Class 10: Structural Bioinformatics

Andrew Sue

What is in the PDB database?

PDB is a protein data bank.

Download a CSV file with current composition data from: https://www.rcsb.org/stats/summary

```
PDBstats <- read.csv("PDB_Stats.csv", row.names = 1 )
head(PDBstats)
```

```
                          X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)          161,663 12,592 12,337              200      74    32
Protein/Oligosaccharide   9,348  2,167     34                8       2     0
Protein/NA                8,404  3,924    286                7       0     0
Nucleic acid (only)       2,758    125  1,477               14       3     1
Other                       164      9     33                0       0     0
Oligosaccharide (only)       11      0      6                1       0     4
                          Total
Protein (only)          186,898
Protein/Oligosaccharide  11,559
Protein/NA               12,621
Nucleic acid (only)       4,378
Other                       206
Oligosaccharide (only)       22
```

There are commas within our data that make them characters rather than numbers. We must remove them in order to do math. Lets make a function that removes the commas, converts to numeric, and sums up the column.

```
PDBstats1 <- gsub(",", "", PDBstats)
```

```r
#Removes commma, converts to numeric, and adds column together
commasum <- function(x) {
  sum(as.numeric(gsub(",","",x)))
}
```

```r
commasum(PDBstats)
```

```
Warning in commasum(PDBstats): NAs introduced by coercion
```

```
[1] NA
```

I can `apply` this function to the entire table to get all the numbers you need.

```r
round(apply(PDBstats,2,commasum) / commasum(PDBstats$Total) *100, 2)
```

```
           X.ray               EM              NMR Multiple.methods 
           84.54             8.72             6.57             0.11 
         Neutron            Other            Total 
            0.04             0.02           100.00 
```

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

84.54% are X-Ray and 8.72% are EM.

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

186,898 are protein

```r
head(PDBstats)
```

```
                          X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)          161,663 12,592 12,337              200      74    32
Protein/Oligosaccharide   9,348  2,167     34                8       2     0
Protein/NA                8,404  3,924    286                7       0     0
Nucleic acid (only)       2,758    125  1,477               14       3     1
Other                       164      9     33                0       0     0
Oligosaccharide (only)       11      0      6                1       0     4
                          Total
Protein (only)          186,898
```

```
Protein/Oligosaccharide  11,559
Protein/NA               12,621
Nucleic acid (only)       4,378
Other                       206
Oligosaccharide (only)       22
```

> Q. How does the total number of protein structures in the PDB relate to total number of protein sequences in Uniprot?

```
186898 / 250322721 *100
```

```
[1] 0.07466282
```

Visualizing the HIV-1 Protease

> 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?

215,684 total, 66,858 in humans.

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

Because it only shows the oxygen. The hydrogen molecule is too small to visualize.

> 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 hydrogen atom is H308.

> 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.

Working with structures in R.

We will use the bio3d package for structural bioinformatics.

```
library(bio3d)
```

