Class 10 Structural BioInformatics 1

Gonzalez A16745338

The main database for structural data is called the PDB (Protein Data Bank) Let's see what it contains:

Read this into R:

Data from: https://www.rcsb.org/stats

Answer the following questions:

```r
pdbdb <- read.csv("pdb_stats.csv", row.names = 1)
pdbdb
```

	X.ray	EM	NMR	Multiple.methods	Neutron	Other
Protein (only)	167,192	15,572	12,529	208	77	32
Protein/Oligosaccharide	9,639	2,635	34	8	2	0
Protein/NA	8,730	4,697	286	7	0	0
Nucleic acid (only)	2,869	137	1,507	14	3	1
Other	170	10	33	0	0	0
Oligosaccharide (only)	11	0	6	1	0	4

	Total
Protein (only)	195,610
Protein/Oligosaccharide	12,318
Protein/NA	13,720
Nucleic acid (only)	4,531
Other	213
Oligosaccharide (only)	22

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

I need to remove the comma and conert to numeric to do math:

```
as.numeric( sub(",", "", pdbdb$Total) )
```

```
[1] 195610   12318   13720    4531     213      22
```

```
#as.numeric(pdbdb$Total)
```

I could turn this into a function to fix the whole table or any future table I read like this:

```
x <- pdbdb$Total
as.numeric( sub(",", "", x) )
```

```
[1] 195610   12318   13720    4531     213      22
```

```
comma2numeric <- function(x) {
  as.numeric( sub(",", "", x) )

}
```

Test it

```
comma2numeric(pdbdb$X.ray)
```

```
[1] 167192    9639    8730    2869     170      11
```

```
apply(pdbdb, 2, comma2numeric)
```

```
       X.ray    EM   NMR Multiple.methods Neutron Other  Total
[1,]  167192 15572 12529              208      77    32 195610
[2,]    9639  2635    34                8       2     0  12318
[3,]    8730  4697   286                7       0     0  13720
[4,]    2869   137  1507               14       3     1   4531
[5,]     170    10    33                0       0     0    213
[6,]      11     0     6                1       0     4     22
```

Or try a different read/import function

```
library(readr)
pdbdb <- read_csv("pdb_stats.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.
```

```
sum(pdbdb$Total)
```

```
[1] 226414
```

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

```
sum(pdbdb$'X-ray')/sum(pdbdb$Total) * 100
```

```
[1] 83.30359
```

```
sum(pdbdb$EM)/sum(pdbdb$Total)*100
```

```
[1] 10.18091
```

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

```
pdbdb$Total
```

```
[1] 195610  12318  13720   4531    213     22
```

Q3: 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?

5 structures

Mol *

Mol* (pronounced "molstar") is a new web-based molecular viewer that we will need to learn the basics of here.

https://molstar.org/viewer/

We will use PDB code: 1HSG

Figure 1: A first image from molstar

Some more custom images:

Figure 2: The all important catalytic ASP25 amino acid
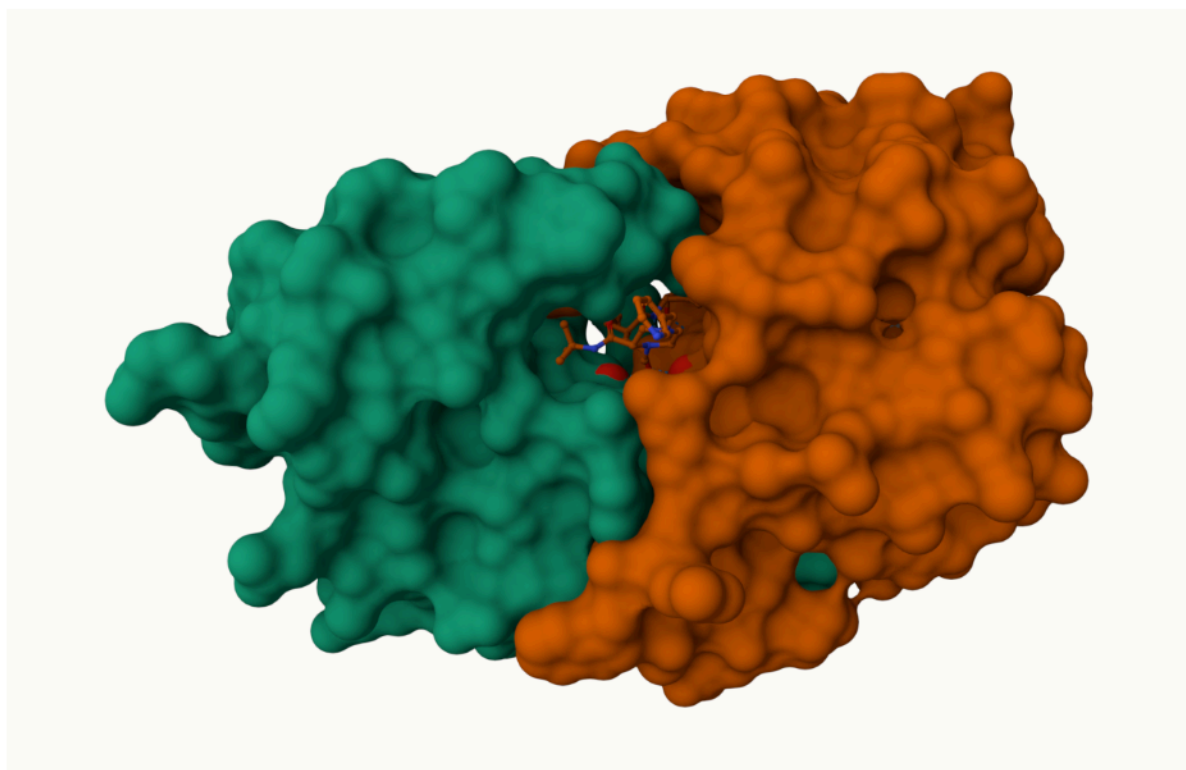
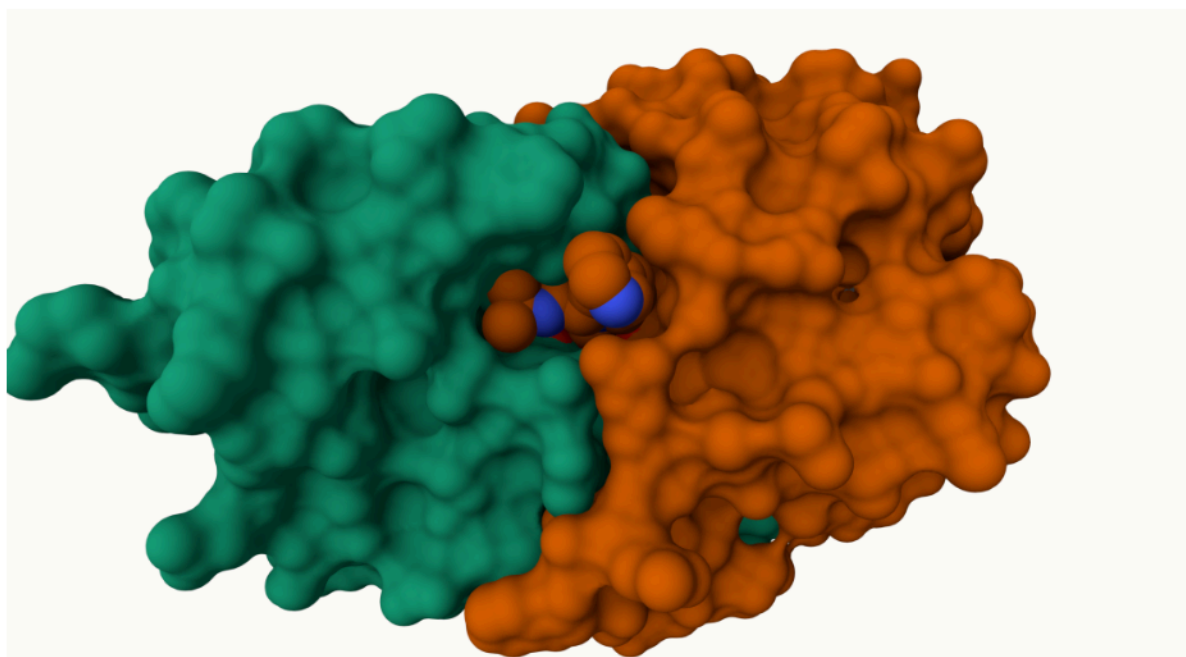Figure 3: Surface display showing Merk compound in the peptide binding pocket

