Class_10_020924

Johann Tailor

Whats in the PDB?

Downloaded a CSV file from PDB Link: http://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
```

```
as.numeric(pdbstats$X.ray)
```

```
Warning: NAs introduced by coercion
```

```
[1]  NA  NA  NA  NA 164  11
```

```r
# we want to remove all commas in the data so that its not characters anymore:

#x <- "2,2222,22"

#gsub(",", "xoxoxoxooxox",x)
#gsub(",", "",x)
#as.numeric(gsub(",", "",x))


#now we want to sum everything for the purpose of this
comma_sum <- function(x) {
  sum(as.numeric(gsub(",", "",x)))
}


#go to code > Extract function > make your function
```

```r
comma_sum(pdbstats$X.ray)
```

```
[1] 182348
```

```r
comma_sum(pdbstats$Total)
```

```
[1] 215684
```

```r
comma_sum(pdbstats["Protein (only)","Total"])
```

```
[1] 186898
```

```r
#applying to the whole table

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
```

```
apply(pdbstats, 2, comma_sum) / comma_sum(pdbstats$Total)
```

```
         X.ray               EM              NMR Multiple.methods
  0.8454405519     0.0872433746     0.0657118748     0.0010663749
       Neutron            Other            Total
  0.0003662766     0.0001715473     1.0000000000
```

```
# I want to round it up now:
round(apply(pdbstats, 2, comma_sum) / comma_sum(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.

ANSWER: X.ray: 84.54% EM: 8.72%

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

```
#pdbstats$Total["Protein (only)", ]

#pdbstats[pdbstats$Total == "Protein (only)", "Total"]

# Assuming pdbstats is the data frame containing information about structures in the PDB
# Access the value for "Protein (only)" from the "Total" column
#protein_only_count <- pdbstats[pdbstats$Total == "Other", "Total"]
```

```
#protein_only_count

# Calculate the proportion of protein structures
#proportion_protein <- protein_only_count / sum((pdbstats$Total)

# Print the proportion
#print(proportion_protein)

Protein_proportion <- (comma_sum(pdbstats["Protein (only)","Total"]) / comma_sum(pdbstats$

Protein_proportion
```

```
[1] 86.65362
```

