligosaccharide (onl~     11     0     6                1       0     4     22
```

Total Number of x-ray structures:

```
sum(df$x_ray)
```

```
[1] 191374
```

total number of em structures:

```
sum(df$em)
```

```
[1] 24836
```

total number of structures:

```
sum(df$total)
```

```
[1] 231029
```

Percent by x-ray and em:

```
(sum(df$x_ray)/sum(df$total))*100
```

```
[1] 82.83549
```

```
(sum(df$em)/sum(df$total))*100
```

```
[1] 10.75017
```

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

```
sum(df$total[1:3])
```

```
[1] 226214
```

```
(sum(df$total[1:3])/sum(df$total)) * 100
```

```
[1] 97.91585
```

> 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?

24,695 Structures

Visualizing the HIV-1 protease structure

Using mol*

The main Mol* homepage at: https://molstar.org/viewer/ will allow us to use protein accession numbers to view protein structures in 3D space (we can also give our own PDB files if we have them)

Figure 1: the HIV-1 Protease molecule(1HSG)

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

This creates more open space to easily view the protein structure. The protein is surrounded by water in reality but we must ignore and simplify that for the clearest and most informative protein view

> 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

This water molecule is HOH 308 This molecule sits just above the ligand and creates hydrogen bonds between the MK1 compound and the HIV polymer.

> 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.

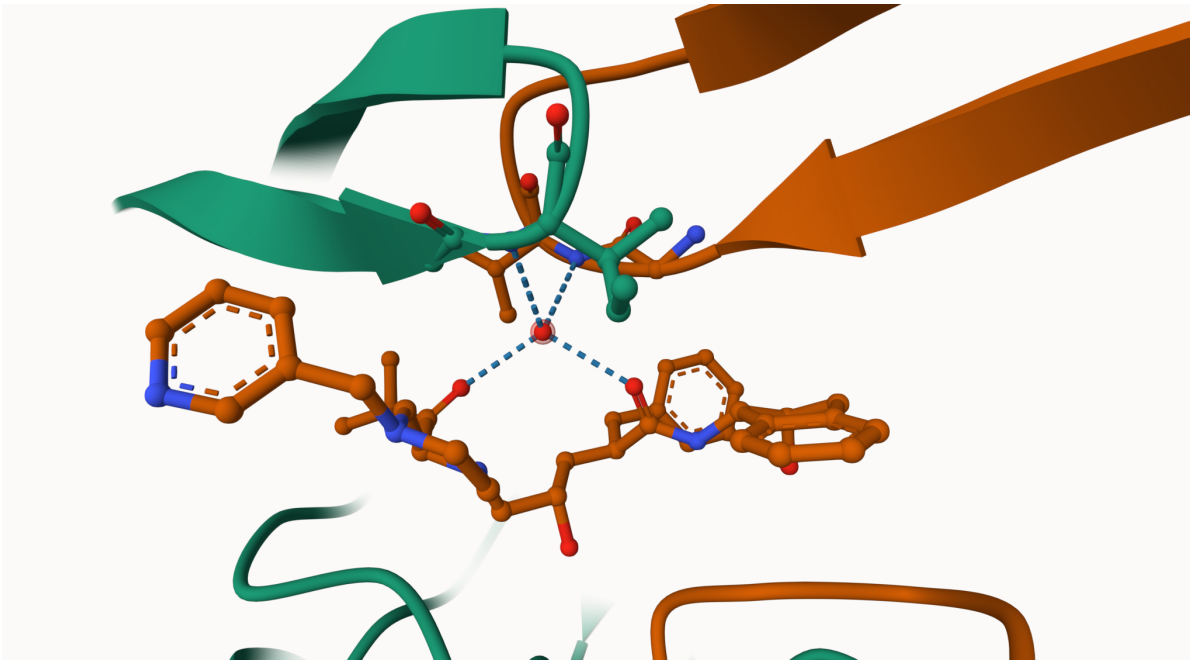


Figure 2: HOH 308 connecting the MK1 ligand to the polymer

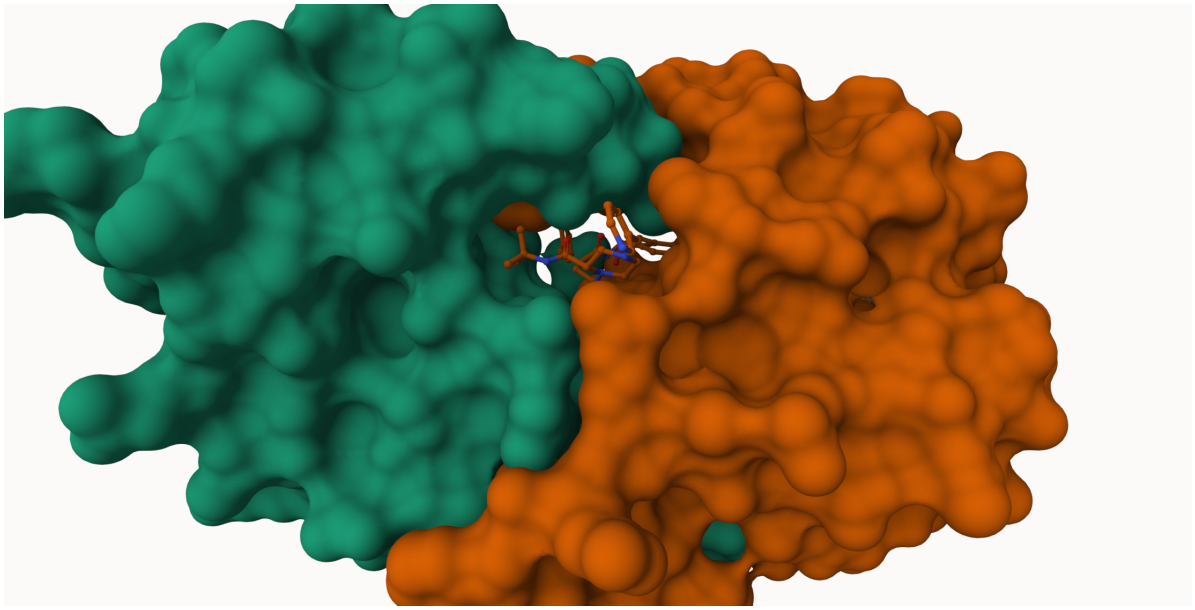


Figure 3: The MK1 ligand fitting inside the polymer cavity

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?

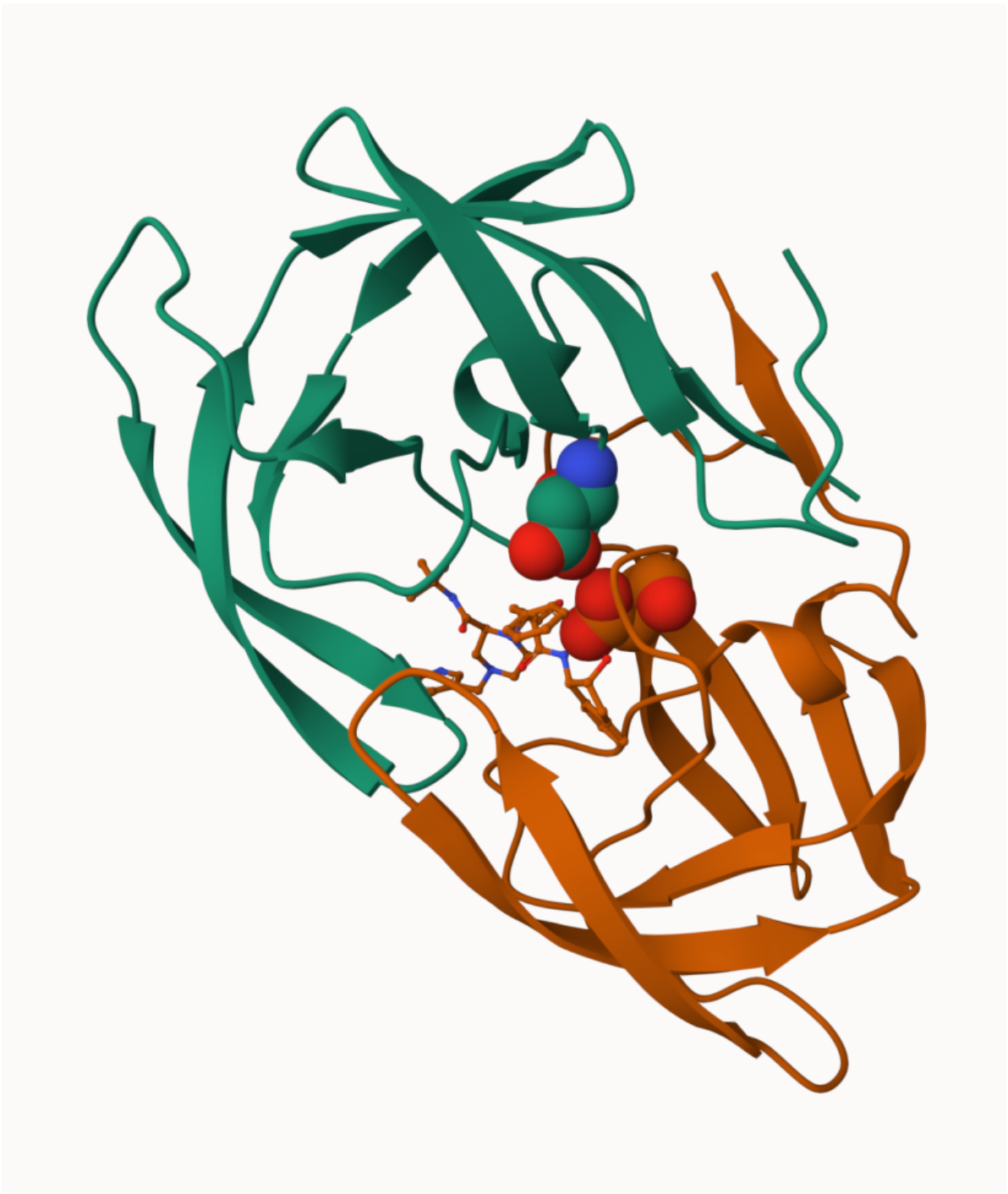


Figure 4: The aspartic acid residues on the homodimer

Introduction to Bio3D in R

We can use the **bio3d** package for stuctural bioinformatics to read PDB into R

