Class 10: Structural Bioinformatics Pt.1

AUTHOR
Vivian Chau (A16913056)

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. PDB > Analyze > PDB Statistics > By Exp method and molecular type (download CSV file)

```
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.

```
pdbstats$X.ray
```

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

The comma in these numbers is causing them to be read as character rather than numeric. I can fix this by replacing "," for nothing "" with the `sub()` function:

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

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

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.
```

```
pdbstats
```

```
# A tibble: 6 × 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 all lower case 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 × 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
```

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

```
[1] 191374
```

Total number of structures:

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

```
[1] 231029
```

Percent of X-ray structures:

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

```
[1] 82.83549
```

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

```
[1] 24836
```

Percent of EM structures:

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

```
[1] 10.75017
```

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

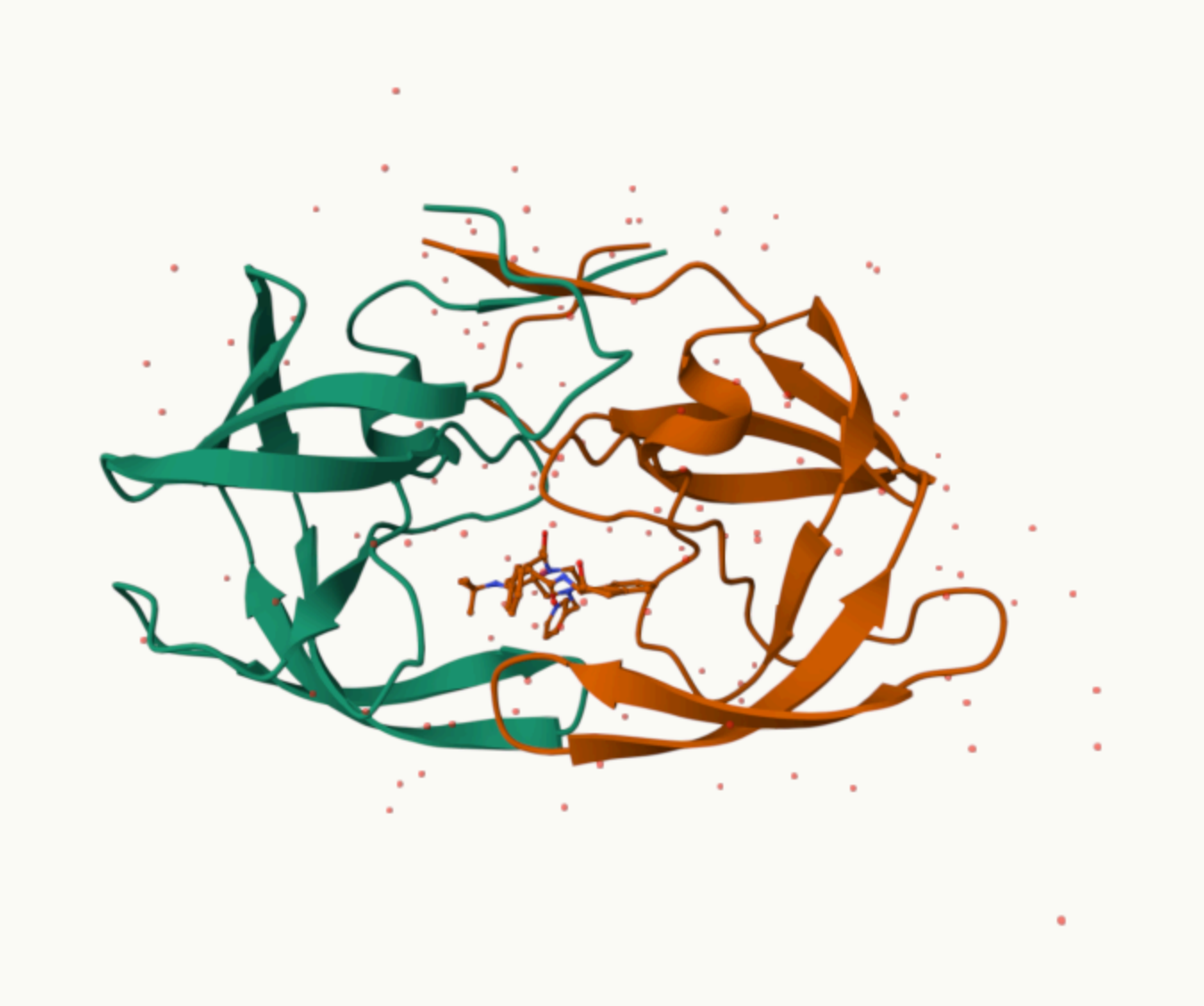
```
199236/sum(df$total)*100
```

```
[1] 86.23852
```

