Lab9

Chan-yu Kuo

Main structure storing biomolecule is PDB.

Q1. We need to obtain Experimental method and molecular type statistics

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

172654 by X-ray, 14105 by EM, total is 200988 for protein 85.9 % for X-ray and 7.02% for EM

```
pdb.stat<- read.csv('PDB.csv')
## substitude , with "", change the string into numbers
x_array<-as.numeric(gsub(",","",pdb.stat$X.ray))
EM_array<-as.numeric(gsub(",","",pdb.stat$EM))
Total_array<-as.numeric(gsub(",","",pdb.stat$Total))
xray_total<-sum(x_array)
xray_total
```

```
[1] 172654
```

```
EM_total<-sum(EM_array)
EM_total
```

```
[1] 14105
```

```
Total_total<-sum(Total_array)
Total_total
```

```
[1] 200988
```

```
round(xray_total/Total_total *100,digits=2)
```

```
[1] 85.9
```

```
round(EM_total/Total_total*100 ,digits=2)
```

```
[1] 7.02
```

```
pdb.stat
```

```
           Molecular.Type   X.ray    EM    NMR Multiple.methods Neutron Other
1          Protein (only) 152,809 9,421 12,117              191      72    32
2 Protein/Oligosaccharide   9,008 1,654     32                7       1     0
3              Protein/NA   8,061 2,944    281                6       0     0
4     Nucleic acid (only)   2,602    77  1,433               12       2     1
5                   Other     163     9     31                0       0     0
6  Oligosaccharide (only)      11     0      6                1       0     4
    Total
1 174,642
2  10,702
3  11,292
4   4,127
5     203
6      22
```

Question 2: What proportion of structures in the PDB are protein? protein_only has 174642, total has 200988 The portion is about 86.89

```
protein_only<-Total_array[1]# this is protein
protein_only
```

```
[1] 174642
```

```
sum(Total_array[1:3])
```

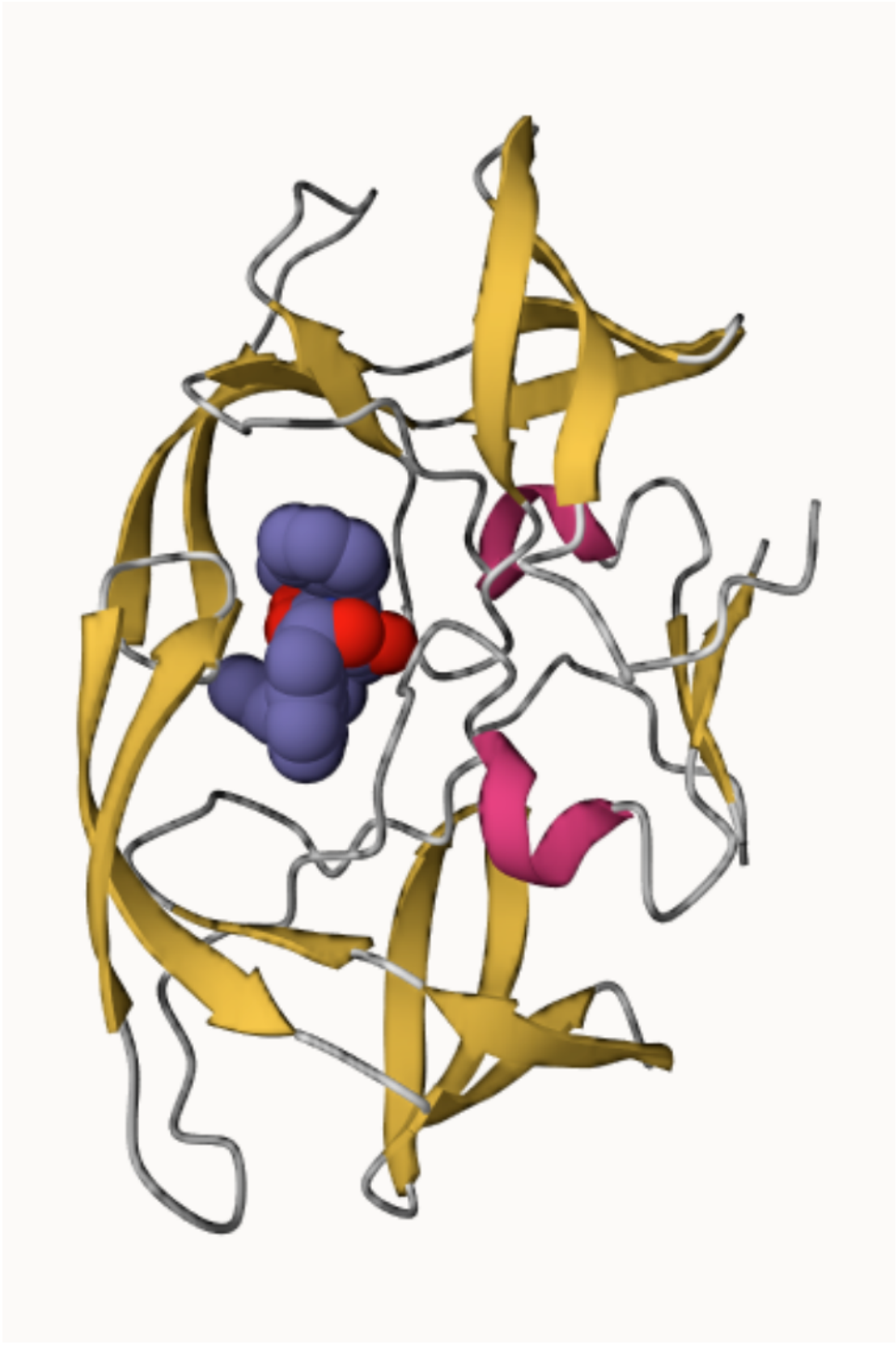
```
[1] 196636
```

```
round(protein_only/sum(Total_array[1:3])*100,2)
```

```
[1] 88.81
```