Figure 4: Ligand in pocket

Figure 5: Water molecule

The Bio3D package

The bio3d package allows us to do all sorts of structural bioinformatics work in R.
Let's start with how it can read these PDB files:

```
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
```

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>
```

```
pdbseq(pdb)[25]
```

```
  25
"D"
```

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

```
sum(pdb$calpha)
```

```
[1] 198
```

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

HOH and MK1

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

2

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

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

Predicting functional motions of a single structure

Let's do a bioinformatics prediction of functional motions - i.e. the movements that one of these molecules needs to make to do its stuff.

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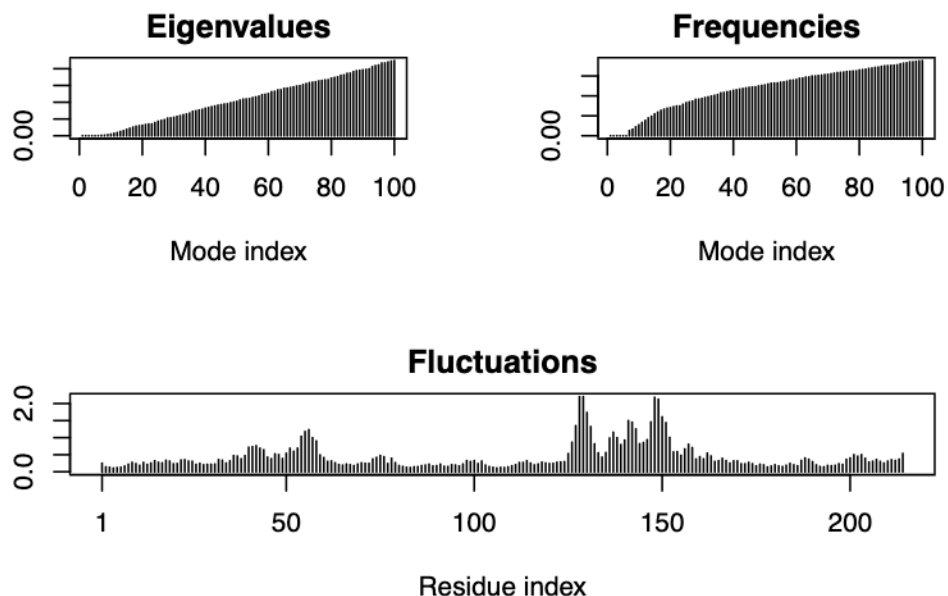
```
# Perform flexiblity prediction
m <- nma(adk)
```

```
 Building Hessian...        Done in 0.064 seconds.
 Diagonalizing Hessian...   Done in 1.007 seconds.
```

```
plot(m)
```

Eigenvalues

0.00

0 20 40 60 80 100

Mode index

Frequencies

0.00

0 20 40 60 80 100

Mode index

Fluctuations

2.0

0.0

1   50   100   150   200

Residue index

Write out multi-model PDB file that we can use to make an animation of the predicted motions.

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

I can open this in Mol* to play the trajectory...

Comparative analysis of protein structures

```
library(bio3d)
```

Here we will find and analyze all ADK structures in the PDB database.

We will start with a single database acession id:

```
id <- "1AKE_A"
aa <- get.seq(id)
```

Warning in get.seq(id): Removing existing file: seqs.fasta

Fetching... Please wait. Done.

I ran….

install.packages("BiocManager") BiocManager::install("msa")

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

The 'msa' package is from BioConductor.

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

the 'bio3d-view' package is not found on neither

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

TRUE

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

214 amino acids in the sequence

```
aa$id
```

```
[1] "pdb|1AKE|A"
```

```
ncol(aa$ali)
```

```
[1] 214
```

```
#b <- blast.pbd(aa)
```

```
#attributes(b)
#plot(b$hit.tbl)
```

```
#hits <- plot(b)
```

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

Pre-calculated results:

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

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

Warning in get.pdb(hits$pdb.id, path = "pdbs", split = TRUE, gzip = TRUE):
pdbs/1AKE.pdb.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%
```

Next 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/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
```

pdbs

