class09_lab

Table of contents

#PDB Presets and import data

```r
library(dplyr)
```

```
Warning: package 'dplyr' was built under R version 4.1.1

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(ggplot2)
theme_set(theme_bw())
pdb_data_export <- read.csv("data_export_summary.csv")
knitr::kable(pdb_data_export)
```

Molecular.Type	X.ray	EM	NMR	Multiple.methods	Neutron	Other	Total
Protein (only)	152,914	9,495	12,121	191	72	32	174,825
Protein/Oligosaccharide	9,008	1,663	32	7	1	0	10,711
Protein/NA	8,069	2,949	282	6	0	0	11,306
Nucleic acid (only)	2,602	78	1,434	12	2	1	4,129
Other	163	9	31	0	0	0	203
Oligosaccharide (only)	11	0	6	1	0	4	22

```r
# above makes table prettier
```

> Q1. What % of structures are solved by Xray and EM?

```r
# doesnt work: pdb_data_export$X.ray <- as.numeric(pdb_data_export$X.ray)

# Xray structures in database
n.xray <- sum(as.numeric( gsub(",", "", pdb_data_export$X.ray) ))
```

```r
# EM structures in database
n.EM <- sum(as.numeric( gsub(",", "", pdb_data_export$EM)))
```

```r
n.total <- sum(as.numeric( gsub(",", "", pdb_data_export$Total)))
```

Lets make a function to automate counting the number of xray/EM structures:

```r
rm.comma <- function(x) {
    sum(as.numeric( gsub(",", "", x) ) )
}
```

Percent of Xray structures

```r
percent_xray_fun <- 100*rm.comma(pdb_data_export$X.ray)/rm.comma(pdb_data_export$Total)
```

85.8699974 % of structures are solved by Xray.

Percent of EM structures

```
percent_EM_fun <- 100*rm.comma(pdb_data_export$EM)/rm.comma(pdb_data_export$Total)
```

7.0548122 % of structures are solved by EM.

> Q2. What proportion of structures in PDB are protein?

```
n.total <- sum(as.numeric( gsub(",", "", pdb_data_export$Total)))
prot_total <- as.numeric(gsub(",", "", pdb_data_export$Total[1]))
percent_prot <- 100*prot_total/n.total
```

86.8928806 % of the PDB database are proteins.

> Q3. How many HIV-1 protease sturctures are in PDB?

There are >200,000 results searching for HIV-1 protease! Don't search by text/name, much better to search by sequence/structure.

Mol* Viewer

Here's the HIV-1 image

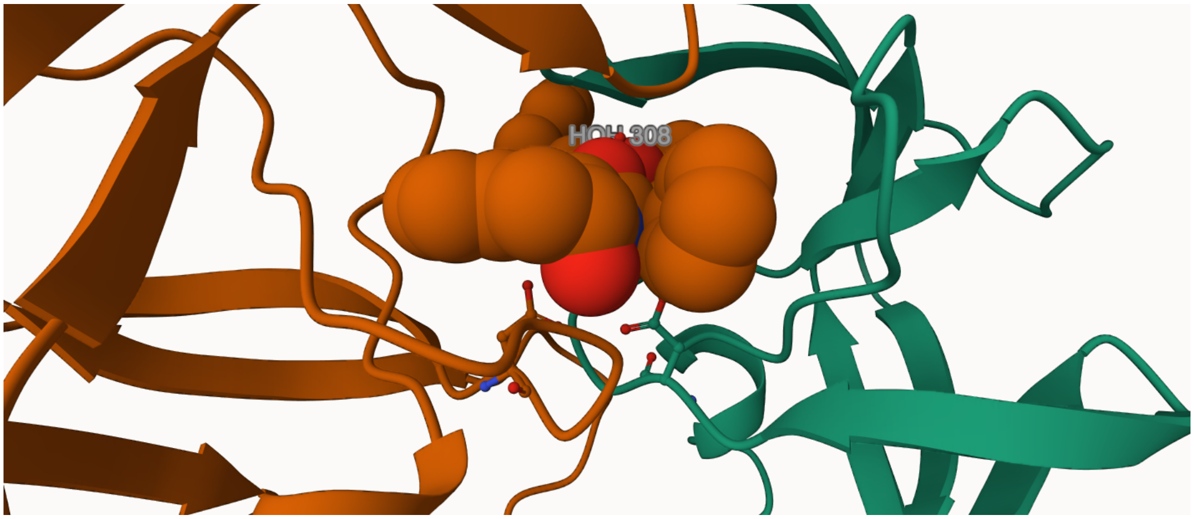


Figure 1: HIV-1 protease with inhibitor and important interactions highlighted.

Bio3D

```
library(bio3d)
```

```
Warning: package 'bio3d' was built under R version 4.1.3
```

```
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
```

> Q7. How many residues?

198 residues (from above readout).

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

HOH

Q9. How many protein chains?

