Class 10: Structural Bioinformatics Part 1

Zainab Fatima (A16880407)

2025-02-06

Table of contents

The PDB Database

The main repository of biomolecular structure data is called the PDB found at: https://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")
pdbstats
```

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

```
2  12,822
3  14,156
4   4,580
5     213
6      22
```

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

The answer is that around 83% are solved by X-ray and 11% is solved by RM. The code is listed below.

```
pdbstats$X.ray
```

```
[1] "169,563" "9,939"   "8,801"   "2,890"   "170"     "11"
```

```
#output has quotes around them because they're characters, can't do math with them
```

The comma in these numbers is causing them to be read as characters rather than numberic.

You can fix this by replacing "," for nothing "" with the `sub()` function:

```
x <- pdbstats$X.ray
#as.numeric(sub(",", "", x)) gets rid of comma
#could then sum it and get total number of x-ray
sum(as.numeric(sub(",", "", x)))
```

```
[1] 191374
```

Or I can use the **readr** package and the `read_csv()` function.

```
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.
```

```
#this read csv has a _
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 to clean the column names so that they are lowercase and don't have spaces in them.

```
colnames(pdbstats)
```

```
[1] "Molecular Type"   "X-ray"            "EM"               "NMR"
[5] "Multiple methods" "Neutron"          "Other"            "Total"
```

```
library(janitor)
```

```
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 structures

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

```
[1] 231029
```

The percentage of Xray structures

```
percent_xray <- (sum(df$x_ray)/sum(df$total))*100
percent_xray
```

```
[1] 82.83549
```

The percentage of EM structures

```
percent_em <- sum(df$em)/sum(df$total) * 100
percent_em
```

```
[1] 10.75017
```

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

The answer is 0.86238. The code is below:

```
library(dplyr)
```

```
Attaching package: 'dplyr'

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

    filter, lag
```

```
The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

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

protein_only_total <- df %>%
  filter(molecular_type == "Protein (only)") %>%
  pull(total)

protein_only_total
```

```
[1] 199236
```

```
# find the proportion of protein (only) over total

prop_protein_only <- protein_only_total/ sum(df$total)