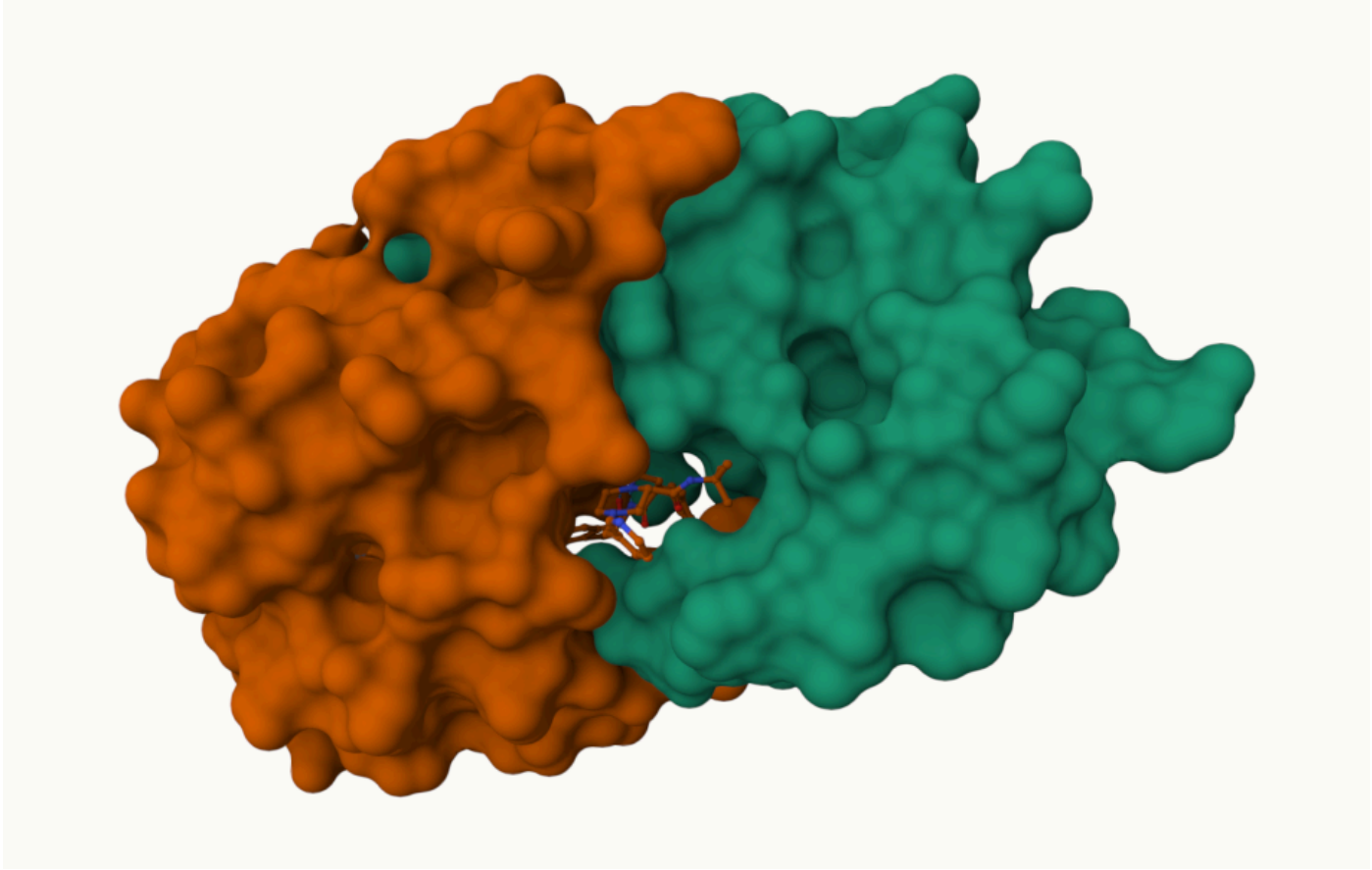
> Q3: Type HIV in the PDB website search box on the home page and determine how many HIV-1 protease structures are in the current PDB? 26, 725

