Lecture9: Structural Bioinformatics

Longmei Zhang A17012012

Introduction to the RCSB Protein Data Bank (PDB)

```
table1 <- read.csv("Data Export Summary.csv")
table1
```

```
           Molecular.Type   X.ray     EM    NMR Multiple.methods Neutron Other
1          Protein (only) 167,192 15,572 12,529              208      77    32
2 Protein/Oligosaccharide   9,639  2,635     34                8       2     0
3              Protein/NA   8,730  4,697    286                7       0     0
4     Nucleic acid (only)   2,869    137  1,507               14       3     1
5                   Other     170     10     33                0       0     0
6  Oligosaccharide (only)      11      0      6                1       0     4
    Total
1 195,610
2  12,318
3  13,720
4   4,531
5     213
6      22
```

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

83.30% of the structures are solved by X-ray, and 10.18% of the structures are solved by EM.

method 1: conversion Create a function to remove the comma and turn the characters into numbers

```
num_convert <- function(col){
  as.numeric(sub(",", "", col))
}
```

```r
# X-ray
sum(num_convert(table1[, "X.ray"]))/ sum(num_convert(table1[, "Total"]))
```

```
[1] 0.8330359
```

```r
# EM
sum(num_convert(table1[, "EM"]))/ sum(num_convert(table1[, "Total"]))
```

```
[1] 0.1018091
```

method 2: different import function

```r
library(readr)
table <- read_csv("Data Export Summary.csv", show_col_types = FALSE)

#convert the table to df and change the first column into row name
table <- as.data.frame(table)
rownames(table) <- table[,1]
table <- table[ , -1]
table
```

```
                        X-ray    EM   NMR Multiple methods Neutron Other
Protein (only)         167192 15572 12529               208      77    32
Protein/Oligosaccharide  9639  2635    34                 8       2     0
Protein/NA               8730  4697   286                 7       0     0
Nucleic acid (only)      2869   137  1507                14       3     1
Other                     170    10    33                 0       0     0
Oligosaccharide (only)     11     0     6                 1       0     4
                        Total
Protein (only)         195610
Protein/Oligosaccharide  12318
Protein/NA               13720
Nucleic acid (only)       4531
Other                      213
Oligosaccharide (only)      22
```

Calculate the percentages of structures solve by X-Ray:

```
sum(table$`X-ray`) / sum(table$Total)
```

```
[1] 0.8330359
```

Calculate the percentage of structures solve by Electron Microscopy:

```
sum(table$`EM`) / sum(table$Total)
```

```
[1] 0.1018091
```

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

around 86.39% of the structures are protein

```
table["Protein (only)", "Total"] * 100/ sum(table$Total)
```

```
[1] 86.39483
```

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

There are 5 HIV-1 protease structures in the current PDB

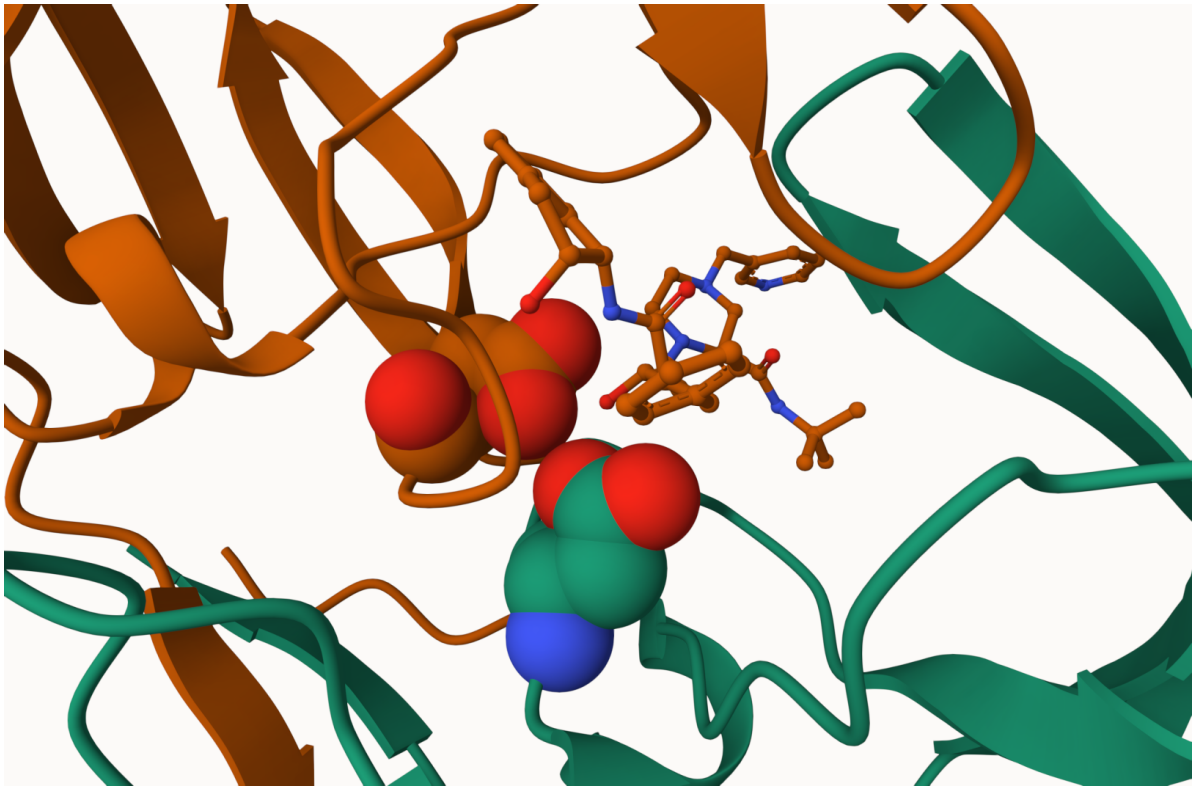
Visualizing the HIV-1 protease structure