prop_protein_only
```

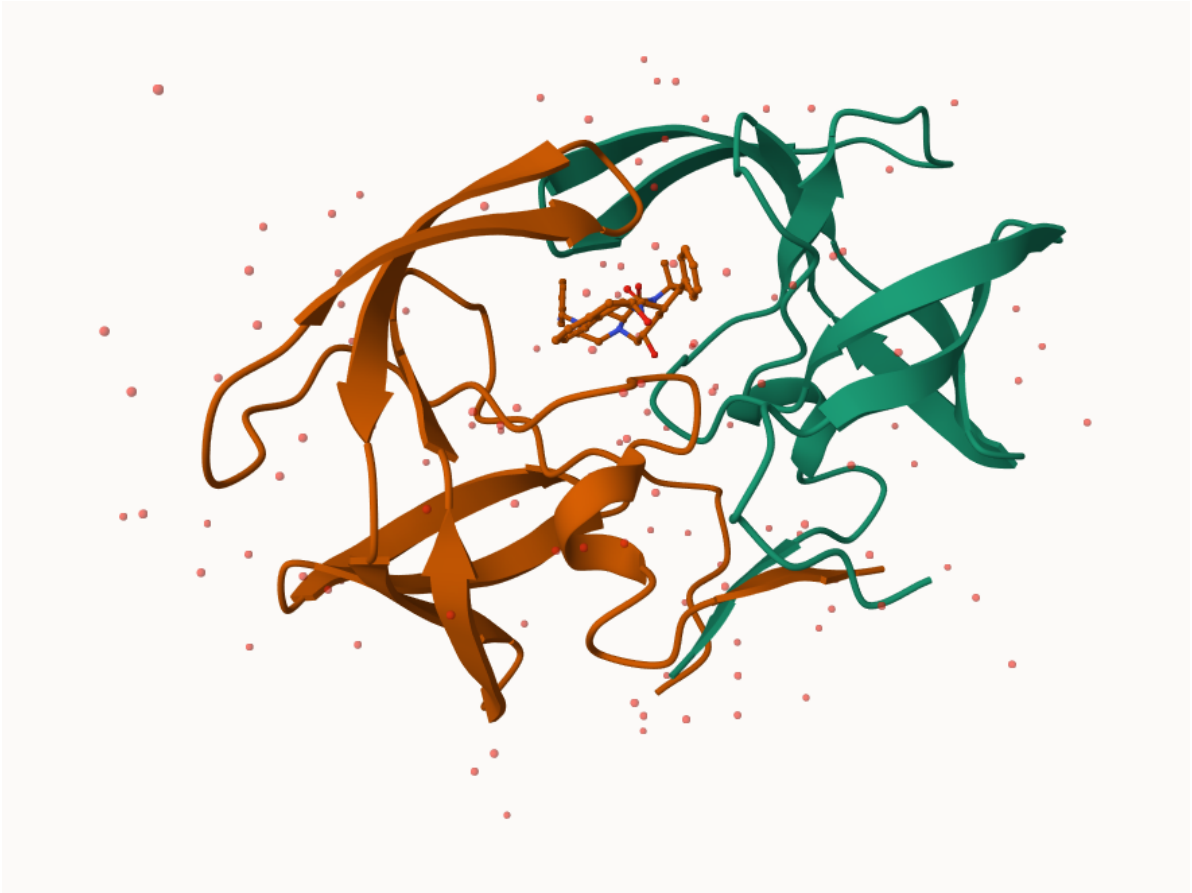
```
[1] 0.8623852
```

> 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?

231,029 HIV protease-1 structures are in the current PDB.

Using Mol*

The main Mol* homepage at: https://molstar.org/viewer/ We can input our own PDB files or just give it a PDB database accession code (4 letter PDB code).



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

In this structure, the water molecules are represented with just the oxygen or O atom. This is because hydrogen atoms have less electron density and thurs are often not resolved in methods that determine protein structures. They can also be omitted for clarity.