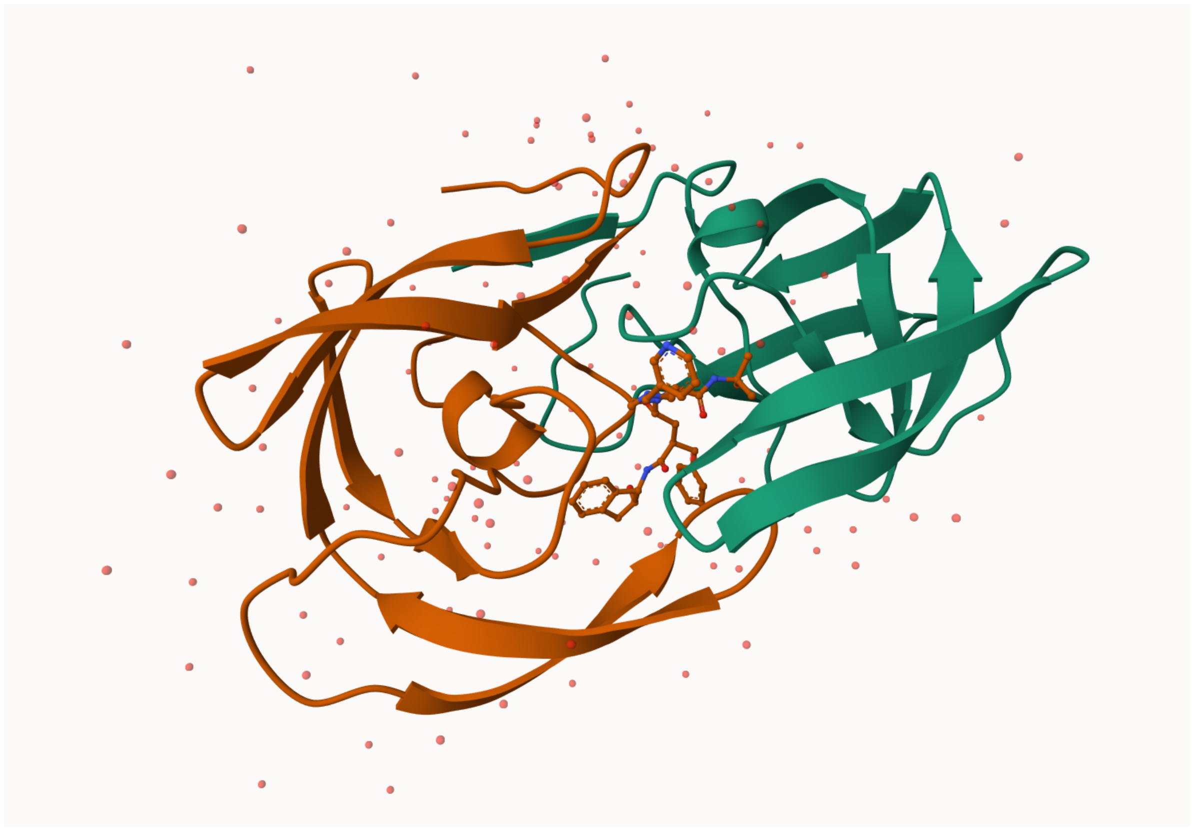
ANSWER: 86.65%

> 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?

```
library(rentrez)

#search and retrieve HIV-1 proteases
search_result <- entrez_search(db = "structure", term = "HIV-1 protease", retmax = 10000)

#Count
HIV1_Count <- search_result$count

# Print the count of HIV-1 protease structures in the PDB
HIV1_Count
```

```
[1] 1065
```

ANSWER=1065



Working with structures in R

We will use the package bio3d for structural bioinformatics.

```r
library(bio3d)

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

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