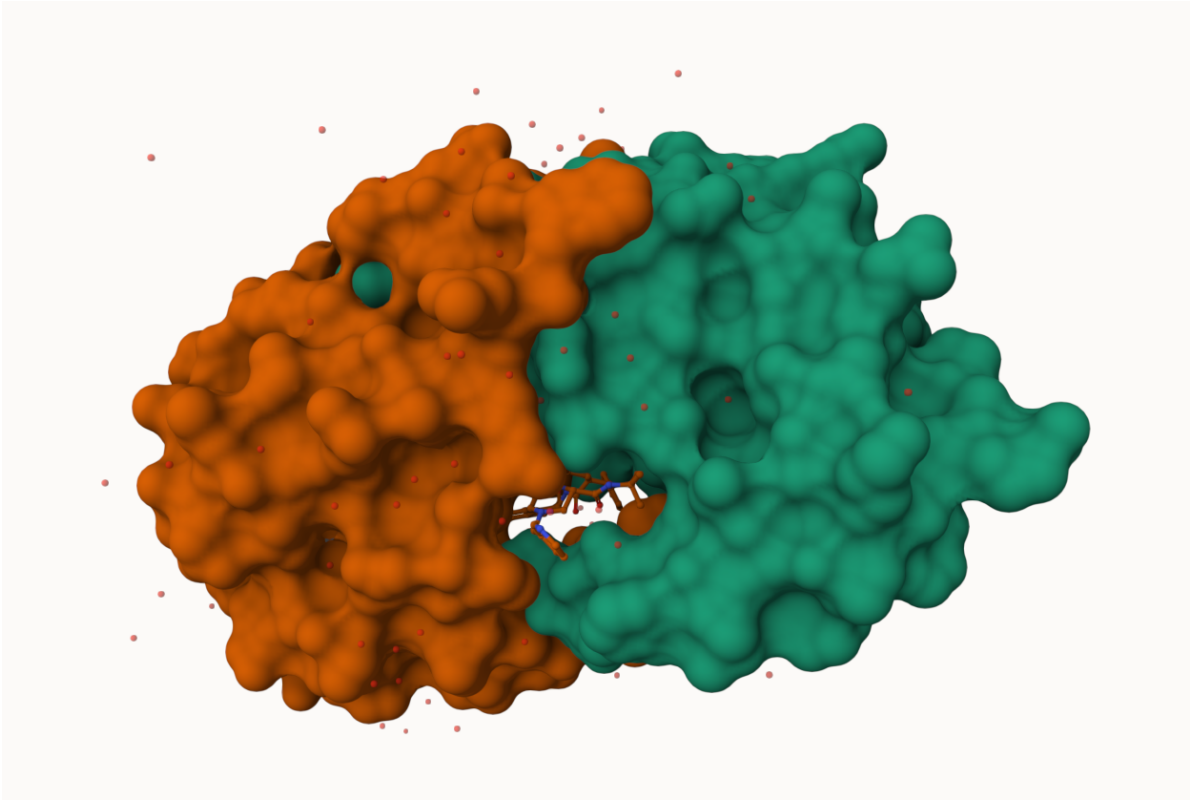
Using Mol*

Mol* is a new web-based molecualr-viewer that we will learn the basics of

some custom images:

Figure 1: a first image from Mol*





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

The molecular viewer simplified water molecules into one atom to prevent water molecule from blocking the protein structures. In this case, we can better observe the protein residues while still being able to see the interaction between residues and water molecules.