Using Mol*

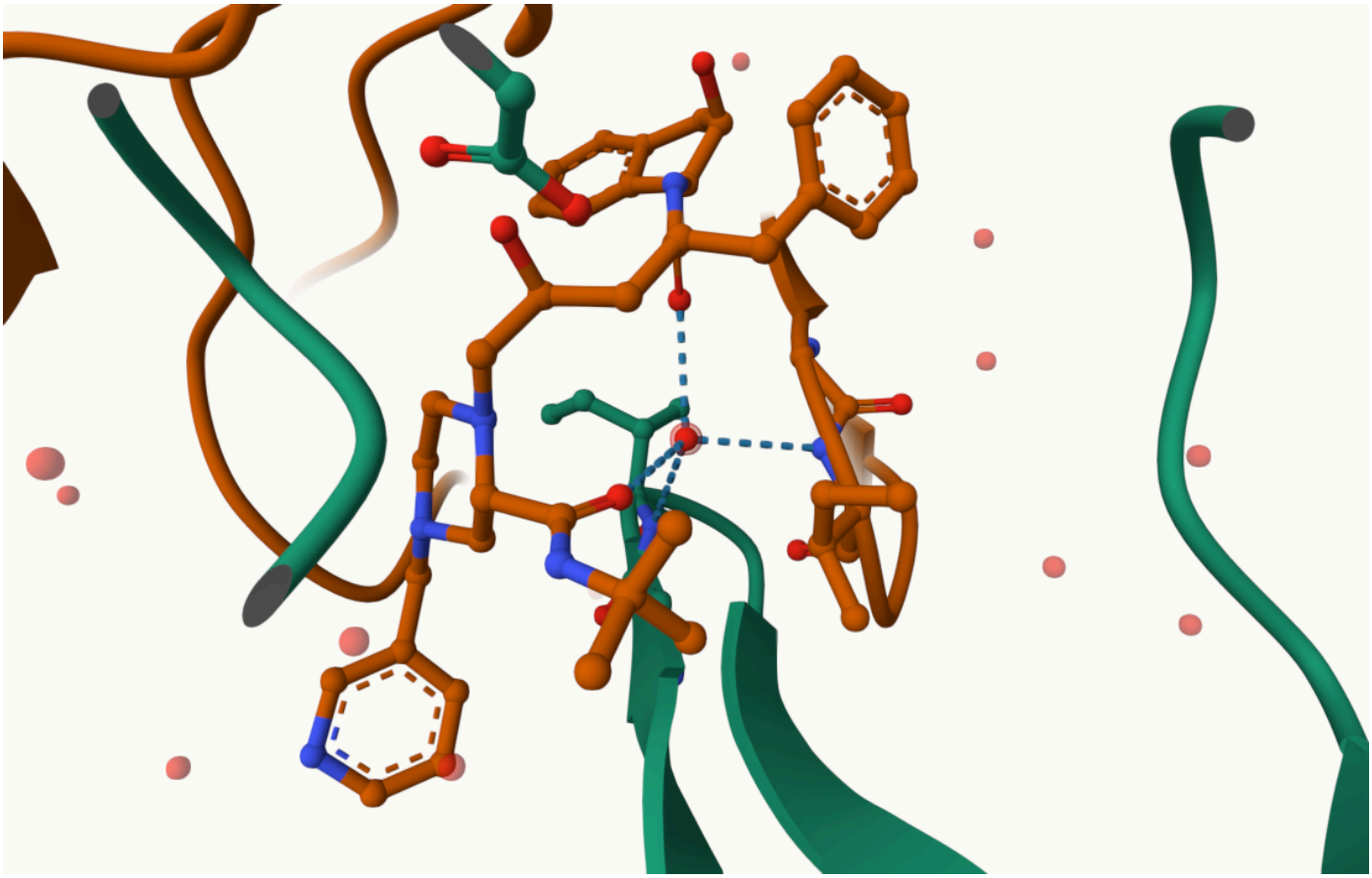
The main Mol* homepage at: https://molstar.org/viewer/ We can input our own PDB files or just give it a PDB database accession code (4 letter PDB code).