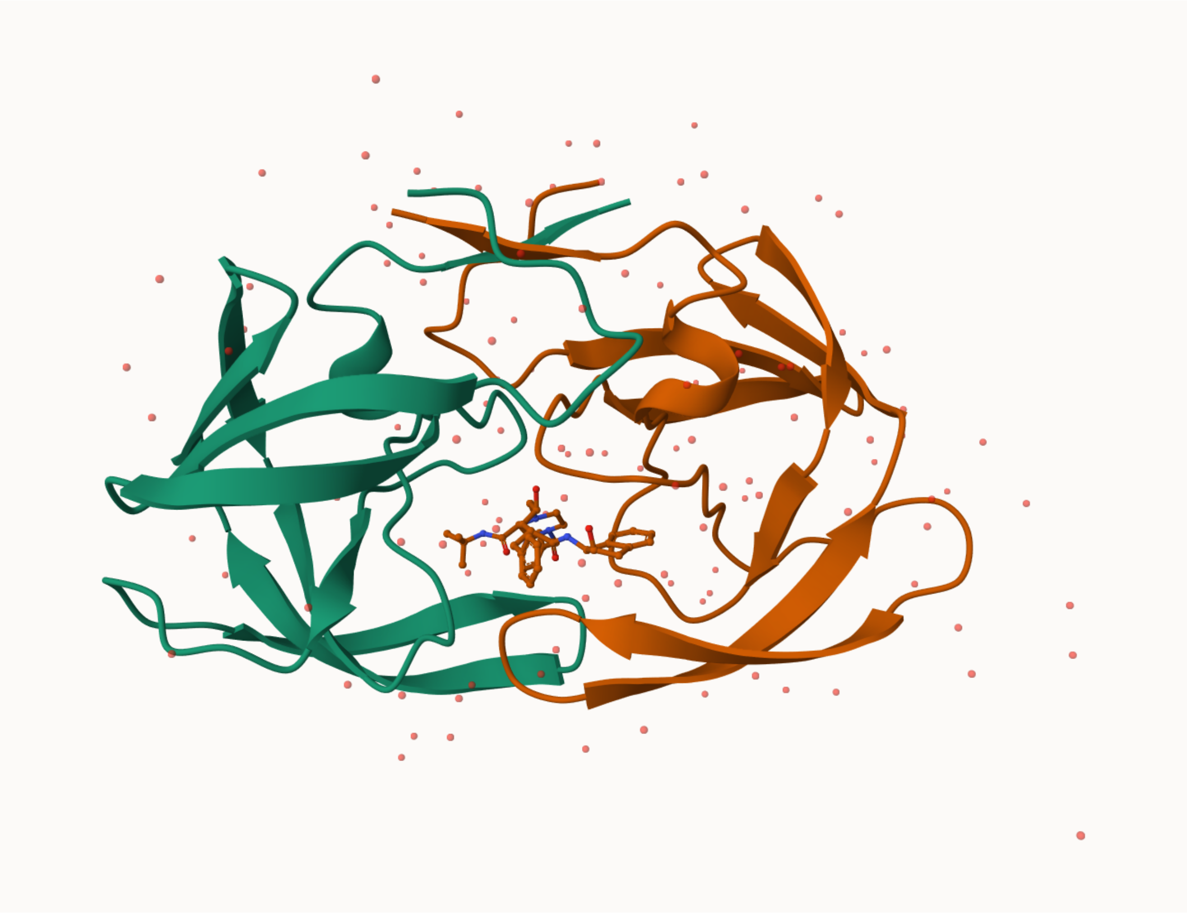


Figure 1: HIV-1 Protease

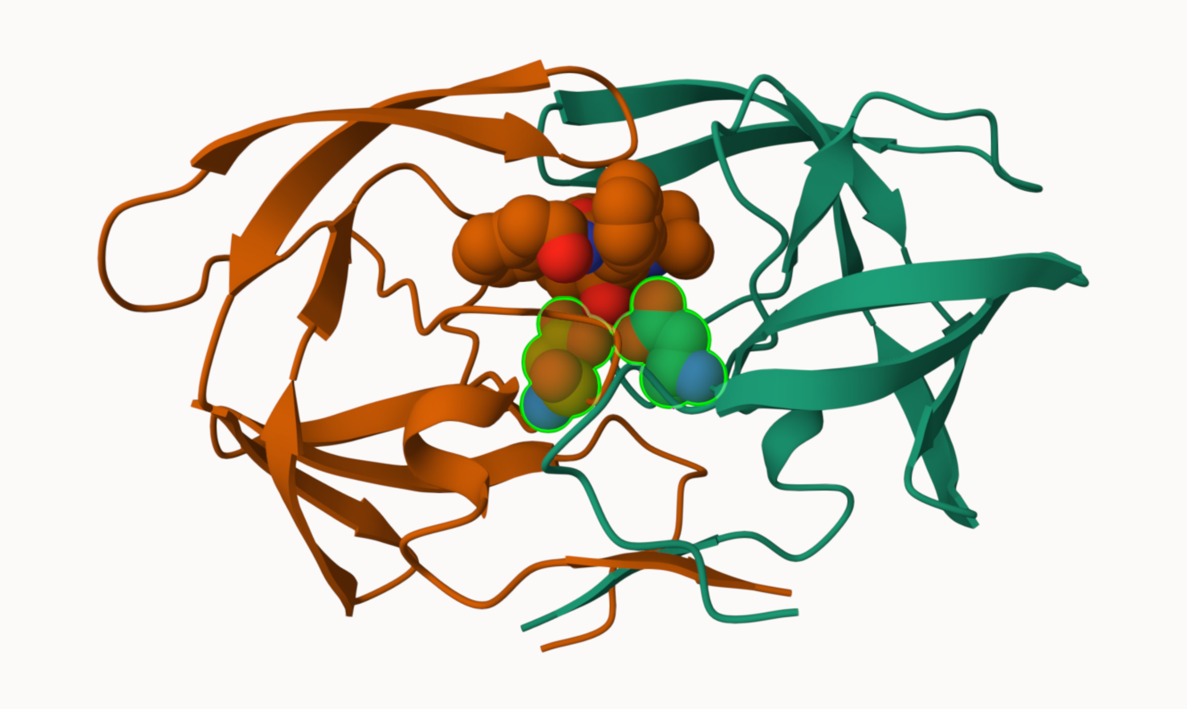


Figure 2: HIV-1 Protease with ASP residues selected

```r
hiv <- read.pdb("1hsg")
```

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

```r
hiv
```

```

 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?

198 amino acid residues > Q8: Name one of the two non-protein residues?

MK1 > Q9: How many protein chains are in this structure?

2 chains

```
head(hiv$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>