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 water 308. It is losely interacting with the ligand and is located at the binding site.

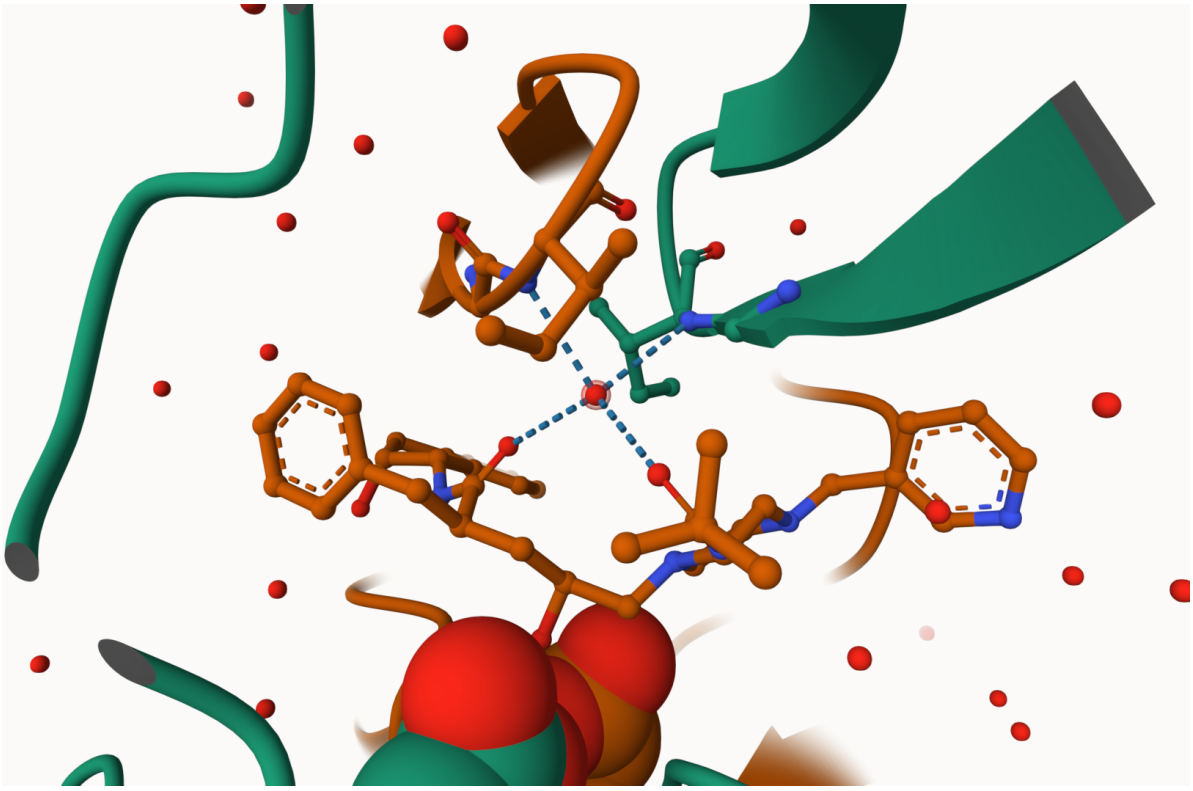


Figure 2: Water308 and its interaction with the ligand

Figure 3: Water308 in Spacefill Representation

The Bio3D package

The bio3d package allows us to do all sorts of structural bioinformatic works in R

Start with how it can read PDB files

```
library(bio3d)
```

```
Warning: package 'bio3d' was built under R version 4.3.3
```

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

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

```
pdb
```

```
 Call:  read.pdb(file = "1hsg")

   Total Models#: 1
     Total Atoms#: 1686,  XYZs#: 5058  Chains#: 2  (values: A B)

     Protein Atoms#: 1514  (residues/Calpha atoms#: 198)
     Nucleic acid Atoms#: 0  (residues/phosphate atoms#: 0)

     Non-protein/nucleic Atoms#: 172  (residues: 128)
     Non-protein/nucleic resid values: [ HOH (127), MK1 (1) ]

   Protein sequence:
      PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYD
      QILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFPQITLWQRPLVTIKIGGQLKE
      ALLDTGADDTVLEEMSLPGRWKPKMIGGIGGFIKVRQYDQILIEICGHKAIGTVLVGPTP
      VNIIGRNLLTQIGCTLNF

+ attr: atom, xyz, seqres, helix, sheet,
        calpha, remark, call
```

```
attributes(pdb)
```