Q5. Identify the critical area where water interacts with molecule

Residue #308 is where water interacts with the molecule.

Figure 1: Ligand Interactions with H2O

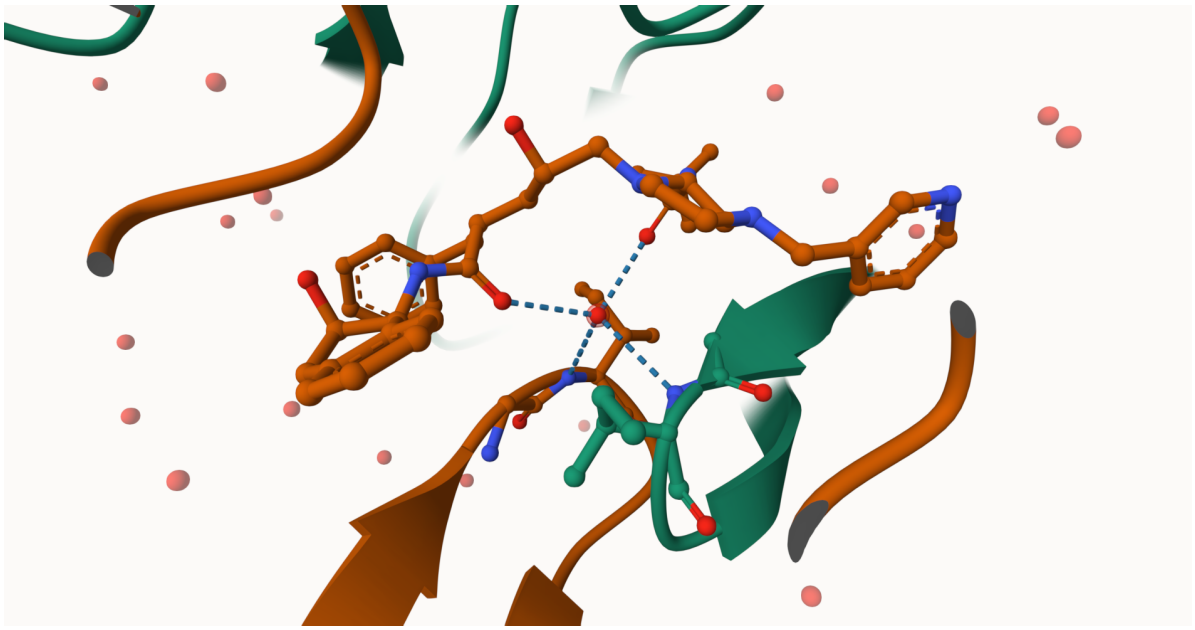


Figure 2: Close-up of Ligand Interactions with Water

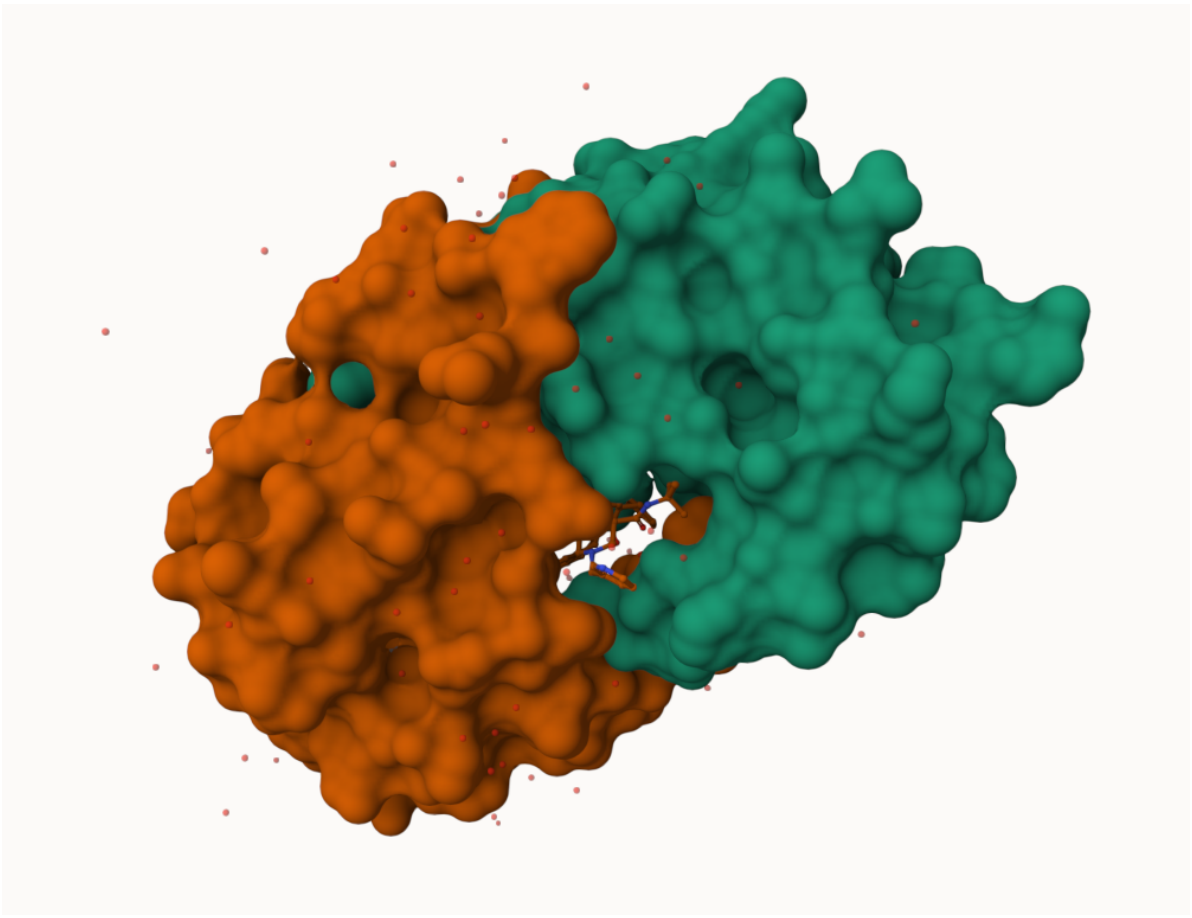


Figure 3: Surface Representation of Ligand - Polymer Interaction

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.

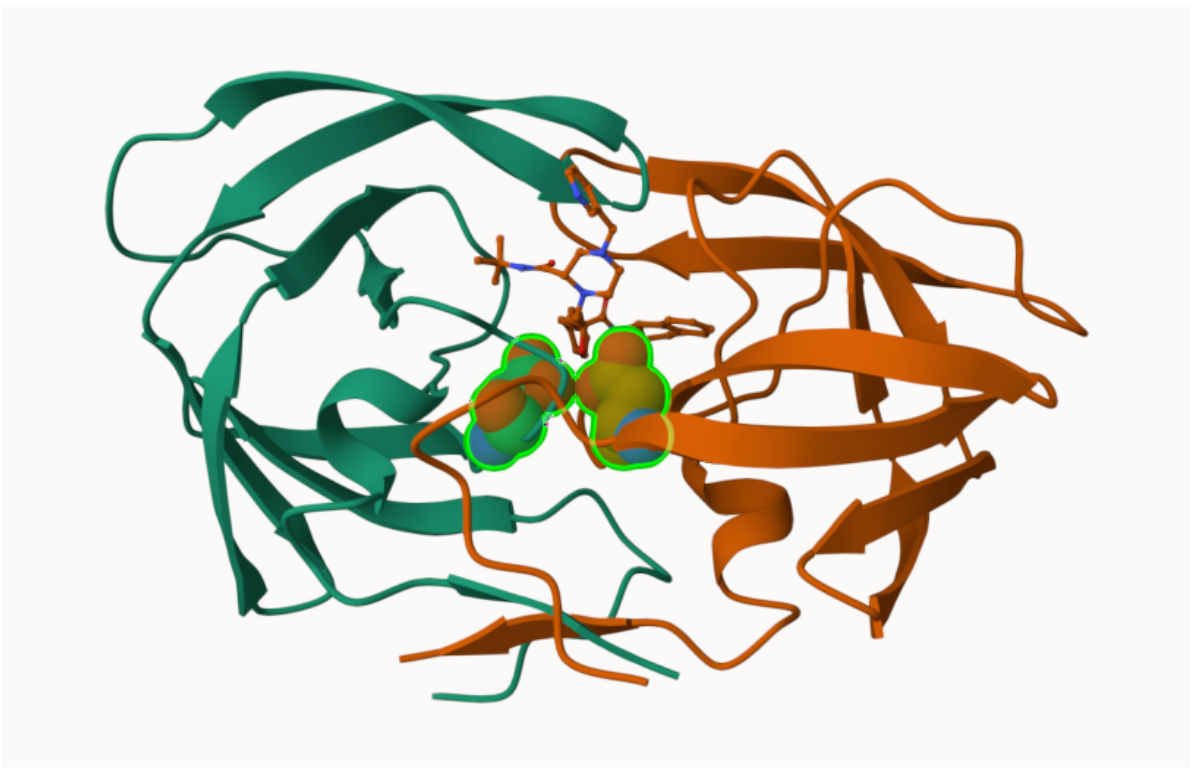


Figure 4: The important Asp25 amino acids

Introduction to Bio3D in R

We can use the **bio3d** package for structural bioinformatics to read PDb data into R