```

```
aa123(pdbseq(hiv)[25])
```

```
[1] "ASP"
```

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

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

```r
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
```

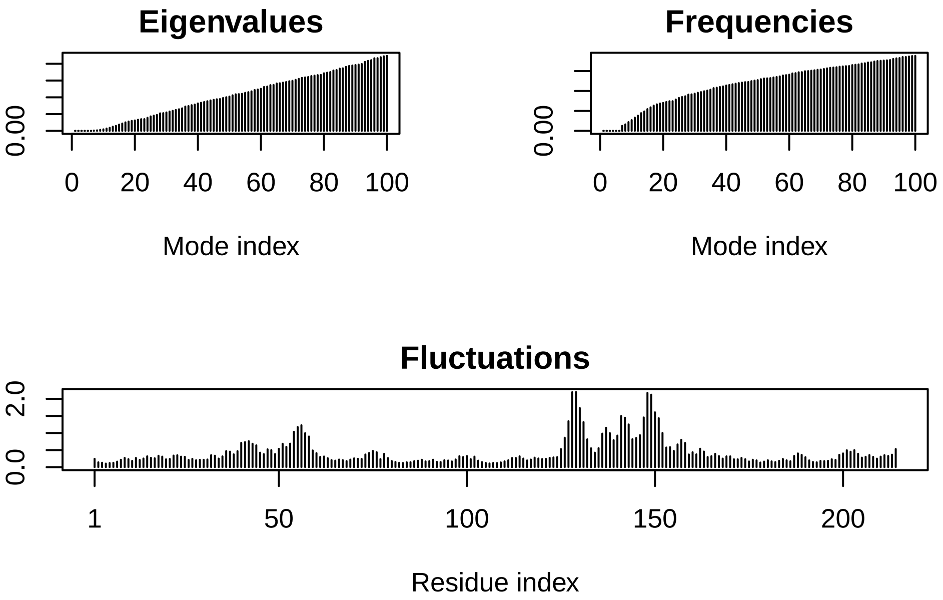
Normal mode analysis (NMA) a bioinformatics method to predict functional motions and large-scale structural changes.

In fluctuations, the peak areas are the flexible regions of proteins.

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