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

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

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

```
  type eleno elety  alt resid chain resno insert      x      y     z o     b
1 ATOM     1     N <NA>   PRO     A     1   <NA> 29.361 39.686 5.862 1 38.10
2 ATOM     2    CA <NA>   PRO     A     1   <NA> 30.307 38.663 5.319 1 40.62
3 ATOM     3     C <NA>   PRO     A     1   <NA> 29.760 38.071 4.022 1 42.64
4 ATOM     4     O <NA>   PRO     A     1   <NA> 28.600 38.302 3.676 1 43.40
5 ATOM     5    CB <NA>   PRO     A     1   <NA> 30.508 37.541 6.342 1 37.87
6 ATOM     6    CG <NA>   PRO     A     1   <NA> 29.296 37.591 7.162 1 38.40
  segid elesy charge
1  <NA>     N   <NA>
2  <NA>     C   <NA>
```

```
3  <NA>     C   <NA>
4  <NA>     O   <NA>
5  <NA>     C   <NA>
6  <NA>     C   <NA>
```

```
head(pdbseq(pdb))
```

```
  1   2   3   4   5   6
"P" "Q" "I" "T" "L" "W"
```

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

there are 198 amino acid residues in this pdb object

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

```
[1] 198
```

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

HOH and MK1

Q9: How many protein chains are in this structure?

there are 2, chain A and chain B

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

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

Predicting functional motions of a single structure

Lets do a bioinformatics prediction of function motions - the movements that one of these molecules needs to make to do its stuff. Use Adenylate Kinase as example.

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

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

```
adk
```

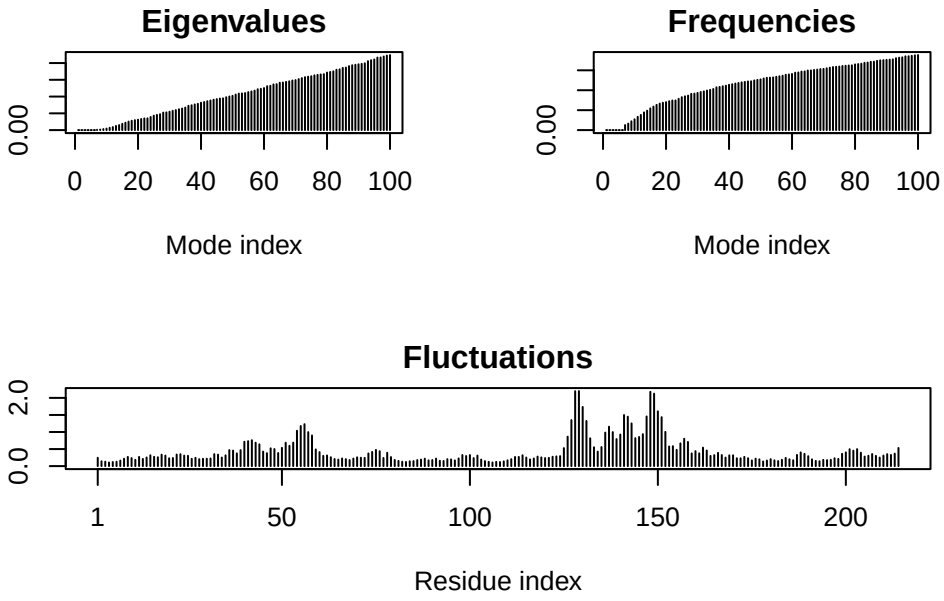
```
 Call:  read.pdb(file = "6s36")

   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
```

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

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

```
 Building Hessian...        Done in 0.023 seconds.
 Diagonalizing Hessian...   Done in 0.459 seconds.
```

```
plot(m)
```



write out multi-model PDB file that we can use to make an animation of the predicted motions. Can open the result in Mol*

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

Comparative structure analysis of Adenylate Kinase

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

The `msa` package is only found on BioConductor.

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

Devtools and Bio3d are not found on BioConductor.

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

True. R packages found on GitHub or BitBucket can be installed using devtools::install_github() and devtools::install_bitbucket() functions.