```
                                  1         .         .         .        40
[Truncated_Name:1]1AKE_A.pdb      ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:2]6S36_A.pdb      ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:3]6RZE_A.pdb      ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:4]3HPR_A.pdb      ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:5]1E4V_A.pdb      ----------MRIILLGAPVAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:6]5EJE_A.pdb      ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:7]1E4Y_A.pdb      ----------MRIILLGALVAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:8]3X2S_A.pdb      ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:9]6HAP_A.pdb      ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:10]6HAM_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:11]4K46_A.pdb     ----------MRIILLGAPGAGKGTQAQFIMAKFGIPQIS
[Truncated_Name:12]3GMT_A.pdb     ----------MRLILLGAPGAGKGTQANFIKEKFGIPQIS
[Truncated_Name:13]4PZL_A.pdb     TENLYFQSNAMRIILLGAPGAGKGTQAKIIEQKYNIAHIS
                                            **^*****  *******  *   *^ *   **
                                  1         .         .         .        40


                                  41        .         .         .        80
[Truncated_Name:1]1AKE_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:2]6S36_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:3]6RZE_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:4]3HPR_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:5]1E4V_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:6]5EJE_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDACKLVTDELVIALVKE
[Truncated_Name:7]1E4Y_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVKE
[Truncated_Name:8]3X2S_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDCGKLVTDELVIALVKE
[Truncated_Name:9]6HAP_A.pdb      TGDMLRAAVKSGSELGKQAKDIMDAGKLVTDELVIALVRE
[Truncated_Name:10]6HAM_A.pdb     TGDMLRAAIKSGSELGKQAKDIMDAGKLVTDEIIIALVKE
[Truncated_Name:11]4K46_A.pdb     TGDMLRAAIKAGTELGKQAKSVIDAGQLVSDDIILGLVKE
[Truncated_Name:12]3GMT_A.pdb     TGDMLRAAVKAGTPLGVEAKTYMDEGKLVPDSLIIGLVKE
[Truncated_Name:13]4PZL_A.pdb     TGDMIRETIKSGSALGQELKKVLDAGELVSDEFIIKIVKD
                                  ****^*  ^* *^ **   *  ^*   ** *  ^^ ^*^^
                                  41        .         .         .        80


                                  81        .         .         .       120
[Truncated_Name:1]1AKE_A.pdb      RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:2]6S36_A.pdb      RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
```

18

```
[Truncated_Name:3]6RZE_A.pdb      RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:4]3HPR_A.pdb      RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:5]1E4V_A.pdb      RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:6]5EJE_A.pdb      RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:7]1E4Y_A.pdb      RIAQEDCRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:8]3X2S_A.pdb      RIAQEDSRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:9]6HAP_A.pdb      RICQEDSRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:10]6HAM_A.pdb     RICQEDSRNGFLLDGFPRTIPQADAMKEAGINVDYVLEFD
[Truncated_Name:11]4K46_A.pdb     RIAQDDCAKGFLLDGFPRTIPQADGLKEVGVVVDYVIEFD
[Truncated_Name:12]3GMT_A.pdb     RLKEADCANGYLFDGFPRTIAQADAMKEAGVAIDYVLEID
[Truncated_Name:13]4PZL_A.pdb     RISKNDCNNGFLLDGVPRTIPQAQELDKLGVNIDYIVEVD
                                  *^      *     *^* ** **** **   ^    *^ ^**^^* *
                             81          .         .         .         120


                             121         .         .         .         160
[Truncated_Name:1]1AKE_A.pdb      VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:2]6S36_A.pdb      VPDELIVDKIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:3]6RZE_A.pdb      VPDELIVDAIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:4]3HPR_A.pdb      VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDGTG
[Truncated_Name:5]1E4V_A.pdb      VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:6]5EJE_A.pdb      VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:7]1E4Y_A.pdb      VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:8]3X2S_A.pdb      VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:9]6HAP_A.pdb      VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:10]6HAM_A.pdb     VPDELIVDRIVGRRVHAPSGRVYHVKFNPPKVEGKDDVTG
[Truncated_Name:11]4K46_A.pdb     VADSVIVERMAGRRAHLASGRTYHNVYNPPKVEGKDDVTG
[Truncated_Name:12]3GMT_A.pdb     VPFSEIIERMSGRRTHPASGRTYHVKFNPPKVEGKDDVTG
[Truncated_Name:13]4PZL_A.pdb     VADNLLIERITGRRIHPASGRTYHTKFNPPKVADKDDVTG
                                  *    ^^^ ^ *** *  *** **  ^*****  *** **
                             121         .         .         .         160