```
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?

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

```
[1] 198
```

198 residues

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

HOH or MK1

Q9: How many protein chains are in this structure?

2 chains (chains A and B)

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:

```
library(r3dmol)
source("https://tinyurl.com/viewpdb")
#view.pdb(pdb, backgroundColor ="pink")
```

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
```

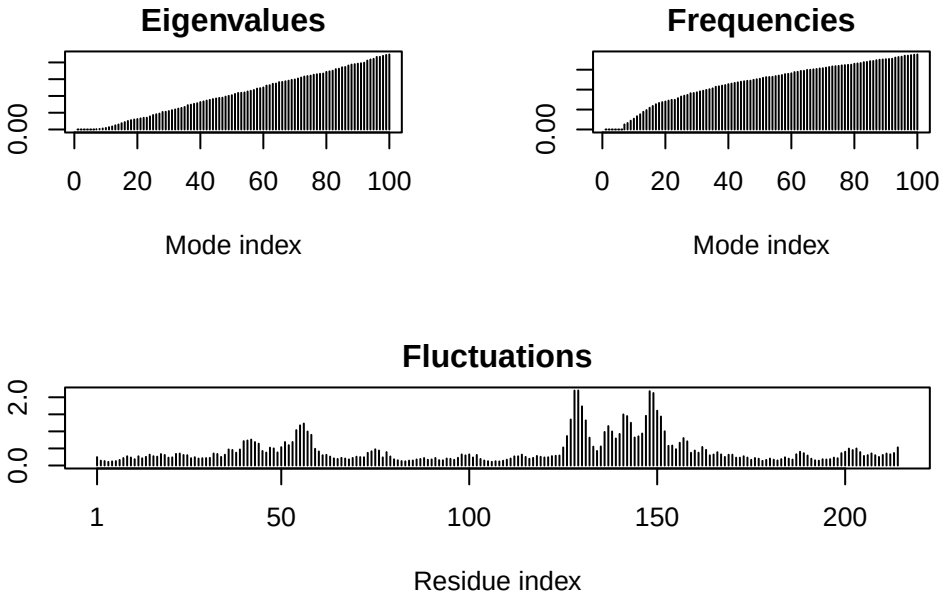
```

 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 0.17 seconds.
```

```
plot(m)
```



Write out a trajectory of the predicted molecuar motion:

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

Comparative Structure Analysis of Adenylate Kinase

The goal of this section is to do a principal component analysis (PCA) on the complete collection of adenylate kinase structures in the PDB.

Starting from one Adk PDB identifier (PDB ID: 1AKE) we will search the entire PDB for related structures using BLAST, fetch, align, and superpose the identified structures, perform PCA, and calculate the normal modes of each individual structure to find potential differences in structural flexibility.

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

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

```
Bioconductor version 3.20 (BiocManager 1.30.25), R 4.4.2 (2024-10-31 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-4.4.2/library
  packages:
    class, cluster, foreign, KernSmooth, MASS, Matrix, nlme, nnet, rpart,
    spatial, survival

Old packages: 'rlang'
```

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

```
WARNING: Rtools is required to build R packages, but no version of Rtools compatible with R 4

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 changed
  Use `force = TRUE` to force installation
```

Q10. Which of the packages above is found only on BioConductor and not CRAN?

`msa` (Multiple Sequence Alignment) is found only on Bioconductor and not CRAN.

Q11. Which of the above packages is not found on BioConductor or CRAN?

`bio3d-view` is not found on either BioConductor or CRAN, it is installed from Bitbucket.

Q12. True or False? Functions from the devtools package can be used to install packages from GitHub and BitBucket?

True. The 'devtools1 package provides functions such as install_github() for installing packages from GitHub and install_bitbucket() for installing packages from BitBucket.

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

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

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

```
aa
```

```
              1        .         .         .         .         .         60
pdb|1AKE|A    MRIILLGAPGAGKGTQAQFIMEKYGIPQISTGDMLRAAVKSGSELGKQAKDIMDAGKLVT
              1        .         .         .         .         .         60

             61        .         .         .         .         .         120
pdb|1AKE|A    DELVIALVKERIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFDVPDELIVDRI
             61        .         .         .         .         .         120

            121        .         .         .         .         .         180
pdb|1AKE|A    VGRRVHAPSGRVYHVKFNPPKVEGKDDVTGEELTTRKDDQEETVRKRLVEYHQMTAPLIG
            121        .         .         .         .         .         180

            181        .         .         .   214
pdb|1AKE|A    YYSKEAEAGNTKYAKVDGTKPVAEVRADLEKILG
            181        .         .         .   214

Call:
```

```
  read.fasta(file = outfile)

Class:
  fasta

Alignment dimensions:
  1 sequence rows; 214 position columns (214 non-gap, 0 gap)

+ attr: id, ali, call
```

Q13. How many amino acids are in this sequence, i.e. how long is this sequence?

There are 214 amino acids in this sequence.

We can use this sequence as a quert to BLAST search the PDB to find similar sequences and structures.