Search and retrieve ADK structures

Here we will find and analyze all ADK structures in the PDB database. We will start with a single database accession id "1ake_A"

we perform a blast search of the PDB database to identify related structures to our query Adenylate kinase (ADK) sequence. In this particular example we use function `get.seq()` to fetch the query sequence for chain A of the PDB ID 1AKE and use this as input to `blast.pdb()`

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

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

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

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

this sequence has 214 amino acids

```
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
```

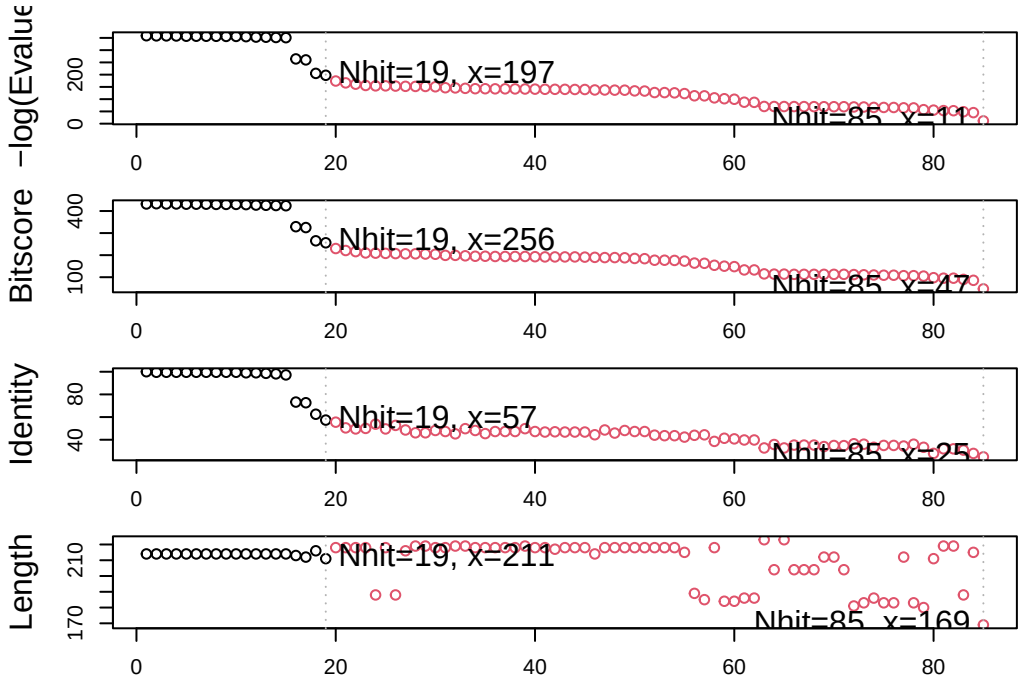
Blast the query sequence of ADK. It will set a seed position to the point of largest drop-off in normalized scores (i.e. the biggest jump in E-values). This is the default cutoff. We can also specify cutoff value.

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

```
 Searching ... please wait (updates every 5 seconds) RID = JUCZ878G016
 ..................................
 Reporting 85 hits
```

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

```
  * Possible cutoff values:    197 11
            Yielding Nhits:    19 85

  * Chosen cutoff value of:    197
            Yielding Nhits:    19
```




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

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

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

```
[1] 19
```

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

```
# 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/8BQF.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/8Q2B.pdb.gz exists. Skipping download

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/8RJ9.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%
  |
  |====                                                                  |   5%
  |
  |=======                                                               |  11%
  |
  |===========                                                           |  16%
  |
  |===============                                                       |  21%
  |
  |==================                                                    |  26%
  |
  |======================                                                |  32%
  |
  |==========================                                            |  37%
  |
  |==============================                                        |  42%
  |
  |==================================                                    |  47%
  |
  |======================================                                |  53%
  |
  |==========================================                            |  58%
  |
  |==============================================                        |  63%
  |
  |==================================================                    |  68%
```

```
  |
  |=================================================                     |  74%
  |
  |=====================================================                 |  79%
  |
  |=========================================================             |  84%
  |
  |=============================================================         |  89%
  |
  |=================================================================     |  95%
  |
  |======================================================================| 100%
```

