Class9: Struectural Bioinformatics pt1

Diana Furlan

The main database for structural data is called PBD

data from: https://www.rcsb.org/stats

#Read file into R

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

```
                         X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)         167,192 15,572 12,529               208      77    32
Protein/Oligosaccharide  9,639  2,635     34                 8       2     0
Protein/NA               8,730  4,697    286                 7       0     0
Nucleic acid (only)      2,869    137  1,507                14       3     1
Other                      170     10     33                 0       0     0
Oligosaccharide (only)      11      0      6                 1       0     4
                         Total
Protein (only)         195,610
Protein/Oligosaccharide  12,318
Protein/NA               13,720
Nucleic acid (only)       4,531
Other                       213
Oligosaccharide (only)       22
```

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

```
pdbdb$Total
```

```
[1] "195,610" "12,318"  "13,720"  "4,531"   "213"     "22"
```

Removing commas for numeric function

```
sub(",", "", pdbdb$Total)
```

```
[1] "195610" "12318"  "13720"  "4531"   "213"    "22"
```

```
#as.numeric(pdbdb$Total)
```

```
x <- pdbdb$Total
as.numeric(sub(",", "", pdbdb$Total))
```

```
[1] 195610  12318  13720   4531    213     22
```

```
#install.packages("readr")
```

```
library(readr)
pdbdb <- 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.
```

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

```
sum(pdbdb$'X-ray')/sum(pdbdb$Total)* 100
```

```
[1] 83.30359
```

```
sum(pdbdb$EM)/sum(pdbdb$Total)* 100
```

```
[1] 10.18091
```

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

```
pdbdb$Total[1] / sum(pdbdb$Total) * 100
```

```
[1] 86.39483
```

> 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?

##Mol*

https://molstar.org/viewer/

we will use PBD code: 1HSG

Accessing PDB file

Figure 1: A first image from molstar

Some more custom images

![showing most expensive water pocket] (1HSGpocket.png)

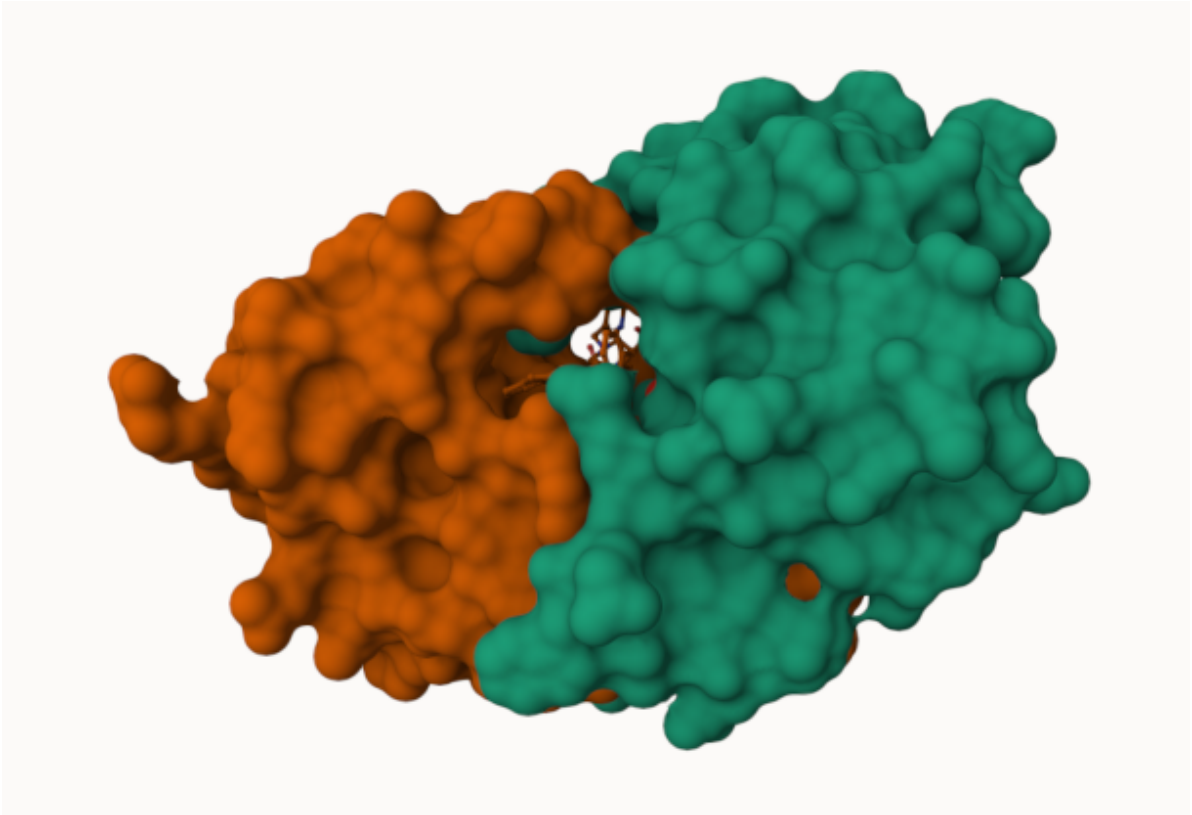


Figure 2: same as before but with molecular space representation

##The Bio3D package

```
#install.packages("bio3d")
library(bio3d)
##   Note: Accessing on-line PDB file
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>
```

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

```
sum(pdb$calpha)
```

```
[1] 198
```

```
length(pdb$seqres)
```

```
[1] 198
```

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

HOH and MK1

Q9: How many protein chains are in this structure?