```
# Blast or hmmer search
#b <- blast.pdb(aa)
```

```
# Plot a summary of search results

#hits <- plot(b)
```

```
# List out some 'top hits'
#head(hits$pdb.id)
```

```
# BLAST timed out, used vector of PDB IDs

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

```
# 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%
```

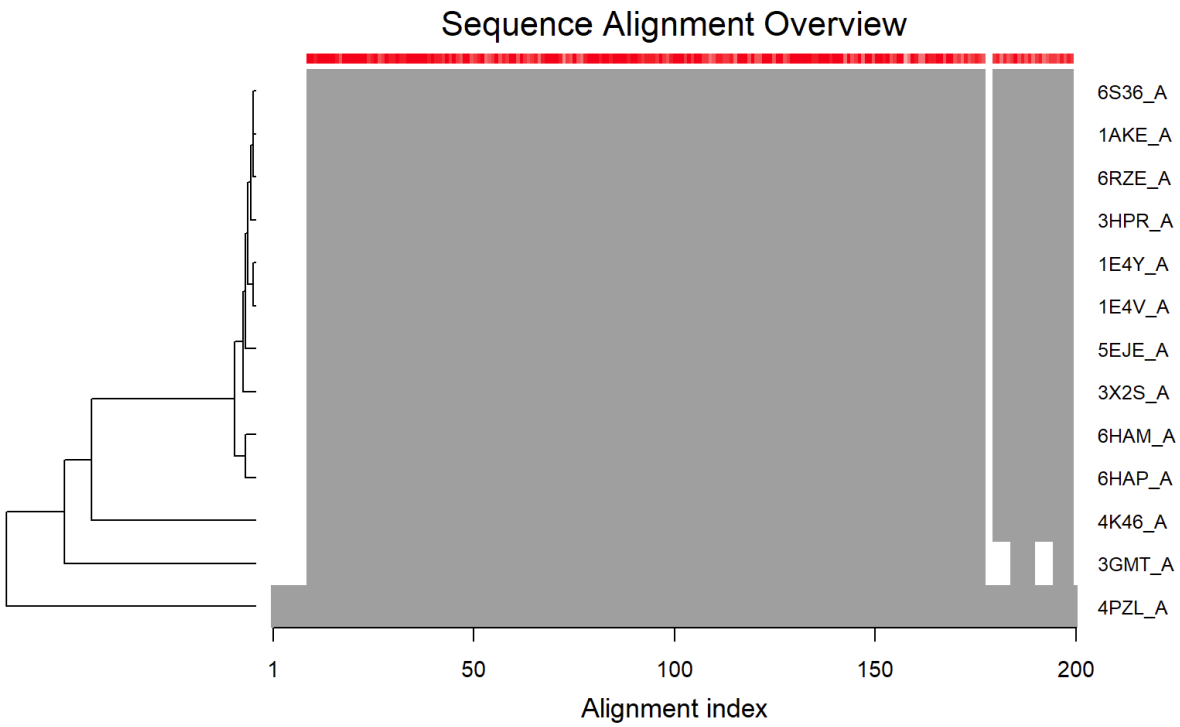
Next we will use the `psbaln()` function to align and optionally fit the identified PDB structures.

```
# Align releated PDBs
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
```

```r
# Vector containing PDB codes for figure axis
ids <- basename.pdb(pdbs$id)

# Draw schematic alignment
#plot(pdbs, labels=ids)
```



The above figure shows the schematic representation of the alignment. The gray reigons show the aligned residues, and the white reigons show gap regions. The red bar shows sequences conservation.

The function `pdb.annotate()` provides a convenient way of annotating the PDB files we collected. We will use the function to annotate each structure to its source species.

```
anno <- pdb.annotate(ids)
unique(anno$source)
```

```
[1] "Escherichia coli"
[2] "Escherichia coli K-12"
[3] "Escherichia coli O139:H28 str. E24377A"
[4] "Escherichia coli str. K-12 substr. MDS42"
[5] "Photobacterium profundum"
[6] "Burkholderia pseudomallei 1710b"
[7] "Francisella tularensis subsp. tularensis SCHU S4"
```