Molecular overview of 1HSG



Surface representation showing ligand binding

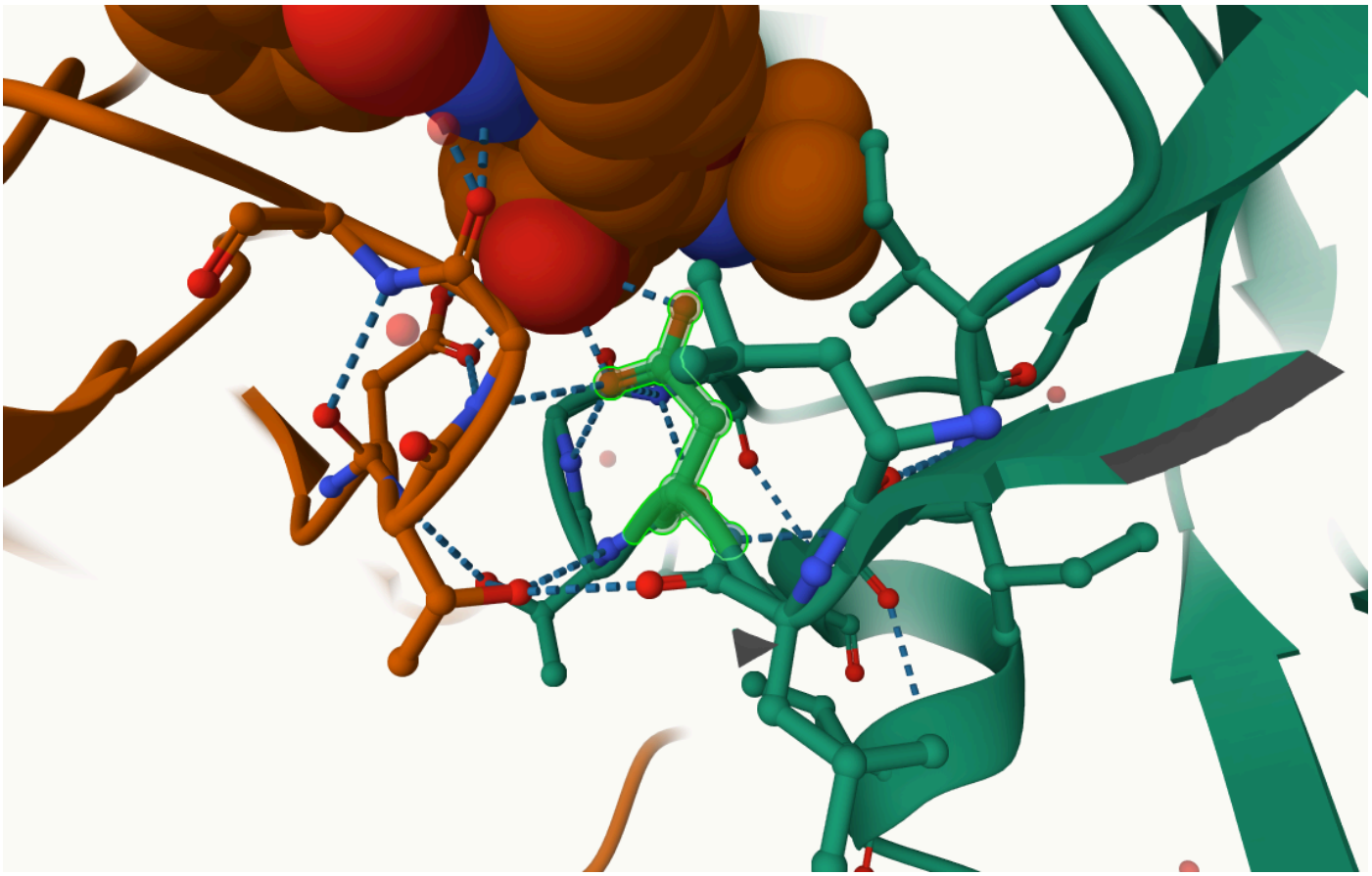


Binding site of HOH 308

> Q4: Water molecules normally have 3 atoms. Why do we see just one atom per water molecule in this structure? Using the ball-and-stick model, the oxygen is shown in greater detail, while the hydrogen atoms are represented smaller.

> 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 found in residue 308.

> 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.



HIV-1 Protease

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? There are 198 amino acid residues in this pdb object.

> Q8: Name one of the two non-protein residues? MK1

> Q9: How many protein chains are in this structure? 2 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"
```

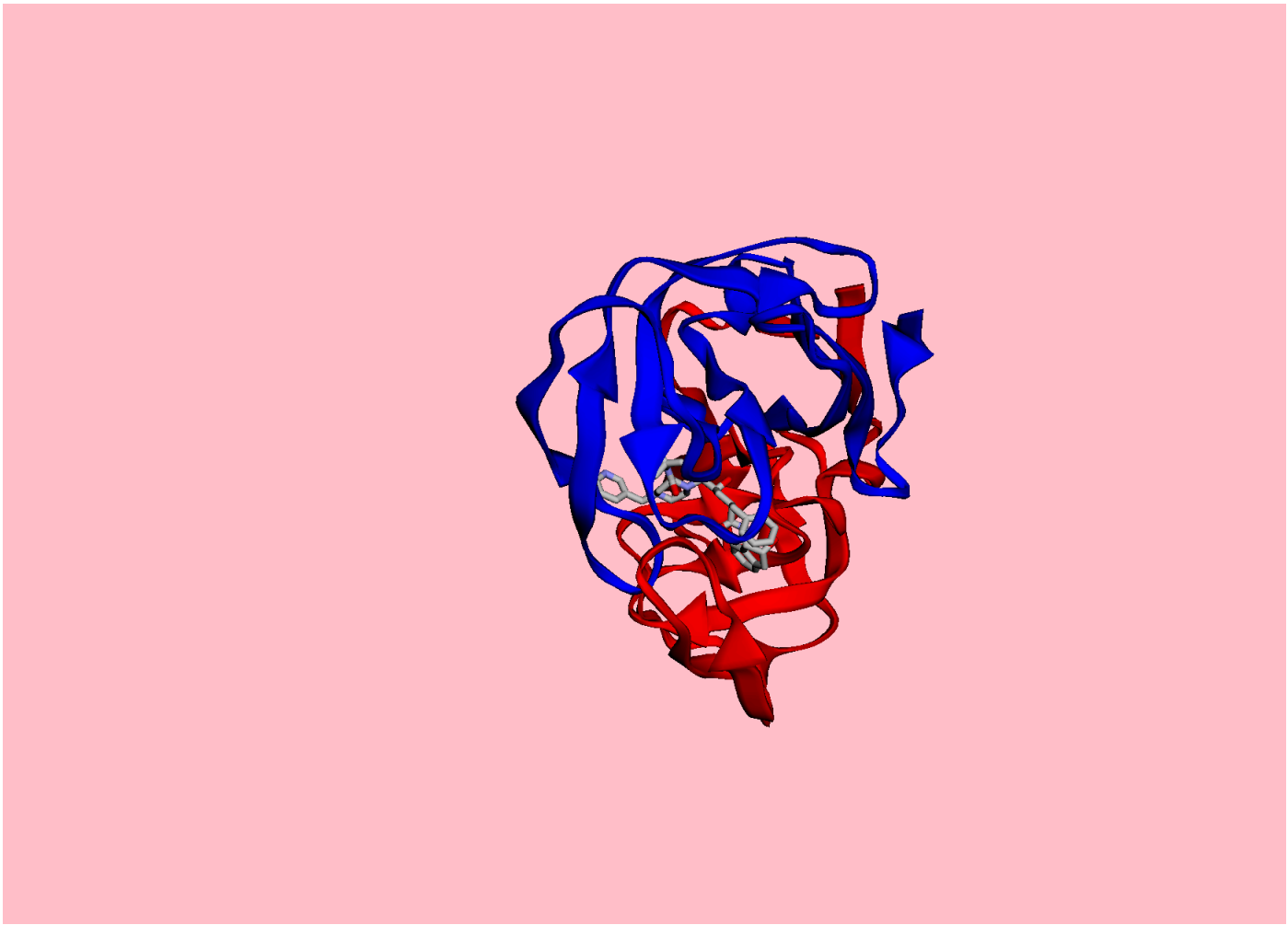
```
head(pdb$atom)
```