                             161         .         .         .         200
[Truncated_Name:1]1AKE_A.pdb      EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:2]6S36_A.pdb      EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:3]6RZE_A.pdb      EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:4]3HPR_A.pdb      EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:5]1E4V_A.pdb      EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:6]5EJE_A.pdb      EELTTRKDDQEECVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:7]1E4Y_A.pdb      EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:8]3X2S_A.pdb      EELTTRKDDQEETVRKRLCEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:9]6HAP_A.pdb      EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:10]6HAM_A.pdb     EELTTRKDDQEETVRKRLVEYHQMTAPLIGYYSKEAEAGN
[Truncated_Name:11]4K46_A.pdb     EDLVIREDDKEETVLARLGVYHNQTAPLIAYYGKEAEAGN
```

19

```
[Truncated_Name:12]3GMT_A.pdb     EPLVQRDDDKEETVKKRLDVYEAQTKPLITYYGDWARRGA
[Truncated_Name:13]4PZL_A.pdb     EPLITRTDDNEDTVKQRLSVYHAQTAKLIDFYRNFSSTNT
                                  * *   * ** *^ *  **   *   *  ** ^*
                                161         .        .          .         200


                                201         .        .    227
[Truncated_Name:1]1AKE_A.pdb      T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:2]6S36_A.pdb      T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:3]6RZE_A.pdb      T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:4]3HPR_A.pdb      T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:5]1E4V_A.pdb      T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:6]5EJE_A.pdb      T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:7]1E4Y_A.pdb      T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:8]3X2S_A.pdb      T--KYAKVDGTKPVAEVRADLEKILG-
[Truncated_Name:9]6HAP_A.pdb      T--KYAKVDGTKPVCEVRADLEKILG-
[Truncated_Name:10]6HAM_A.pdb     T--KYAKVDGTKPVCEVRADLEKILG-
[Truncated_Name:11]4K46_A.pdb     T--QYLKFDGTKAVAEVSAELEKALA-
[Truncated_Name:12]3GMT_A.pdb     E-------NGLKAPA-----YRKISG-
[Truncated_Name:13]4PZL_A.pdb     KIPKYIKINGDQAVEKVSQDIFDQLNK
                                             *
                                201         .        .    227


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

Class:
  pdbs, fasta

Alignment dimensions:
  13 sequence rows; 227 position columns (204 non-gap, 23 gap)

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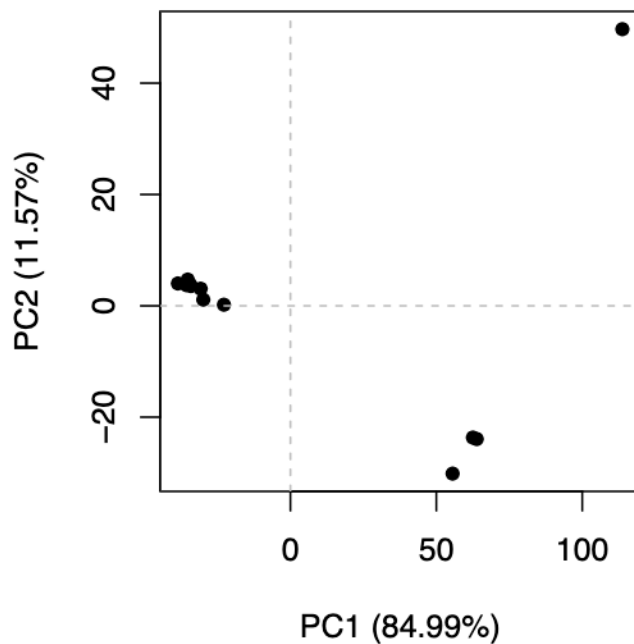
Principal Component Analysis

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

To visualize the major structural variations in the ensemble the function mktrj() can be used to generate a trajectory PDB file by interpolating along a give PC (eigenvector):

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

```
uniprot <- 248838887
pdb <- 195610

pdb/uniprot * 100
```

```
[1] 0.0786091
```

```
# 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%
```