We can view all avaliable annotation data

```
anno
```

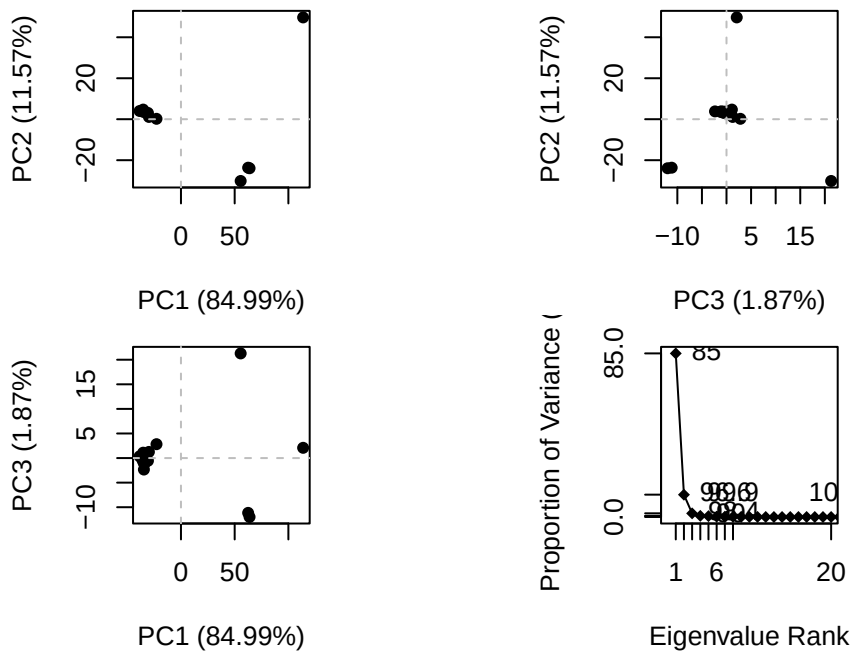
```
       structureId chainId macromoleculeType chainLength experimentalTechnique
1AKE_A        1AKE       A           Protein         214                  X-ray
6S36_A        6S36       A           Protein         214                  X-ray
6RZE_A        6RZE       A           Protein         214                  X-ray
3HPR_A        3HPR       A           Protein         214                  X-ray
1E4V_A        1E4V       A           Protein         214                  X-ray
5EJE_A        5EJE       A           Protein         214                  X-ray
1E4Y_A        1E4Y       A           Protein         214                  X-ray
3X2S_A        3X2S       A           Protein         214                  X-ray
6HAP_A        6HAP       A           Protein         214                  X-ray
6HAM_A        6HAM       A           Protein         214                  X-ray
4K46_A        4K46       A           Protein         214                  X-ray
3GMT_A        3GMT       A           Protein         230                  X-ray
4PZL_A        4PZL       A           Protein         242                  X-ray
       resolution       scopDomain                                    pfam
1AKE_A       2.00 Adenylate kinase Adenylate kinase, active site lid (ADK_lid)
6S36_A       1.60             <NA>                       Adenylate kinase (ADK)
6RZE_A       1.69             <NA> Adenylate kinase, active site lid (ADK_lid)
3HPR_A       2.00             <NA> Adenylate kinase, active site lid (ADK_lid)
1E4V_A       1.85 Adenylate kinase                       Adenylate kinase (ADK)
5EJE_A       1.90             <NA> Adenylate kinase, active site lid (ADK_lid)
1E4Y_A       1.85 Adenylate kinase Adenylate kinase, active site lid (ADK_lid)
3X2S_A       2.80             <NA>                       Adenylate kinase (ADK)
6HAP_A       2.70             <NA>                       Adenylate kinase (ADK)
6HAM_A       2.55             <NA> Adenylate kinase, active site lid (ADK_lid)
4K46_A       2.01             <NA>                       Adenylate kinase (ADK)
3GMT_A       2.10             <NA>                       Adenylate kinase (ADK)
4PZL_A       2.10             <NA>                       Adenylate kinase (ADK)
               ligandId
1AKE_A              AP5
6S36_A CL (3),NA,MG (2)
6RZE_A    NA (3),CL (2)
3HPR_A              AP5
1E4V_A              AP5
5EJE_A           AP5,CO
1E4Y_A              AP5
3X2S_A   JPY (2),AP5,MG
6HAP_A              AP5
6HAM_A              AP5
4K46_A      ADP,AMP,PO4
```

```
3GMT_A          SO4 (2)
4PZL_A       CA,FMT,GOL
                                                                                ligandName
1AKE_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
6S36_A                                    CHLORIDE ION (3),SODIUM ION,MAGNESIUM ION (2)
6RZE_A                                                   SODIUM ION (3),CHLORIDE ION (2)
3HPR_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
1E4V_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
5EJE_A                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE,COBALT (II) ION
1E4Y_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
3X2S_A N-(pyren-1-ylmethyl)acetamide (2),BIS(ADENOSINE)-5'-PENTAPHOSPHATE,MAGNESIUM ION
6HAP_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
6HAM_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
4K46_A                   ADENOSINE-5'-DIPHOSPHATE,ADENOSINE MONOPHOSPHATE,PHOSPHATE ION
3GMT_A                                                                    SULFATE ION (2)
4PZL_A                                                  CALCIUM ION,FORMIC ACID,GLYCEROL
                                                 source
1AKE_A                                Escherichia coli
6S36_A                                Escherichia coli
6RZE_A                                Escherichia coli
3HPR_A                             Escherichia coli K-12
1E4V_A                                Escherichia coli
5EJE_A           Escherichia coli O139:H28 str. E24377A
1E4Y_A                                Escherichia coli
3X2S_A         Escherichia coli str. K-12 substr. MDS42
6HAP_A           Escherichia coli O139:H28 str. E24377A
6HAM_A                             Escherichia coli K-12
4K46_A                         Photobacterium profundum
3GMT_A                   Burkholderia pseudomallei 1710b
4PZL_A Francisella tularensis subsp. tularensis SCHU S4

1AKE_A STRUCTURE OF THE COMPLEX BETWEEN ADENYLATE KINASE FROM ESCHERICHIA COLI AND THE INHIBI
6S36_A
6RZE_A
3HPR_A
1E4V_A
5EJE_A                                                                            Cryst
1E4Y_A
3X2S_A
6HAP_A
6HAM_A
4K46_A
3GMT_A
```

```
4PZL_A                                                                    The cryst
                                                    citation rObserved   rFree
1AKE_A                 Muller, C.W., et al. J Mol Biol (1992)   0.19600      NA
6S36_A                  Rogne, P., et al. Biochemistry (2019)   0.16320 0.23560
6RZE_A                  Rogne, P., et al. Biochemistry (2019)   0.18650 0.23500
3HPR_A  Schrank, T.P., et al. Proc Natl Acad Sci U S A (2009)   0.21000 0.24320
1E4V_A                   Muller, C.W., et al. Proteins (1993)   0.19600      NA
5EJE_A  Kovermann, M., et al. Proc Natl Acad Sci U S A (2017)   0.18890 0.23580
1E4Y_A                   Muller, C.W., et al. Proteins (1993)   0.17800      NA
3X2S_A                Fujii, A., et al. Bioconjug Chem (2015)   0.20700 0.25600
6HAP_A               Kantaev, R., et al. J Phys Chem B (2018)   0.22630 0.27760
6HAM_A               Kantaev, R., et al. J Phys Chem B (2018)   0.20511 0.24325
4K46_A                    Cho, Y.-J., et al. To be published    0.17000 0.22290
3GMT_A Buchko, G.W., et al. Biochem Biophys Res Commun (2010)   0.23800 0.29500
4PZL_A                       Tan, K., et al. To be published    0.19360 0.23680
         rWork spaceGroup
1AKE_A 0.19600  P 21 2 21
6S36_A 0.15940    C 1 2 1
6RZE_A 0.18190    C 1 2 1
3HPR_A 0.20620  P 21 21 2
1E4V_A 0.19600  P 21 2 21
5EJE_A 0.18630  P 21 2 21
1E4Y_A 0.17800   P 1 21 1
3X2S_A 0.20700 P 21 21 21
6HAP_A 0.22370    I 2 2 2
6HAM_A 0.20311       P 43
4K46_A 0.16730 P 21 21 21
3GMT_A 0.23500   P 1 21 1
4PZL_A 0.19130       P 32
```