```
  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: It requires the **r3dmol** and **shiny** packages that we need to install.

```
library(r3dmol)
library(shiny)

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
```

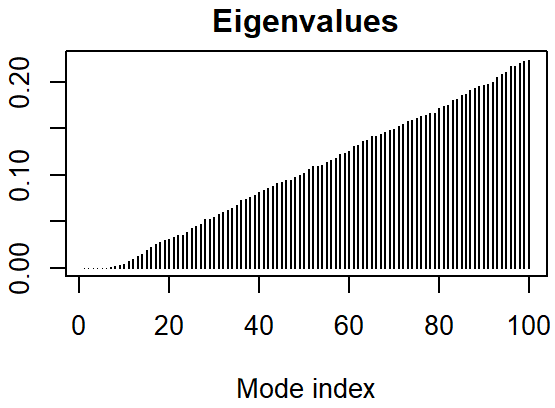
```
adk
```

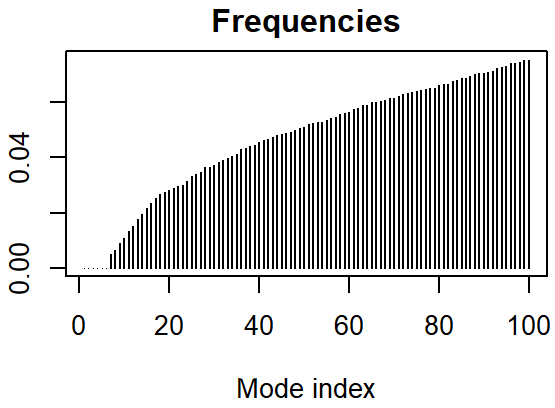
```
 Call:  read.pdb(file = "6s36")

   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
```

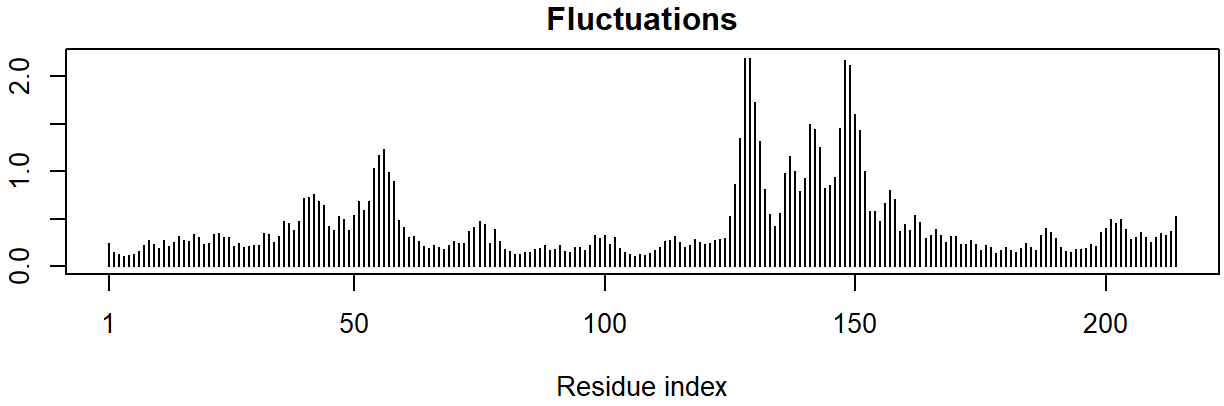
```
m <-nma(adk)
```

```
 Building Hessian...        Done in 0.05 seconds.
 Diagonalizing Hessian...   Done in 0.31 seconds.
```

```
plot(m)
```







Write out a trajectory of the predicted molecular motion:

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

Comparative structure analysis of Adenylate Kinase

```
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
```