2

```
 unique(pdb$atom$chain)
```

```
[1] "A" "B"
```

##Predicting functional motions of a single structure

Let's read a new PDB structure of Adenylate Kinase and perform Normal mode analysis.

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

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

```
##   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
```

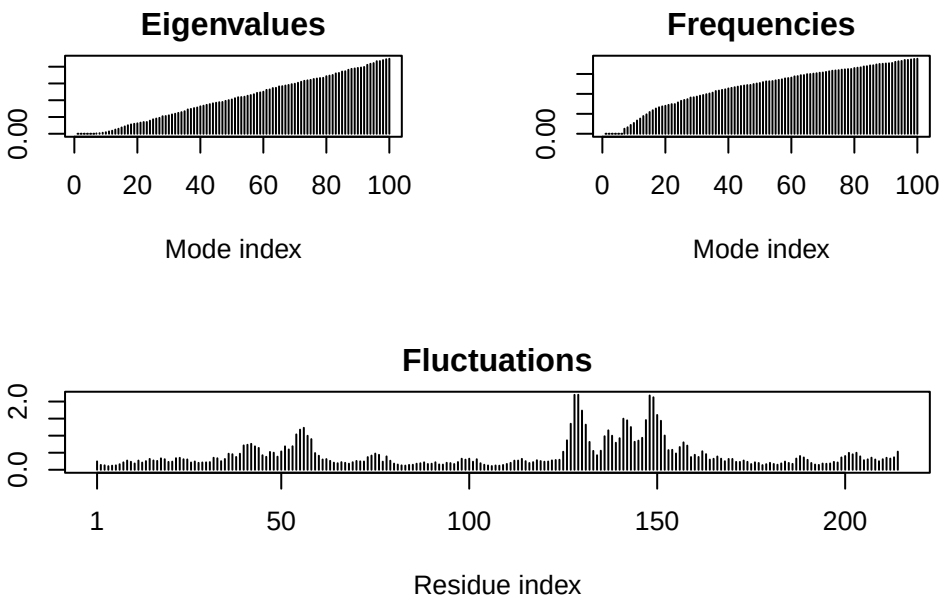
Normal mode analysis (NMA) is a structural bioinformatics method to predict protein flexibility and potential functional motions (a.k.a. conformational changes).

#prediction

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

```
 Building Hessian...        Done in 0.08 seconds.
 Diagonalizing Hessian...   Done in 0.82 seconds.
```

```
plot(m)
```



Movie :) molecular "trajectory"

#{r} `mktrj(m, file="adk_m7.pdb") #`

I open in molstar

##comparative analisis of protein structure

```
#install.packages("bio3d")
```

```
library(bio3d)
aa <- get.seq("1ake_A")
```

```
Warning in get.seq("1ake_A"): Removing existing file: seqs.fasta
```

```
Fetching... Please wait. Done.
```

i ran these cmds in the R bran/console

```
#install.packages("bio3d")
#install.packages("devtools")
#install.packages("BiocManager")
```

Q10 'msa' pac is from BioConductor.

```
BiocManager::install("msa")
```

```
Bioconductor version 3.20 (BiocManager 1.30.25), R 4.4.1 (2024-06-14 ucrt)

Warning: package(s) not installed when version(s) same as or greater than current; use
  `force = TRUE` to re-install: 'msa'

Installation paths not writeable, unable to update packages
  path: C:/Program Files/R/R-4.4.1/library
  packages:
    boot, foreign, MASS, Matrix, nlme, survival

Old packages: 'curl', 'evaluate', 'fs', 'glue', 'gtable', 'Rcpp', 'rmarkdown',
  'tinytex', 'withr', 'xfun'
```

```
devtools::install_bitbucket("Grantlab/bio3d-view")
```

```
WARNING: Rtools is required to build R packages, but is not currently installed.

Please download and install Rtools 4.4 from https://cran.r-project.org/bin/windows/Rtools/.

Skipping install of 'bio3d.view' from a bitbucket remote, the SHA1 (dd153987) has not change
  Use `force = TRUE` to force installation
```

Q13:

```
ncol(aa$ali)
```

```
[1] 214
```

Blast or hmmer search

```
#b <- blast.pdb(aa)
```

```
#head(hits$pdb.id)
```

```
#hits <- plot(b)
```