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? 200,988 structures

Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure? Because the resolution is 2A. Hydrogen is too small to be detected. 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 The residue number is 308. It has 4 h-bond Q6 image from molstar



How many amino acid residues are there in this pdb object? 198 Name one of the two non-protein residues? HOH How many protein chains are in this structure? 2

```
library(bio3d)
pdb <- read.pdb("1hsg")
```

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

```
head(pdb$atom$resid[1])
```

```
[1] "PRO"
```

```
attributes(pdb)
```

```
$names
[1] "atom"   "xyz"    "seqres" "helix"  "sheet"  "calpha" "remark" "call"

$class
[1] "pdb" "sse"
```

Performing Normal mode analysis to preduct protein flexibitlity and potential functional motions.

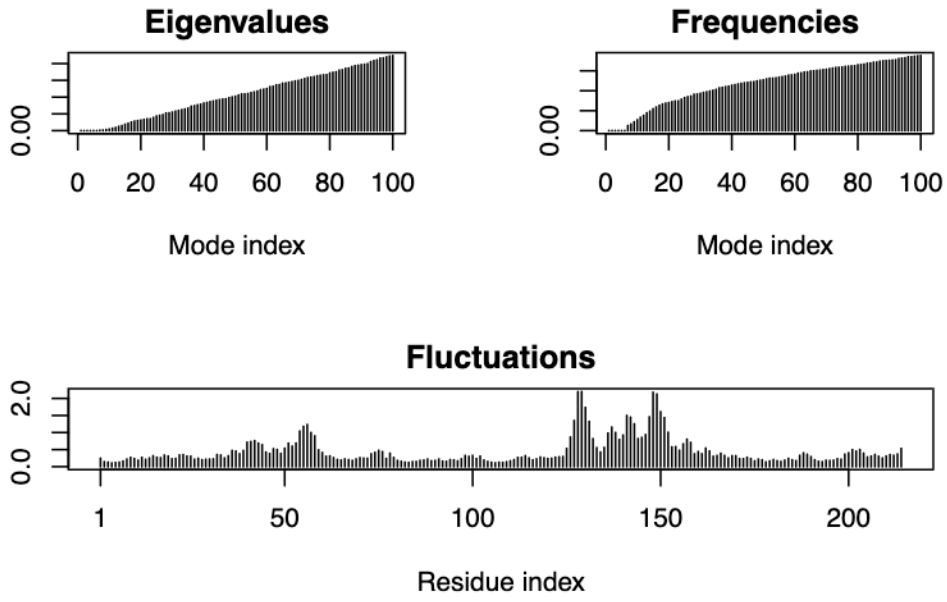
```
adk <- read.pdb("6s36")
```

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

```
# Perform flexiblity prediction
m <- nma(adk)
```

```
 Building Hessian...        Done in 0.021 seconds.
 Diagonalizing Hessian...   Done in 0.445 seconds.
```

```
plot(m)
```



This file e will be import into pdb to see the movement

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

Comparitive Structure Analysis of Adenylate Kinase

```
# Install packages in the R console NOT your Rmd/Quarto file

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

#BiocManager::install("msa")
#devtools::install_bitbucket("Grantlab/bio3d-view")
```

q10: Which of the packages above is found only on BioConductor and not CRAN? MSA q11 Which of the above packages is not found on BioConductor or CRAN?: bio3d-view Functions from the devtools package can be used to install packages from GitHub and BitBucket True

```
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? 214 aa