Align and superpose structures

we will use the `pdbaln()` function to align and also optionally fit (i.e. superpose) 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/8BQF_A.pdb
pdbs/split_chain/4X8M_A.pdb
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/8Q2B_A.pdb
pdbs/split_chain/8RJ9_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
..   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/8BQF_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 3   name: pdbs/split_chain/4X8M_A.pdb
pdb/seq: 4   name: pdbs/split_chain/6S36_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 5   name: pdbs/split_chain/8Q2B_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 6   name: pdbs/split_chain/8RJ9_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 7   name: pdbs/split_chain/6RZE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 8   name: pdbs/split_chain/4X8H_A.pdb
pdb/seq: 9   name: pdbs/split_chain/3HPR_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 10   name: pdbs/split_chain/1E4V_A.pdb
pdb/seq: 11   name: pdbs/split_chain/5EJE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 12   name: pdbs/split_chain/1E4Y_A.pdb
pdb/seq: 13   name: pdbs/split_chain/3X2S_A.pdb
pdb/seq: 14   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 15   name: pdbs/split_chain/6HAM_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 16   name: pdbs/split_chain/4K46_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 17   name: pdbs/split_chain/4NP6_A.pdb
pdb/seq: 18   name: pdbs/split_chain/3GMT_A.pdb
pdb/seq: 19   name: pdbs/split_chain/4PZL_A.pdb
```

```
pdbs
```

```
                                 1        .         .         .         40
[Truncated_Name:1]1AKE_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:2]8BQF_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:3]4X8M_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:4]6S36_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:5]8Q2B_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:6]8RJ9_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:7]6RZE_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:8]4X8H_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:9]3HPR_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:10]1E4V_A.pdb    ----------MRIILLGAPVAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:11]5EJE_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:12]1E4Y_A.pdb    ----------MRIILLGALVAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:13]3X2S_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:14]6HAP_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:15]6HAM_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:16]4K46_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMAKFGIPQIS
[Truncated_Name:17]4NP6_A.pdb    --------NAMRIILLGAPGAGKGTQAQFIMEKFGIPQIS
[Truncated_Name:18]3GMT_A.pdb    ----------MRLILLGAPGAGKGTQANFIKEKFGIPQIS
[Truncated_Name:19]4PZL_A.pdb    TENLYFQSNAMRIILLGAPGAGKGTQAKIIEQKYNIAHIS
                                           **^*****  *******  *  *^ *  **
                                 1        .         .         .         40

                                41        .         .         .         80
[Truncated_Name:1]1AKE_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:2]8BQF_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:3]4X8M_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:4]6S36_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:5]8Q2B_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:6]8RJ9_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:7]6RZE_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:8]4X8H_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:9]3HPR_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:10]1E4V_A.pdb     TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:11]5EJE_A.pdb     TGDMLRAAVKSGSELGKQAKDIMDACKLVTDELVIALVKE
[Truncated_Name:12]1E4Y_A.pdb     TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:13]3X2S_A.pdb     TGDMLRAAVKSGSELGKQAKDIMDCGKLVTDELVIALVKE
[Truncated_Name:14]6HAP_A.pdb     TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVRE
[Truncated_Name:15]6HAM_A.pdb     TGDMLRAAIKSGSELGKQAKDIMDAGKLVTDEIIIALVKE
[Truncated_Name:16]4K46_A.pdb     TGDMLRAAIKAGTELGKQAKSVIDAGQLVSDDIILGLVKE
```

```
[Truncated_Name:17]4NP6_A.pdb   TGDMLRAAIKAGTELGKQAKAVIDAGQLVSDDIILGLIKE
[Truncated_Name:18]3GMT_A.pdb   TGDMLRAAVKAGTPLGVEAKTYMDEGKLVPDSLIIGLVKE
[Truncated_Name:19]4PZL_A.pdb   TGDMIRETIKSGSALGQELKKVLDAGELVSDEFIIKIVKD
                                ****^*  ^* *^ **   *  ^*   ** *  ^^ ^^^^
                               41        .         .         .         80

                               81        .         .         .         120
[Truncated_Name:1]1AKE_A.pdb    RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:2]8BQF_A.pdb    RIAQE----GFLLDGFPRTIPQADAMKEAGINVDYVIEFD
[Truncated_Name:3]4X8M_A.pdb    RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:4]6S36_A.pdb    RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:5]8Q2B_A.pdb    RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:6]8RJ9_A.pdb    RIAQEDCRNGFLLAGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:7]6RZE_A.pdb    RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:8]4X8H_A.pdb    RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:9]3HPR_A.pdb    RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:10]1E4V_A.pdb   RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:11]5EJE_A.pdb   RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:12]1E4Y_A.pdb   RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:13]3X2S_A.pdb   RIAQEDSRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:14]6HAP_A.pdb   RICQEDSRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:15]6HAM_A.pdb   RICQEDSRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:16]4K46_A.pdb   RIAQDDCAKGFLLDGFPRTIPQADGLKEVGVVVDYVIEFD
[Truncated_Name:17]4NP6_A.pdb   RIAQADCEKGFLLDGFPRTIPQADGLKEMGINVDYVIEFD
[Truncated_Name:18]3GMT_A.pdb   RLKEADCANGYLFDGFPRTIAQADAMKEAGVAIDYVLEID
[Truncated_Name:19]4PZL_A.pdb   RISKNDCNNGFLLDGVPRTIPQAQELDKLGVNIDYIVEVD
                                *^       *^*  * **** **  ^   *^ ^**^^* *
                               81        .         .         .         120