Precalculated results:

```
hits <- NULL
hits$pdb.id <- c('1AKE_A','6S36_A','6RZE_A','3HPR_A','1E4V_A','5EJE_A','1E4Y_A','3X2S_A','6H
```

Download releated PDB files

```
files <- get.pdb(hits$pdb.id, path="pdbs", split=TRUE, gzip=TRUE)
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1AKE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6S36.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6RZE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3HPR.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1E4V.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/5EJE.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1E4Y.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3X2S.pdb exists. Skipping download
```

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6HAP.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/6HAM.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4K46.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/3GMT.pdb exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4PZL.pdb exists. Skipping download


  |
  |                                                                      |   0%
  |
  |=====                                                                 |   8%
  |
  |===========                                                           |  15%
  |
  |================                                                      |  23%
  |
  |======================                                                |  31%
  |
  |============================                                          |  38%
  |
  |================================                                      |  46%
  |
  |====================================                                  |  54%
  |
  |========================================                              |  62%
  |
  |============================================                          |  69%
  |
  |================================================                      |  77%
  |
  |========================================================              |  85%
  |
```

```
  |=================================================================     |  92%
  |
  |======================================================================| 100%
```

#Aign superimposed structrures

```
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

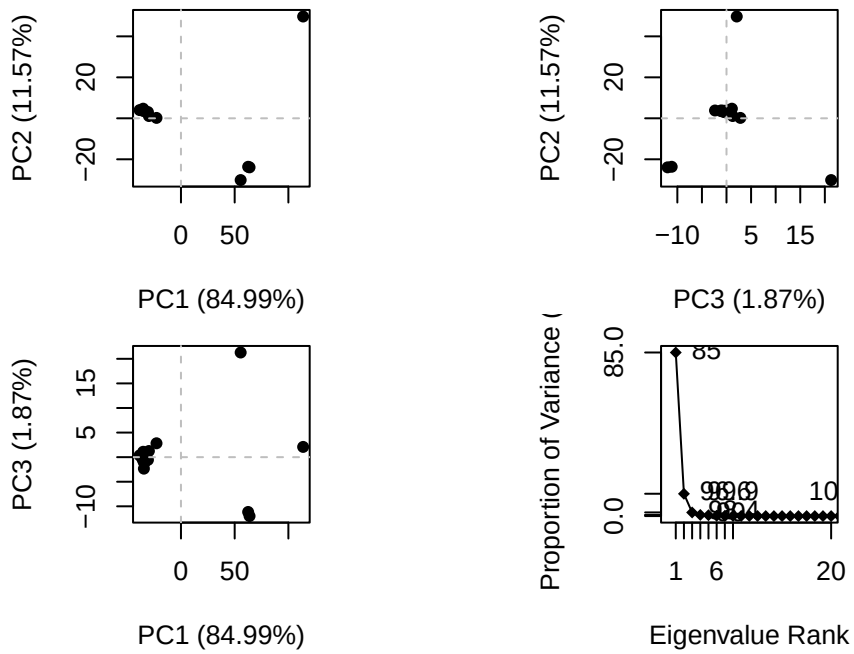
```
Reading PDB files:
pdbs/split_chain/1AKE_A.pdb
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/6RZE_A.pdb
pdbs/split_chain/3HPR_A.pdb
pdbs/split_chain/1E4V_A.pdb
pdbs/split_chain/5EJE_A.pdb
pdbs/split_chain/1E4Y_A.pdb
pdbs/split_chain/3X2S_A.pdb
pdbs/split_chain/6HAP_A.pdb
pdbs/split_chain/6HAM_A.pdb
pdbs/split_chain/4K46_A.pdb
pdbs/split_chain/3GMT_A.pdb
pdbs/split_chain/4PZL_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
..   PDB has ALT records, taking A only, rm.alt=TRUE
....   PDB has ALT records, taking A only, rm.alt=TRUE
.   PDB has ALT records, taking A only, rm.alt=TRUE
...

Extracting sequences

pdb/seq: 1   name: pdbs/split_chain/1AKE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 2   name: pdbs/split_chain/6S36_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 3   name: pdbs/split_chain/6RZE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 4   name: pdbs/split_chain/3HPR_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 5   name: pdbs/split_chain/1E4V_A.pdb
```

```
pdb/seq: 6   name: pdbs/split_chain/5EJE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 7   name: pdbs/split_chain/1E4Y_A.pdb
pdb/seq: 8   name: pdbs/split_chain/3X2S_A.pdb
pdb/seq: 9   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 10   name: pdbs/split_chain/6HAM_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 11   name: pdbs/split_chain/4K46_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 12   name: pdbs/split_chain/3GMT_A.pdb
pdb/seq: 13   name: pdbs/split_chain/4PZL_A.pdb
```

```
pc.xray <- pca(pdbs)
plot(pc.xray)
```




```
# Visualize first principal component
pc1 <- mktrj(pc.xray, pc=1, file="pc_1.pdb")
```

```
uniprot <- 248838887
  pdb <- 195610
pdb/uniprot *100
```

```
[1] 0.0786091
```