```r
hiv
```

```

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

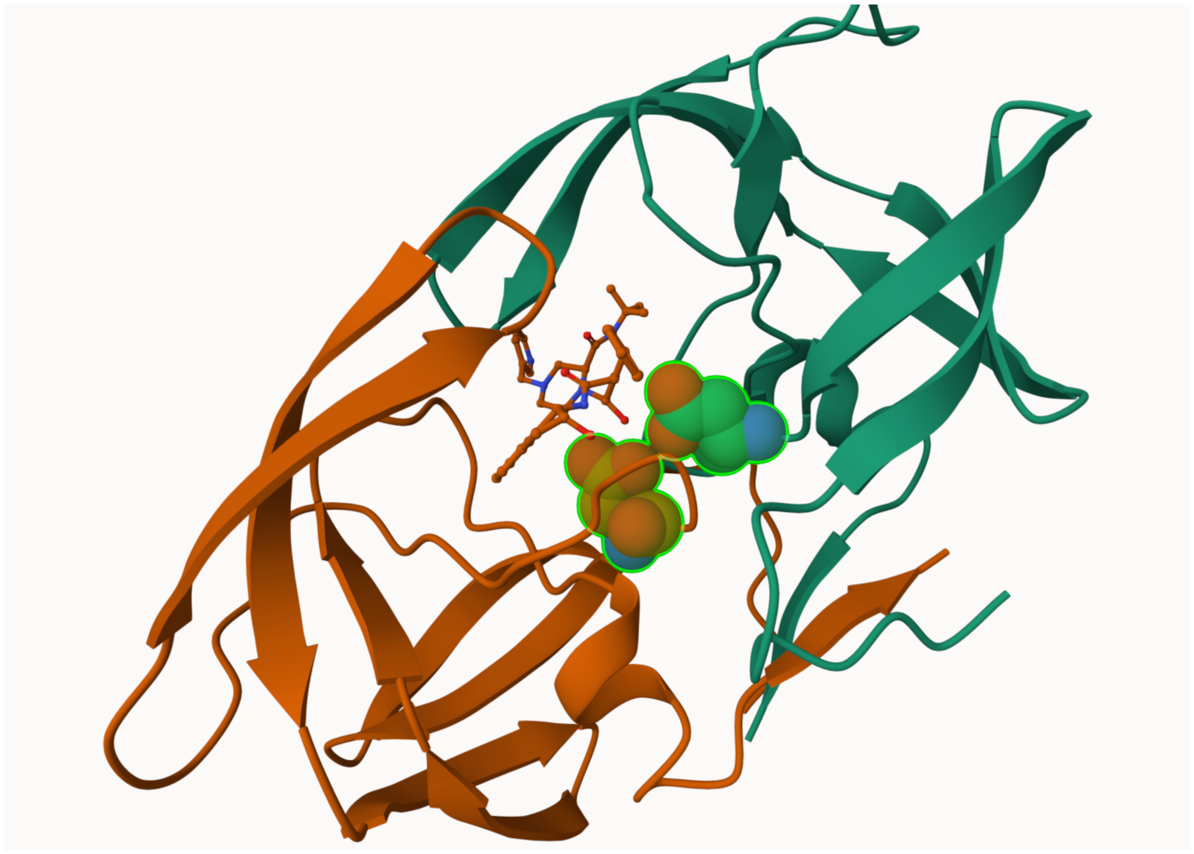


Figure 1: A nice display showing the homodimeric inhibitor with the Asp25 higlighted

```
   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
```

```
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"
```

Predicting functional motions of a single structure

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

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

```
adk
```

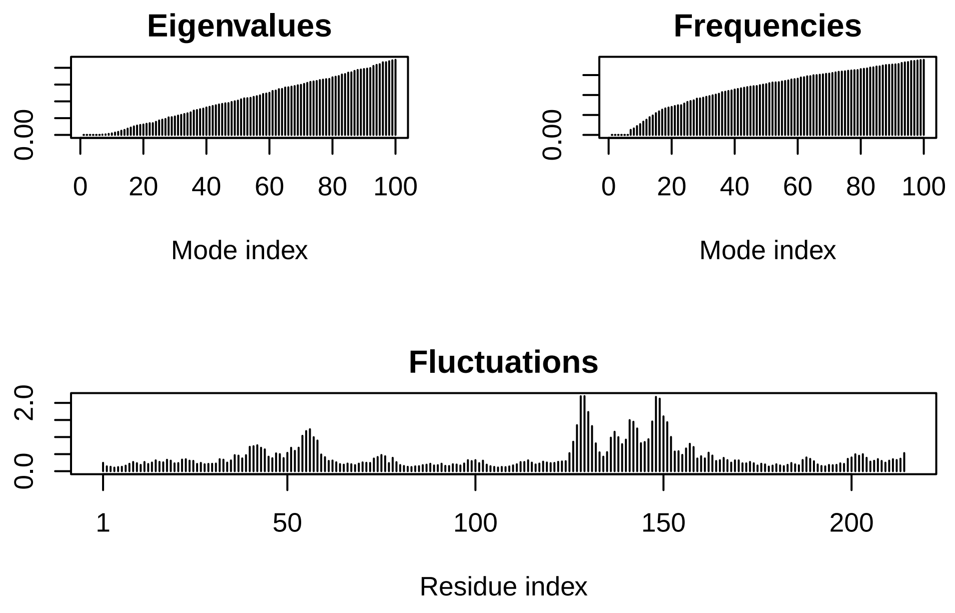
```

 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) a tool to predict motions and large-scale structure changes

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

```
 Building Hessian...        Done in 0.02 seconds.
 Diagonalizing Hessian...   Done in 0.462 seconds.
```

```
plot(m)
```



Let's make a movie (a.k.a "trajectory)

```
#To view a "movie" of these predicted motions we can generate a molecular "trajectory" wit

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

Quick comparative analysis of structures

Workflow:

1-PDB seq is in adk 2-Get seq 3-BLAST against PDB 4-Download all the hits 5-Superpose all structures from the blast hit