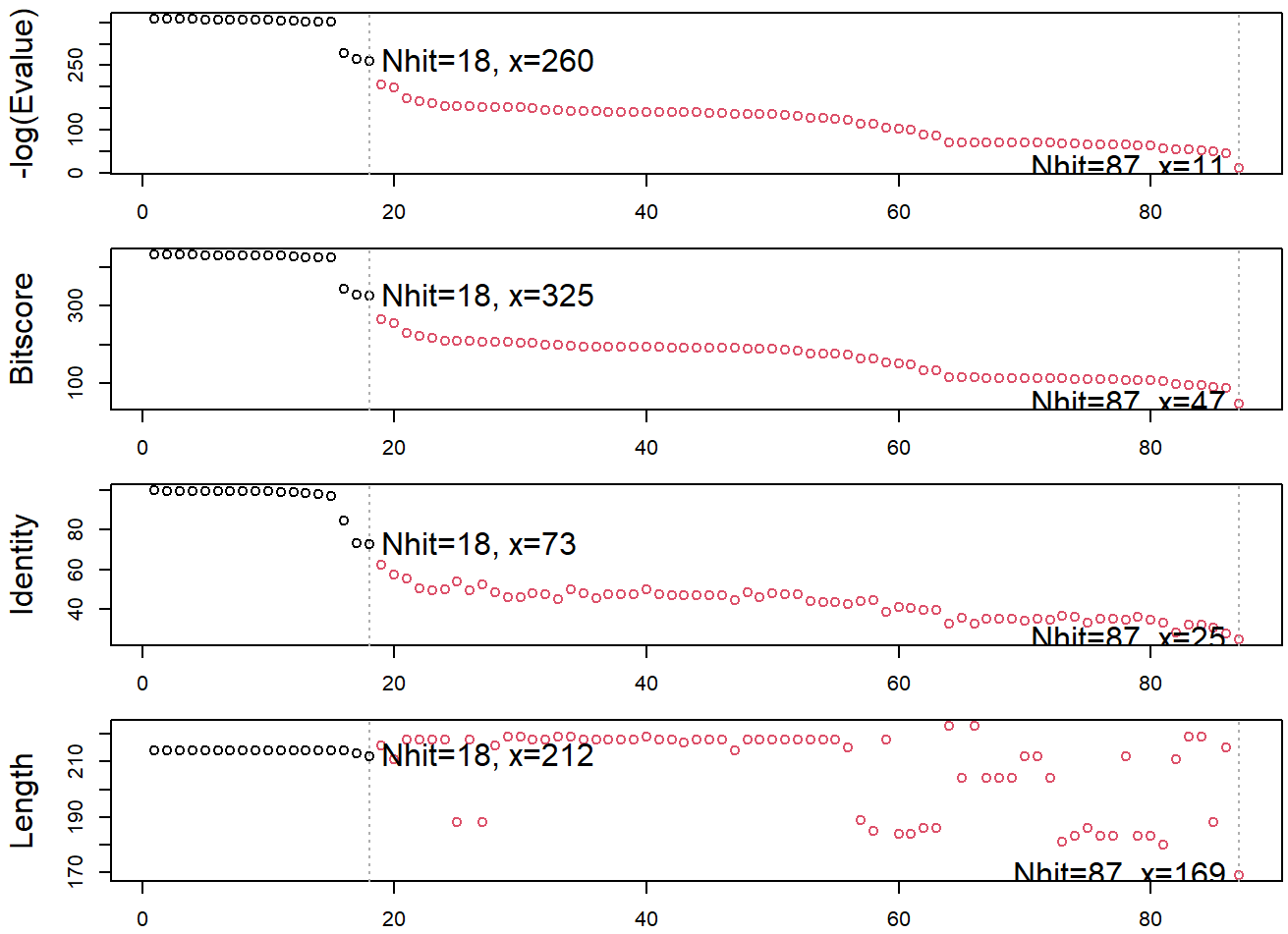
```
            61        .         .         .         .         .         120

           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
```

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

```
 Searching ... please wait (updates every 5 seconds) RID = UMDJ725S016
 .
 Reporting 87 hits
```

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

```
  * Possible cutoff values:    260 11
            Yielding Nhits:    18 87

  * Chosen cutoff value of:    260
            Yielding Nhits:    18
```




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

```
[1] "1AKE_A" "8BQF_A" "4X8M_A" "6S36_A" "8Q2B_A" "8RJ9_A"
```