We can then perform a PCA on the structural ensemmble (stored in the pdbs object).

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

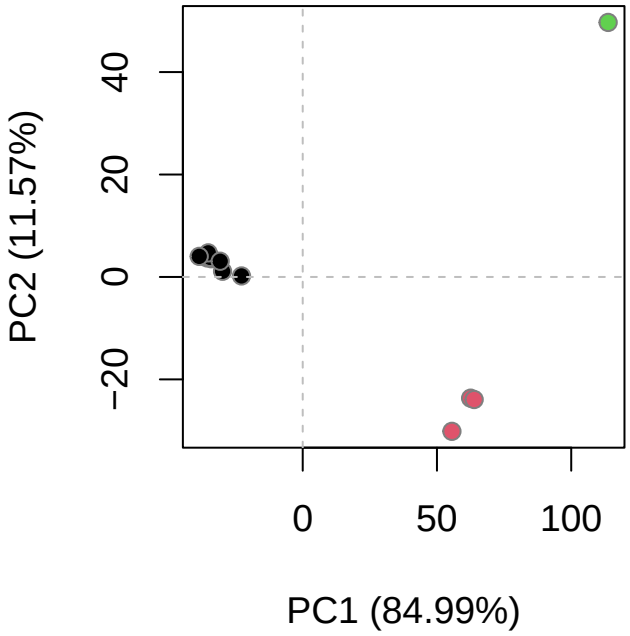


This figure is the results of the PCA on adenylate kinase X-ray structures. Each dot represents one PDB structure.

```
#Calculate RMSD
rd <- rmsd(pdbs)
```

```
Warning in rmsd(pdbs): No indices provided, using the 204 non NA positions
```