```
 Building Hessian...        Done in 0.011 seconds.
 Diagonalizing Hessian...   Done in 0.226 seconds.
```

```r
plot(m)
```



Make a movie of this predicted motion (a.k.a "trajectory")

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

Quick comparative analysis

Workflow: 1. Get protein code from PDB and read it in 2. Get the sequence. 3. Blast against PDB 4. Download all the HITS

Step 1: Extract sequence and run a blast search using `blast.pdb()`

```r
# s <- pdbseq(adk)
# blast <- blast.pdb(s)
# plot(blast)
```

```r
# hits<- plot(blast)
# hits
```

```r
# hits$pdb.id
```

Get the results from BLAST and download top hits.

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

When you view them on **MolStar**, they are all mixed and hard to decipher because they are all on different reference frames as they are all individual photos. So we must fix the reference frame to superimpose them on top of each other.

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

PCA of all structures

```
# Perform PCA
# pc.xray <- pca(pdbs)
# plot(pc.xray)
```

```
# mktrj(pc.xray, file = "pca_movie.pdb")
```

Comparative analysis of Adenylate Kinase

> 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 > Q12. True or False? 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.
```

```r
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

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

```r
# Plot a summary of search results
# hits <- plot(b)
```

```r
hits <- NULL
hits$pdb.id <- c('1AKE_A','6S36_A','6RZE_A','3HPR_A','1E4V_A','5EJE_A','1E4Y_A','3X2S_A','
# List out some 'top hits'
head(hits$pdb.id)
```

```
[1] "1AKE_A" "6S36_A" "6RZE_A" "3HPR_A" "1E4V_A" "5EJE_A"
```

```
# 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.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/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/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%
  |
  |=====                                                                 |   8%
  |
  |===========                                                           |  15%
  |
  |================                                                      |  23%
  |
  |======================                                                |  31%
  |
  |===========================                                           |  38%
  |
  |================================                                      |  46%
  |
  |======================================                                |  54%
  |
  |===========================================                           |  62%
  |
  |================================================                      |  69%
  |
  |======================================================                |  77%
  |
  |===========================================================           |  85%
  |
  |=================================================================     |  92%
  |
  |======================================================================| 100%
```

Align and super impose 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
```

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

# Draw schematic alignment
# plot(pdbs, labels=ids, par(mar=c(1,1,1,1)))
```

```
library(bio3d.view)
library(rgl)
```

```
Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display

Warning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'.
```

```
view.pdbs(pdbs)
```

#Annotated PDB structures

The function `pdb.annotate()` provides a convenient way of annotating the PDB files we have collected.