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

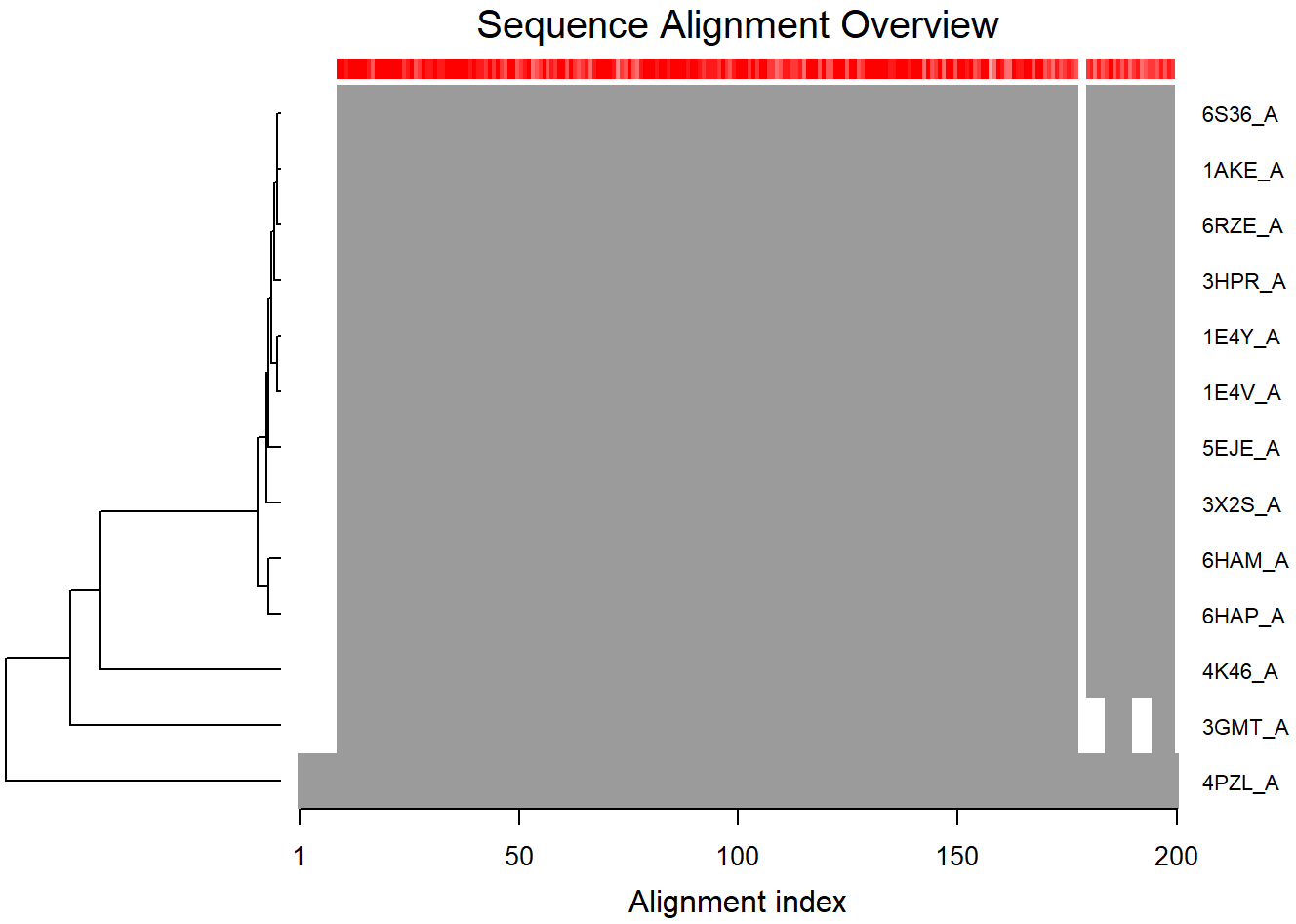
```
  |
  |=======================================================               |  77%
  |
  |============================================================          |  85%
  |
  |================================================================      |  92%
  |
  |======================================================================| 100%
```

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

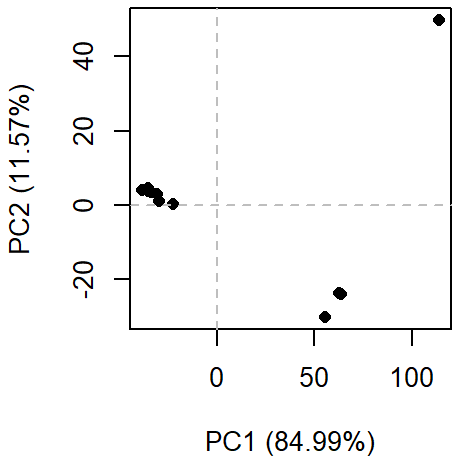
```
Reading PDB files:
pdbs/split_chain/1AKE_A.pdb
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/6RZE_A.pdb
pdbs/split_chain/3HPR_A.pdb
pdbs/split_chain/1E4V_A.pdb
pdbs/split_chain/5EJE_A.pdb
pdbs/split_chain/1E4Y_A.pdb
pdbs/split_chain/3X2S_A.pdb
pdbs/split_chain/6HAP_A.pdb
pdbs/split_chain/6HAM_A.pdb
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
```