```
  #Blast or hmmer search, then we save the RDs file so we can access next time
  b <- blast.pdb(aa)
```

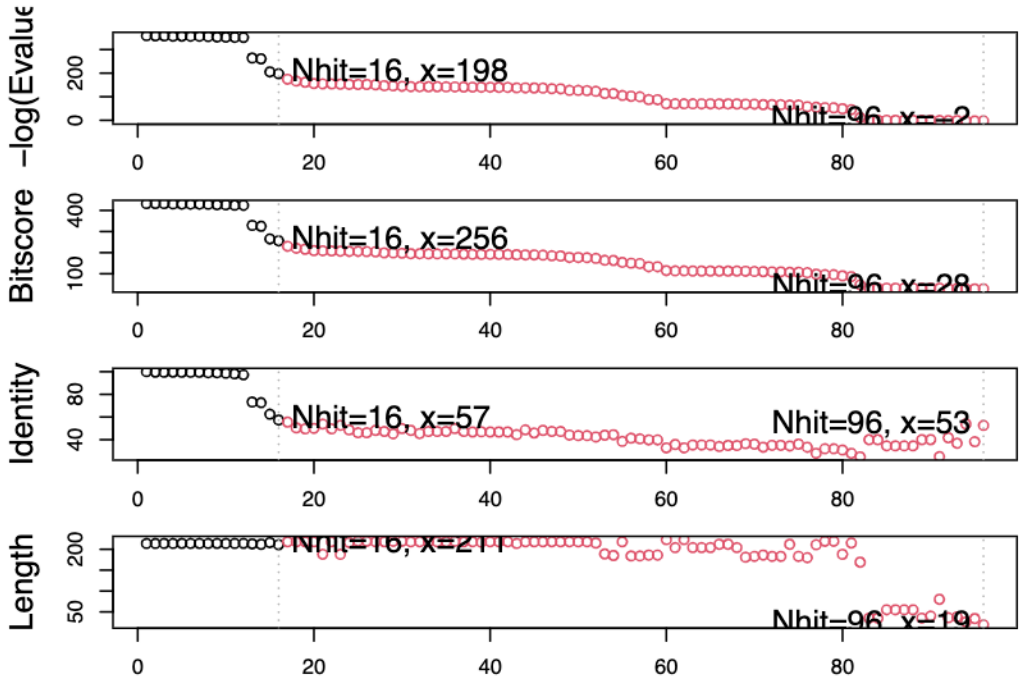
```
 Searching ... please wait (updates every 5 seconds) RID = YKEWRAJC016
 .
 Reporting 96 hits
```

```
saveRDS(b, file = "blast_1ake_A.RDS")
```

```
b <- readRDS("blast_1ake_A.RDS")
hits <- plot(b)
```

```
 * Possible cutoff values:    197 -3
          Yielding Nhits:    16 96

 * Chosen cutoff value of:    197
          Yielding Nhits:    16
```




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

```
[1] "1AKE_A" "4X8M_A" "6S36_A" "6RZE_A" "4X8H_A" "3HPR_A"
```

```
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.gz exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8M.pdb.gz exists. Skipping download

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

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

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4X8H.pdb.gz exists. Skipping download

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

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

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

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

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

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

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

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

```
Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/4NP6.pdb.gz exists. Skipping download

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

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


  |
  |                                                                      |   0%
  |
  |====                                                                  |   6%
  |
  |=========                                                             |  12%
  |
  |=============                                                         |  19%
  |
  |==================                                                    |  25%
  |
  |======================                                                |  31%
  |
  |==========================                                            |  38%
  |
  |==============================                                        |  44%
  |
  |==================================                                    |  50%
  |
  |======================================                                |  56%
  |
  |==========================================                            |  62%
  |
  |==============================================                        |  69%
  |
  |==================================================                    |  75%
  |
  |======================================================                |  81%
  |
  |==========================================================            |  88%
  |
  |==================================================================    |  94%
```

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

```
# Align PDBs downloaded before
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

```
Reading PDB files:
pdbs/split_chain/1AKE_A.pdb
pdbs/split_chain/4X8M_A.pdb
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/6RZE_A.pdb
pdbs/split_chain/4X8H_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/4NP6_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/4X8M_A.pdb
pdb/seq: 3   name: pdbs/split_chain/6S36_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 4   name: pdbs/split_chain/6RZE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
```