```
library(bio3d)
```

```
Warning: package 'bio3d' was built under R version 4.3.3
```

```
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?

There are 198 amino acid residues in this pdb object

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

```
[1] 198
```

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

MK1

Q9: How many protein chains are in this structure?

There are 2 protein chains

Looking at the `pdb` object in more detail:

```
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>
```

Let's try a new function not yet in the bio3d package. It requires the **r3dmol** and **shiny** package:

```
library(r3dmol)
```

```
Warning: package 'r3dmol' was built under R version 4.3.3
```

```
library(shiny)
```

```
Warning: package 'shiny' was built under R version 4.3.3
```

```
source("https://tinyurl.com/viewpdb")
#view.pdb(pdb, backgroundColor = "lavender")
```

Predicting Functional Dynamics

we can use the `nma()` function in bio3d to predict the large scale functional motions of biomolecules:

```
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
```

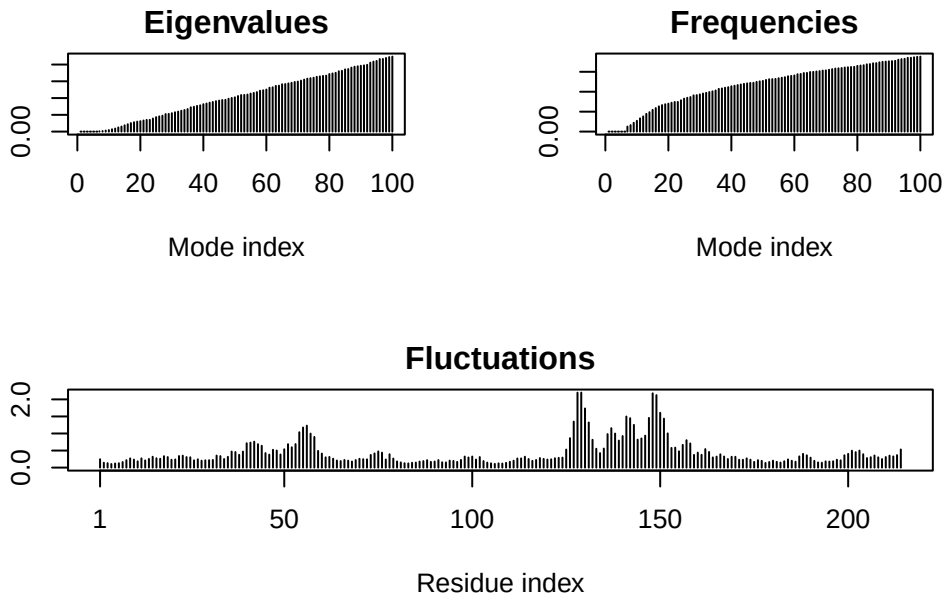
```
      VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
      YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG

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

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

```
 Building Hessian...        Done in 0.03 seconds.
 Diagonalizing Hessian...   Done in 0.23 seconds.
```

```
plot(m)
```



Write out a predicted trajectory of the molecular motion:

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