```
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"
```

```
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 (ADK)
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 (ADK)
1E4Y_A       1.85 Adenylate kinase                      Adenylate kinase (ADK)
3X2S_A       2.80             <NA>                      Adenylate kinase (ADK)
6HAP_A       2.70             <NA> Adenylate kinase, active site lid (ADK_lid)
6HAM_A       2.55             <NA> Adenylate kinase, active site lid (ADK_lid)
4K46_A       2.01             <NA> Adenylate kinase, active site lid (ADK_lid)
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
```

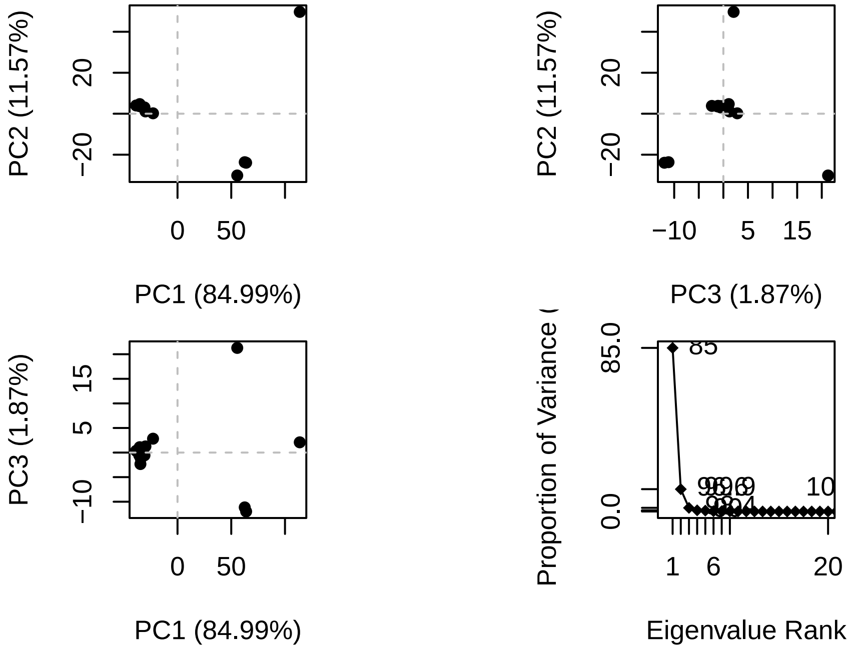
```
3GMT_A          SO4 (2)
4PZL_A       CA,FMT,GOL
                                                                             ligandName
1AKE_A                                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
6S36_A                                    CHLORIDE ION (3),SODIUM ION,MAGNESIUM 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 INHIB
6S36_A
6RZE_A
3HPR_A
1E4V_A
5EJE_A                                                                                  Crys
1E4Y_A
3X2S_A
6HAP_A
6HAM_A
4K46_A
3GMT_A
```

```
4PZL_A                                                                              The crys
                                                     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
```

Principal component analysis

PCA PCA can be performed on the structural ensemble (stored in the pdbs object) with the function `pca.xyz()`, or more simply `pca()`.