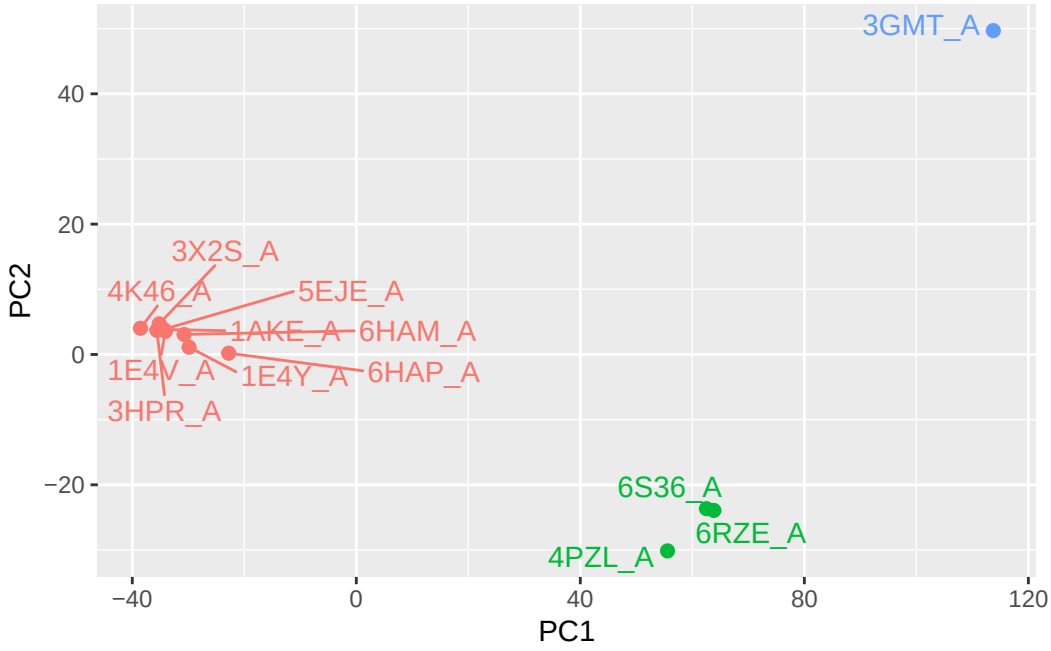
```
#Structure-based clustering
hc.rd <- hclust(dist(rd))
grps.rd <- cutree(hc.rd, k = 3)

plot(pc.xray, 1:2, col = "grey50", bg = grps.rd, pch = 21, cex = 1)
```



This figure is the projection of the adenylate kinase X-ray structures. Each dot represents one PDB structure.

We can plot our main PCA results with ggplot:

```
#Plotting results with ggplot2
library(ggplot2)
library(ggrepel)

df <- data.frame(PC1=pc.xray$z[,1],
                 PC2=pc.xray$z[,2],
                 col=as.factor(grps.rd),
                 ids=ids)

p <- ggplot(df) +
  aes(PC1, PC2, col=col, label=ids) +
  geom_point(size=2) +
  geom_text_repel(max.overlaps = 20) +
  theme(legend.position = "none")
p
```



Normal Mode Analysis (optional)

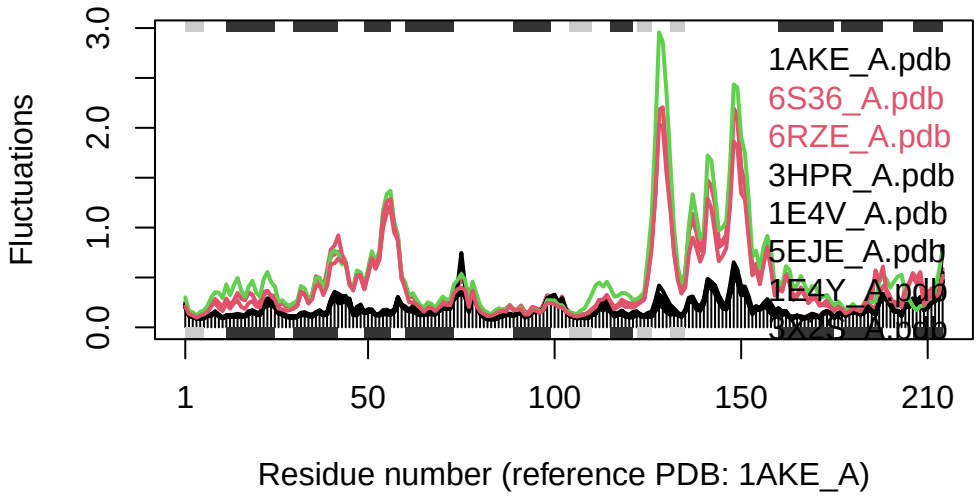
```
#NMA of all structures
modes <- nma(pdbs)
```

```
Details of Scheduled Calculation:
  ... 13 input structures
  ... storing 606 eigenvectors for each structure
  ... dimension of x$U.subspace: ( 612x606x13 )
  ... coordinate superposition prior to NM calculation
  ... aligned eigenvectors (gap containing positions removed)
  ... estimated memory usage of final 'eNMA' object: 36.9 Mb


  |
  |                                                                      |   0%
  |
  |=====                                                                 |   8%
  |
  |===========                                                           |  15%
```

```
  |
  |================                                                      |  23%
  |
  |======================                                                |  31%
  |
  |============================                                          |  38%
  |
  |================================                                      |  46%
  |
  |====================================                                  |  54%
  |
  |========================================                              |  62%
  |
  |============================================                          |  69%
  |
  |================================================                      |  77%
  |
  |============================================================          |  85%
  |
  |====================================================================  |  92%
  |
  |======================================================================| 100%
```

```
plot(modes, pdbs, col = grps.rd)
```

```
Extracting SSE from pdbs$sse attribute
```



Q14. What do you note about this plot? Are the black and colored lines similar or different? Where do you think they differ most and why?

The black lines are different from the colored lines. They differ the most near #30 - #60 residue and #120 to #160 residue. The differences show 2 distinct confrontational states for Adk. They differ by displacement of 2 nucleotide-binding site reigons that have distinct flexibilities once the nucleotide