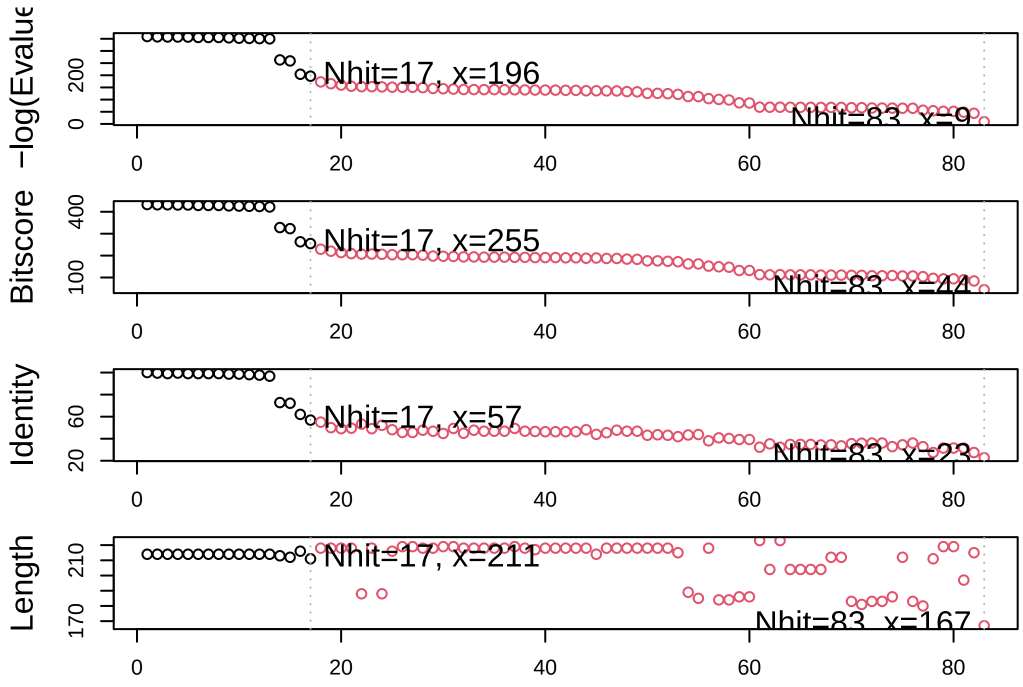
```
s <- pdbseq(adk)
blast <- blast.pdb(s)
```

```
 Searching ... please wait (updates every 5 seconds) RID = WMHWK27X01N
```

```
.
Reporting 83 hits
```

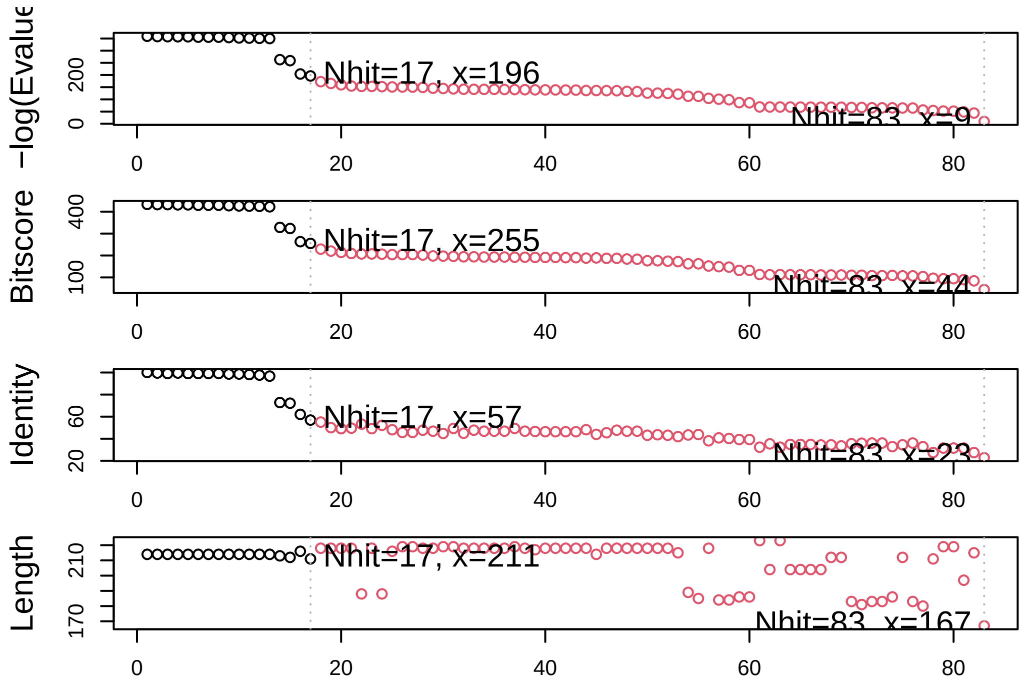
```r
plot(blast)
```

```
 * Possible cutoff values:    196 9
            Yielding Nhits:    17 83

 * Chosen cutoff value of:    196
            Yielding Nhits:    17
```



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

```
 * Possible cutoff values:    196 9
            Yielding Nhits:    17 83

 * Chosen cutoff value of:    196
            Yielding Nhits:    17
```



```r
#this will give us all the accession numbers of the 17 hits that matched the protein seque

hits$pdb.id
```

```
 [1] "6S36_A" "1AKE_A" "8BQF_A" "6RZE_A" "4X8M_A" "4X8H_A" "1E4V_A" "3HPR_A"
 [9] "5EJE_A" "1E4Y_A" "3X2S_A" "6HAP_A" "6HAM_A" "4K46_A" "4NP6_A" "3GMT_A"
[17] "4PZL_A"
```