                              121        .         .         .         160
[Truncated_Name:1]1AKE_A.pdb    VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:2]8BQF_A.pdb    VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:3]4X8M_A.pdb    VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:4]6S36_A.pdb    VPDELIVDKIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:5]8Q2B_A.pdb    VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:6]8RJ9_A.pdb    VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:7]6RZE_A.pdb    VPDELIVDAIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:8]4X8H_A.pdb    VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:9]3HPR_A.pdb    VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDGTG
[Truncated_Name:10]1E4V_A.pdb   VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:11]5EJE_A.pdb   VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:12]1E4Y_A.pdb   VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:13]3X2S_A.pdb   VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
```

```
[Truncated_Name:14]6HAP_A.pdb   VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:15]6HAM_A.pdb   VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:16]4K46_A.pdb   VADSVIVERMAGRRAHLASGRTYHNVYNPPKVEGKDDVTG
[Truncated_Name:17]4NP6_A.pdb   VADDVIVERMAGRRAHLPSGRTYHVVYNPPKVEGKDDVTG
[Truncated_Name:18]3GMT_A.pdb   VPFSEIIERMSGRRTHPASGRTYHVKFNPPKVEGKDDVTG
[Truncated_Name:19]4PZL_A.pdb   VADNLLIERITGRRIHPASGRTYHTKFNPPKVADKDDVTG
                                *    ^^^ ^ *** *  *** **  ^*****  *** **
                               121        .         .         .        160

                               161        .         .         .        200
[Truncated_Name:1]1AKE_A.pdb    EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:2]8BQF_A.pdb    EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:3]4X8M_A.pdb    EELTTRKDDQEETVRKRLVEWHQMTAPLIGYYSKEAEAGN
[Truncated_Name:4]6S36_A.pdb    EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:5]8Q2B_A.pdb    EELTTRKADQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:6]8RJ9_A.pdb    EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:7]6RZE_A.pdb    EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:8]4X8H_A.pdb    EELTTRKDDQEETVRKRLVEYHQMTAALIGYYSKEAEAGN
[Truncated_Name:9]3HPR_A.pdb    EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:10]1E4V_A.pdb   EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:11]5EJE_A.pdb   EELTTRKDDQEECVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:12]1E4Y_A.pdb   EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:13]3X2S_A.pdb   EELTTRKDDQEETVRKRLCEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:14]6HAP_A.pdb   EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:15]6HAM_A.pdb   EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:16]4K46_A.pdb   EDLVIREDDKEETVLARLGVYHNQTAPLIAYYGKEAEAGN
[Truncated_Name:17]4NP6_A.pdb   EDLVIREDDKEETVRARLNVYHTQTAPLIEYYGKEAAAGK
[Truncated_Name:18]3GMT_A.pdb   EPLVQRDDDKEETVKKRLDVYEAQTKPLITYYGDWARRGA
[Truncated_Name:19]4PZL_A.pdb   EPLITRTDDNEDTVKQRLSVYHAQTAKLIDFYRNFSSTNT
                                * *  *  * *^ *  **  ^   *  ** ^*
                               161        .         .         .        200

                               201        .         .      227
[Truncated_Name:1]1AKE_A.pdb    T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:2]8BQF_A.pdb    T--KYAKVDGTKPVAEVRADLEKIL--
[Truncated_Name:3]4X8M_A.pdb    T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:4]6S36_A.pdb    T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:5]8Q2B_A.pdb    T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:6]8RJ9_A.pdb    T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:7]6RZE_A.pdb    T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:8]4X8H_A.pdb    T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:9]3HPR_A.pdb    T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:10]1E4V_A.pdb   T--KYAKVDGTKPVAEVRADLEKILG-
```