```
# Perform PCA
pc.xray <- pca(pdbs)
plot(pc.xray)
```

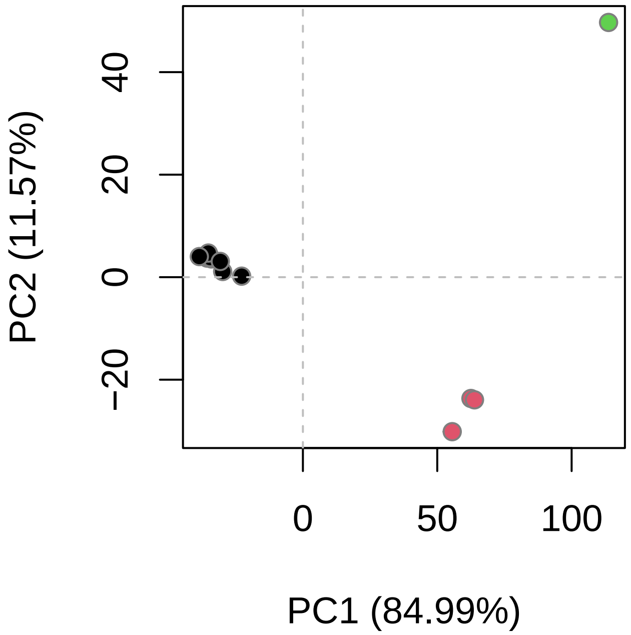


The function `rmsd()` calculates all pairwise RMSD values in a dataset.

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

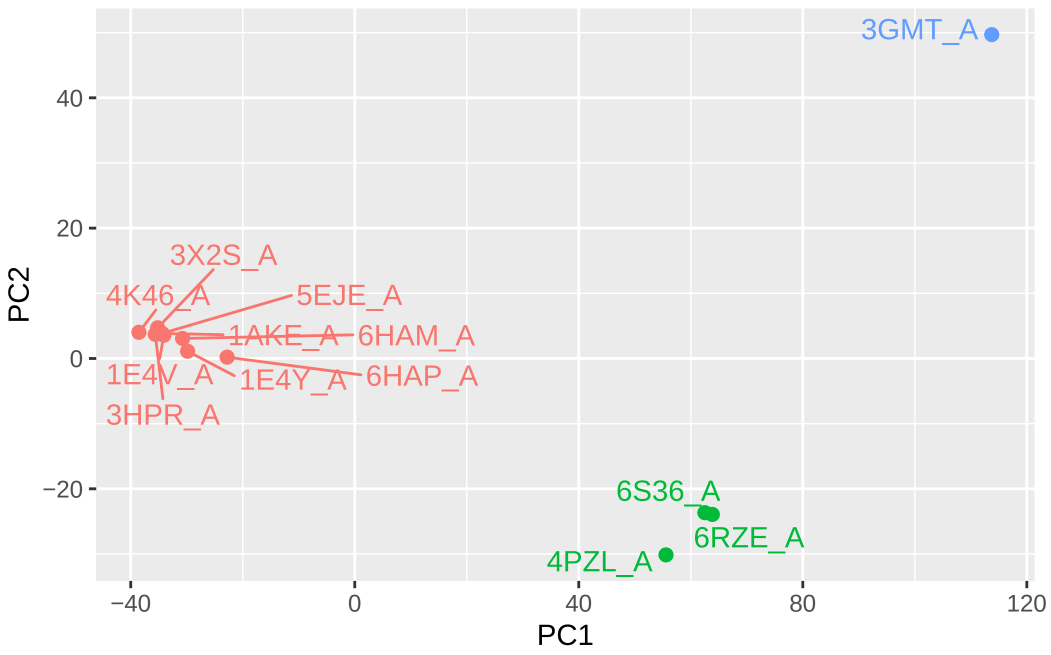
```
Warning in rmsd(pdbs): No indices provided, using the 204 non NA positions
```

```
# 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")
```

```
#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
```

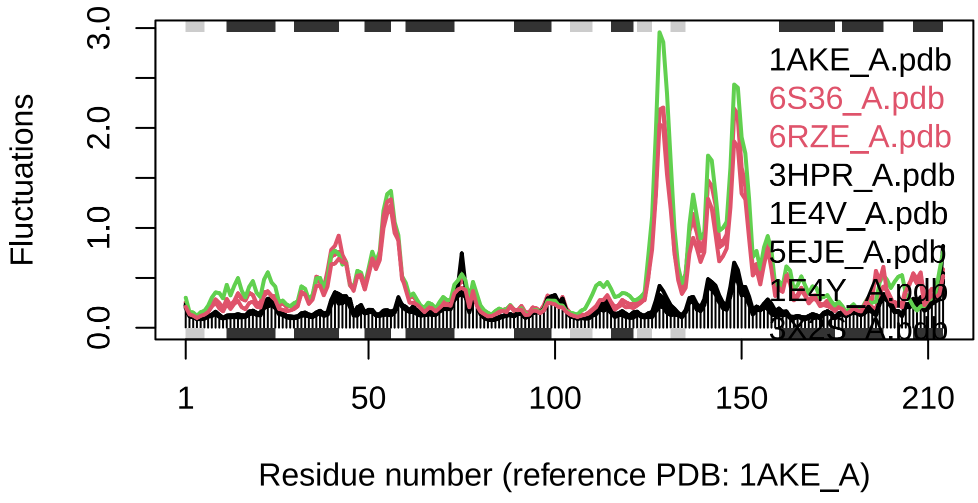


```
# NMA of all structures
modes <- nma(pdbs)
```

```
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%
```

```r
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?

These are the flexible structures of the proteins seen at the peaks with their fluctuation graphs super imposed. THe color has the ones that most closely cluster together representing different states of the structure.