But, lets automate this process using a string of code:

```r
# Download releated PDB files in a "pdbs" folder
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/6S36.pdb.gz exists. Skipping download

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/6RZE.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/4X8H.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/3HPR.pdb.gz exists. Skipping download

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

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

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

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

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

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

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

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

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

```
  |
  |                                                                      |   0%
  |
  |====                                                                  |   6%
  |
  |========                                                              |  12%
  |
  |============                                                          |  18%
  |
  |================                                                      |  24%
  |
  |=====================                                                 |  29%
  |
  |=========================                                             |  35%
  |
  |=============================                                         |  41%
  |
  |=================================                                     |  47%
  |
  |=====================================                                 |  53%
  |
  |=========================================                             |  59%
  |
  |=============================================                         |  65%
  |
  |=================================================                     |  71%
  |
  |======================================================                |  76%
  |
  |==========================================================            |  82%
  |
  |==============================================================        |  88%
  |
  |==================================================================    |  94%
  |
  |======================================================================| 100%
```

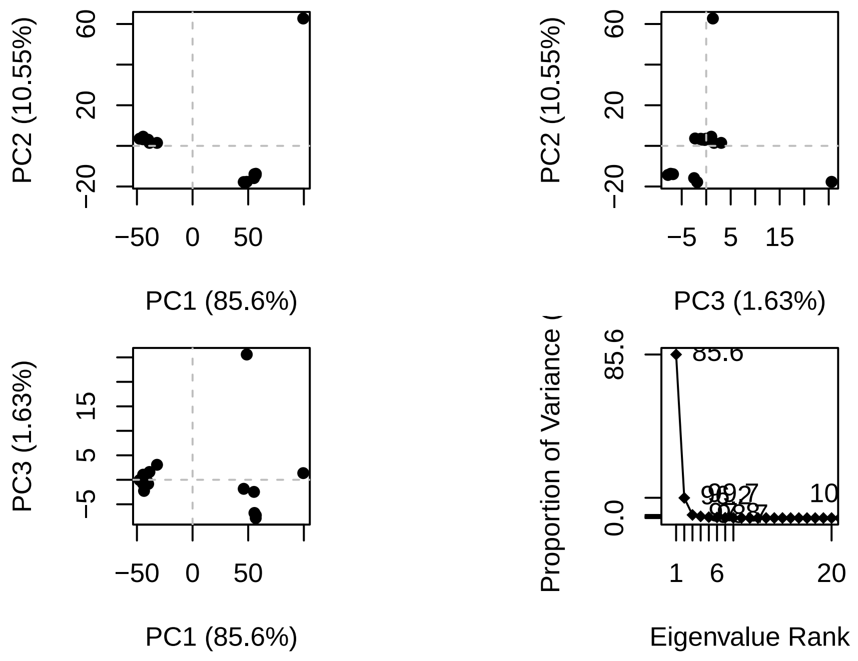
```r
# Align releated PDBs using MSA and putting structures on top of each other.
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

```
Reading PDB files:
pdbs/split_chain/6S36_A.pdb
pdbs/split_chain/1AKE_A.pdb
pdbs/split_chain/8BQF_A.pdb
pdbs/split_chain/6RZE_A.pdb
pdbs/split_chain/4X8M_A.pdb
pdbs/split_chain/4X8H_A.pdb
pdbs/split_chain/1E4V_A.pdb
pdbs/split_chain/3HPR_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
....

Extracting sequences

pdb/seq: 1   name: pdbs/split_chain/6S36_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 2   name: pdbs/split_chain/1AKE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 3   name: pdbs/split_chain/8BQF_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 4   name: pdbs/split_chain/6RZE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 5   name: pdbs/split_chain/4X8M_A.pdb
```

```
pdb/seq: 6   name: pdbs/split_chain/4X8H_A.pdb
pdb/seq: 7   name: pdbs/split_chain/1E4V_A.pdb
pdb/seq: 8   name: pdbs/split_chain/3HPR_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 9   name: pdbs/split_chain/5EJE_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 10   name: pdbs/split_chain/1E4Y_A.pdb
pdb/seq: 11   name: pdbs/split_chain/3X2S_A.pdb
pdb/seq: 12   name: pdbs/split_chain/6HAP_A.pdb
pdb/seq: 13   name: pdbs/split_chain/6HAM_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 14   name: pdbs/split_chain/4K46_A.pdb
   PDB has ALT records, taking A only, rm.alt=TRUE
pdb/seq: 15   name: pdbs/split_chain/4NP6_A.pdb
pdb/seq: 16   name: pdbs/split_chain/3GMT_A.pdb
pdb/seq: 17   name: pdbs/split_chain/4PZL_A.pdb
```