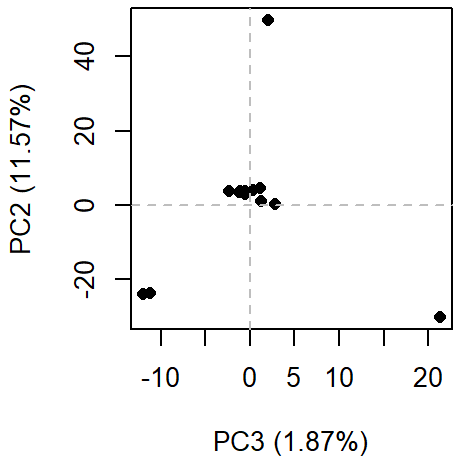
```
pdb/seq: 9   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 10   name: pdbs/split_chain/6HAM_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 11   name: pdbs/split_chain/4K46_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 12   name: pdbs/split_chain/3GMT_A.pdb
pdb/seq: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

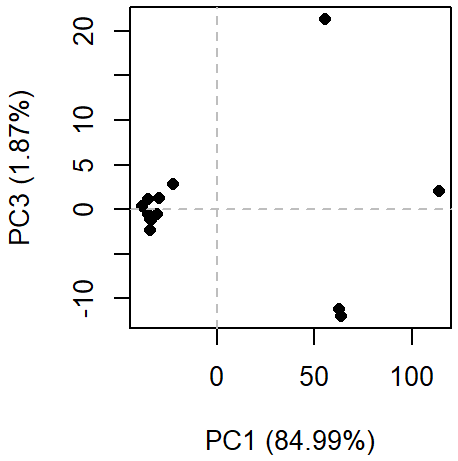
```r
# Vector containing PDB codes for figure axis
ids <- basename.pdb(pdbs$id)

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

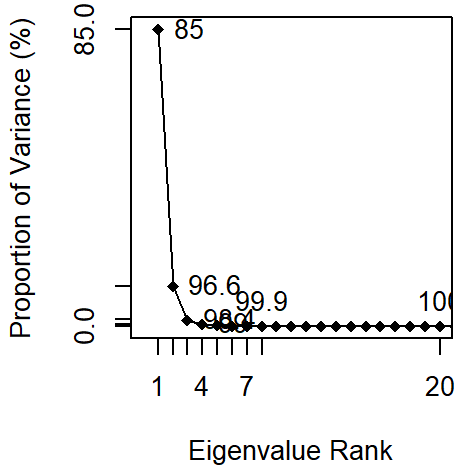



```r
# Perform PCA
pc.xray <- pca(pdbs)
plot(pc.xray)
```





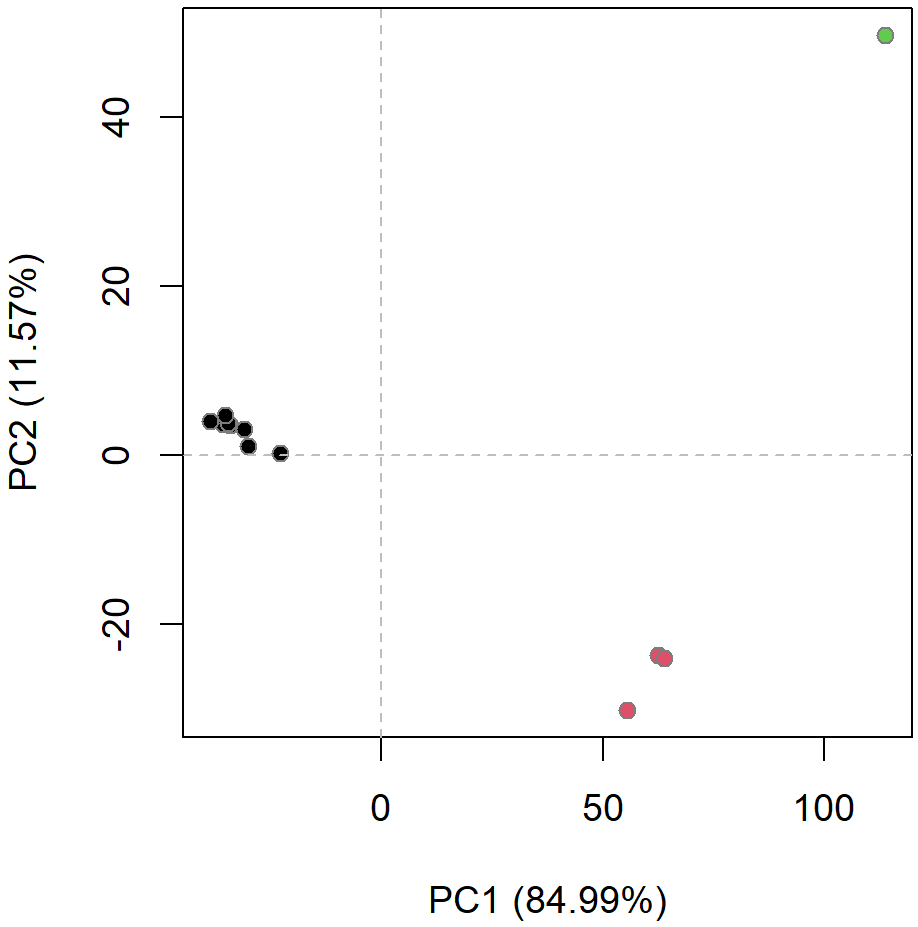





```
# Calculate RMSD
rd <- rmsd(pdbs)
```

```
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)
```


PC2 (11.57%)
40
20
0
-20
0
50
100
PC1 (84.99%)