```
[Truncated_Name:11]5EJE_A.pdb   T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:12]1E4Y_A.pdb   T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:13]3X2S_A.pdb   T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:14]6HAP_A.pdb   T--KYAKVDGTKPVCEVRADLEKILG-
[Truncated_Name:15]6HAM_A.pdb   T--KYAKVDGTKPVCEVRADLEKILG-
[Truncated_Name:16]4K46_A.pdb   T--QYLKFDGTKAVAEVSAELEKALA-
[Truncated_Name:17]4NP6_A.pdb   T--QYLKFDGTKQVSEVSADIAKALA-
[Truncated_Name:18]3GMT_A.pdb   E-------NGLKAPA-----YRKISG-
[Truncated_Name:19]4PZL_A.pdb   KIPKYIKINGDQAVEKVSQDIFDQLNK
                                         *
                               201        .         .      227

Call:
  pdbaln(files = files, fit = TRUE, exefile = "msa")

Class:
  pdbs, fasta

Alignment dimensions:
  19 sequence rows; 227 position columns (199 non-gap, 28 gap)

+ attr: xyz, resno, b, chain, id, ali, resid, sse, call
```

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

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

Annotate collected PDB structures

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

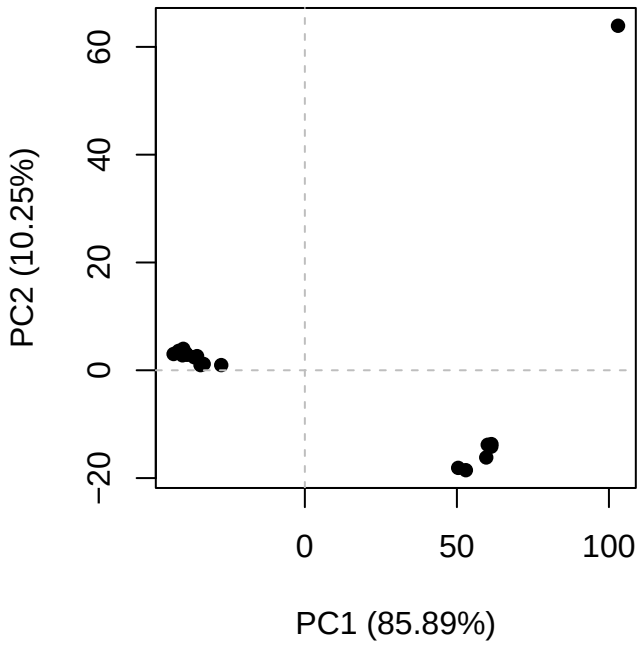
```
#anno <- pdb.annotate(ids)
#unique(anno$source)
```

Principle Component Analysis

Function pca() provides principal component analysis (PCA) of the structure data. In terms of protein structures PCA is used to capture major structural variations within similar protein

structures (top hits of ADK)

```
# Perform PCA
pc.xray <- pca(pdbs)
plot(pc.xray, pc.axes = c(1, 2)) # just the PC1 and PC2
```



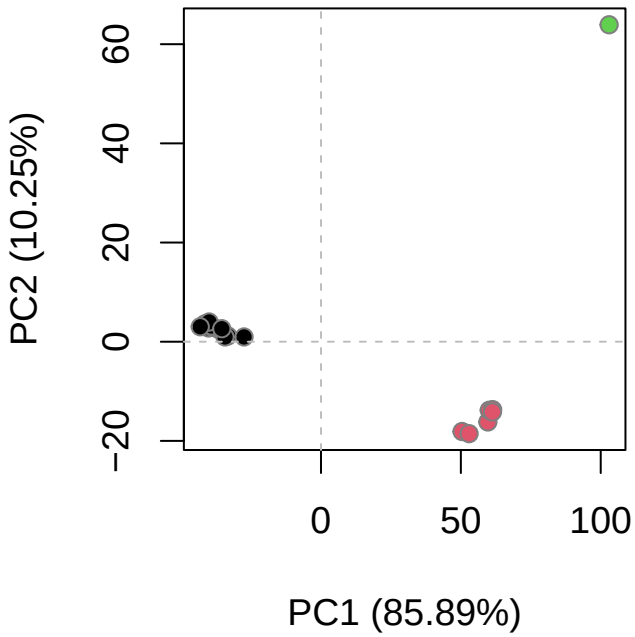

```
# Visualize first principal component. Showing the structures of homolog proteins according t
pc1 <- mktrj(pc.xray, pc=1, file="pc_1.pdb")
```

Function rmsd() will calculate all pairwise RMSD values of the structural ensemble. This facilitates clustering analysis based on the pairwise structural variations:

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

```
Warning in rmsd(pdbs): No indices provided, using the 199 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 (10.25%)
60
40
20
0
−20
0
50
100
PC1 (85.89%)