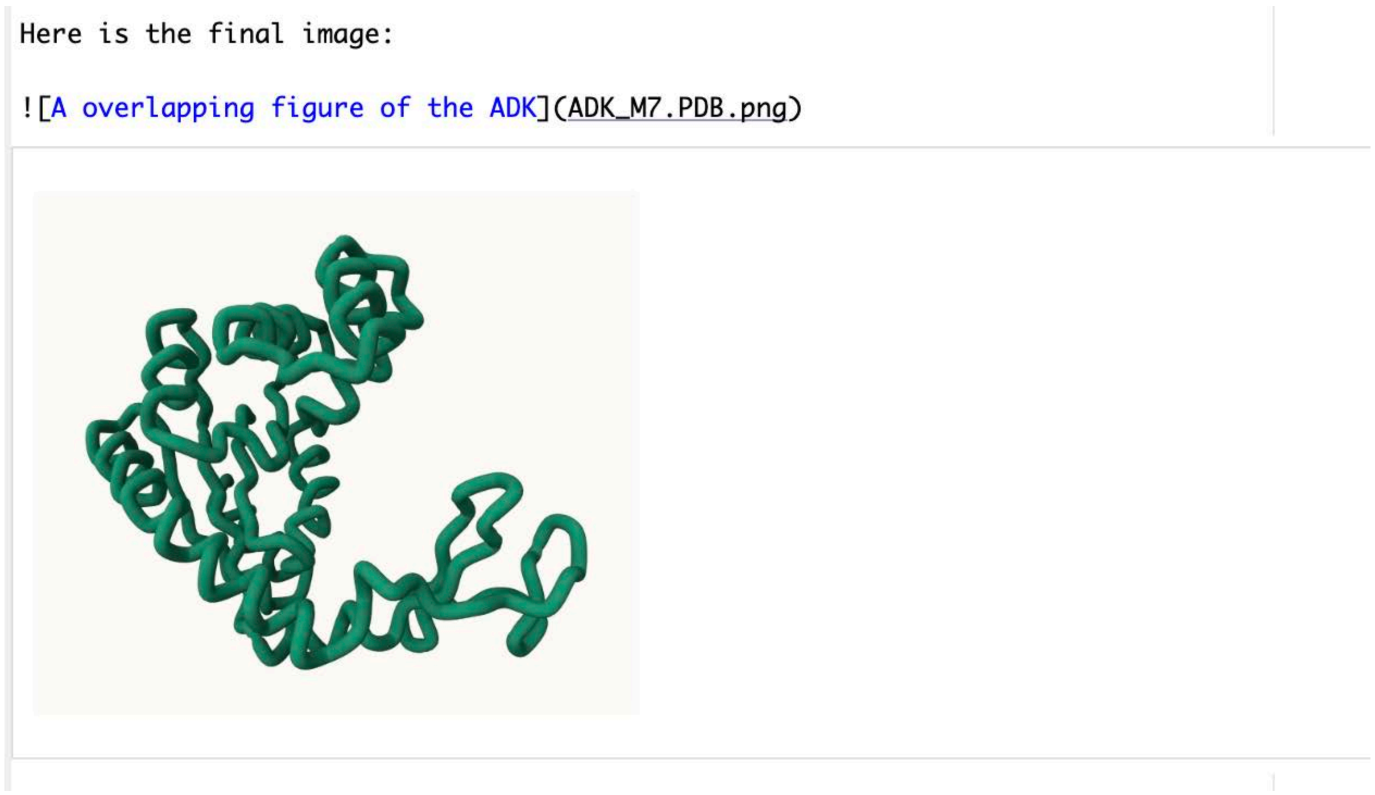
PCA of structures

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



Let's make a trajectory:

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



```
Here is the final image:

![A overlapping figure of the ADK](ADK_M7.PDB.png)
```

Note: For the last figure, I added it manually (inserted from screenshot) because NCBI blast was taking very long when rendering the file.