```
pdb/seq: 5   name: pdbs/split_chain/4X8H_A.pdb
pdb/seq: 6   name: pdbs/split_chain/3HPR_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 7   name: pdbs/split_chain/1E4V_A.pdb
pdb/seq: 8   name: pdbs/split_chain/5EJE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 9   name: pdbs/split_chain/1E4Y_A.pdb
pdb/seq: 10   name: pdbs/split_chain/3X2S_A.pdb
pdb/seq: 11   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 12   name: pdbs/split_chain/6HAM_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 13   name: pdbs/split_chain/4K46_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 14   name: pdbs/split_chain/4NP6_A.pdb
pdb/seq: 15   name: pdbs/split_chain/3GMT_A.pdb
pdb/seq: 16   name: pdbs/split_chain/4PZL_A.pdb
```

```
  head(pdbs$id)
```

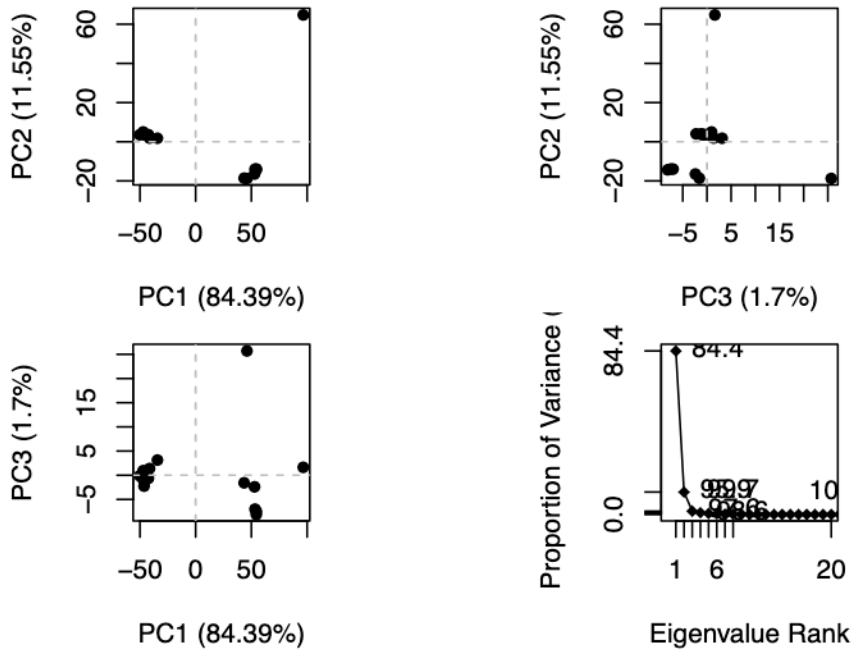
```
[1] "pdbs/split_chain/1AKE_A.pdb" "pdbs/split_chain/4X8M_A.pdb"
[3] "pdbs/split_chain/6S36_A.pdb" "pdbs/split_chain/6RZE_A.pdb"
[5] "pdbs/split_chain/4X8H_A.pdb" "pdbs/split_chain/3HPR_A.pdb"
```

```
ids <- basename.pdb(pdbs$id)
```

```
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] "Vibrio cholerae O1 biovar El Tor str. N16961"
[7] "Burkholderia pseudomallei 1710b"
[8] "Francisella tularensis subsp. tularensis SCHU S4"
```

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



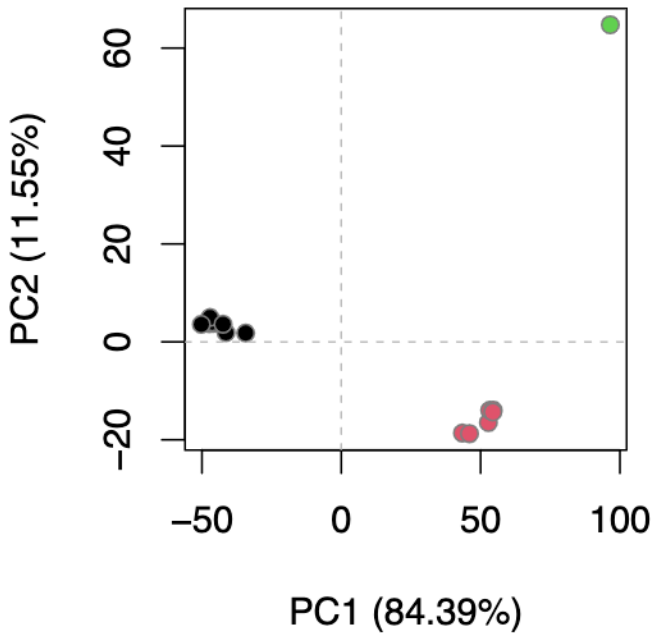
each dot represent one PDB structure, and the RMSD is the value of the distance between each pdb structure pair.

```
# 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)
```




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