There are two chains (chain A and chain B).

```
attributes(pdb)
```

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

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

Atoms

```
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>
```

Residue of the first atom:

```
pdb$atom$resid[1]
```

```
[1] "PRO"
```

```
# or pdb$atom["resid"]
```

Convert residue to 1 letter code

```r
aa321(pdb$atom$resid[1])
```

```
[1] "P"
```

Predicting Functional Motions with Normal Mode Analysis (NMA)

NMA predicts flexibility based on a static structure

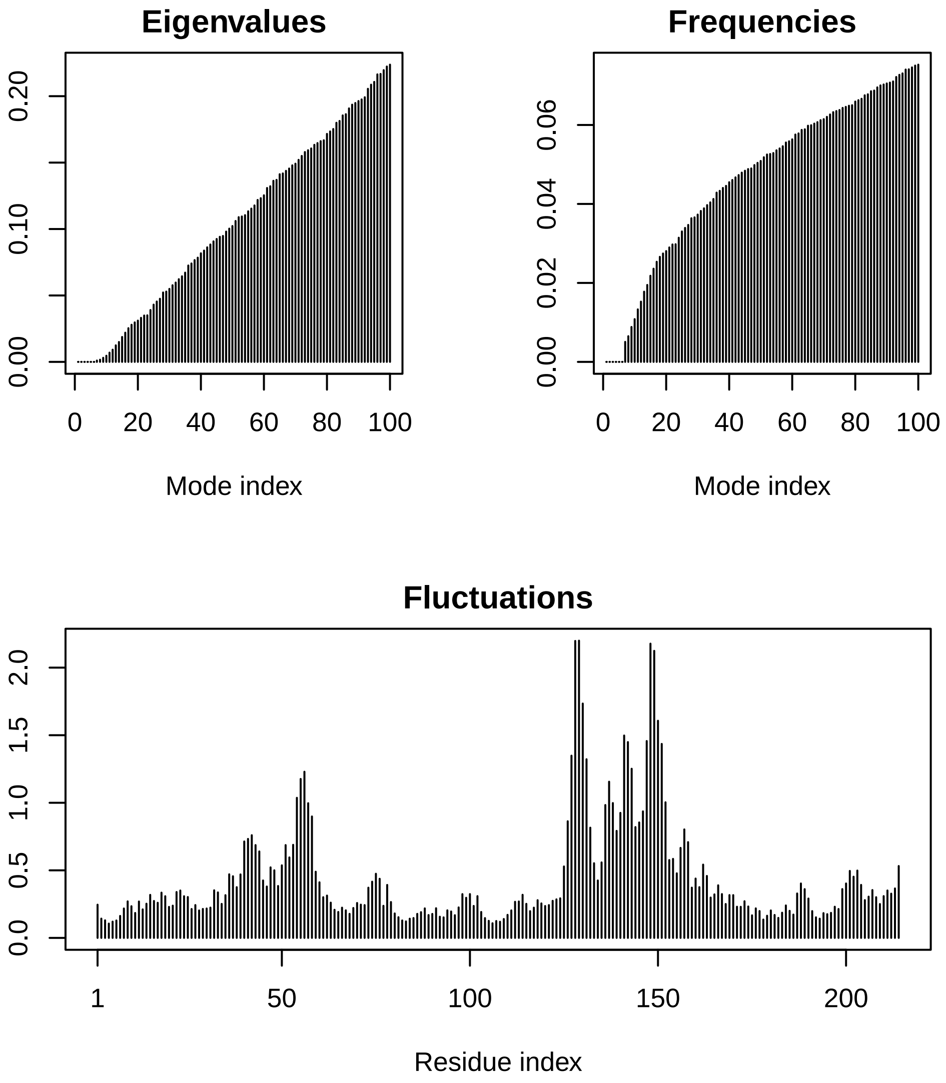
```r
adk <- read.pdb("6s36")
```

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

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

```
 Building Hessian...        Done in 0.11 seconds.
 Diagonalizing Hessian...   Done in 0.97 seconds.
```

```r
plot(m)
```



The third plot (Fluctuations) has peaks that show the most flexible regions of the protein.

Display motion:

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

Comparitive Structure Analysis

```
Bioconductor version 3.14 (BiocManager 1.30.19), R 4.1.0 (2021-05-18)
```

```
Installation paths not writeable, unable to update packages
  path: C:/Program Files/R/R-4.1.0/library
  packages:
    boot, class, cluster, codetools, foreign, lattice, MASS, Matrix, mgcv,
    nlme, nnet, rpart, spatial, survival
```

```
Old packages: 'amap', 'ashr', 'babelgene', 'BayesFactor', 'bayestestR',
  'bbmle', 'bdsmatrix', 'BH', 'bit', 'blob', 'broom', 'Cairo', 'clipr',
  'colorDF', 'colorspace', 'correlation', 'cpp11', 'crayon', 'curl',
  'data.table', 'datawizard', 'DBI', 'dbplyr', 'deSolve', 'DiffBind', 'digest',
  'dplyr', 'dtplyr', 'effectsize', 'evaluate', 'extrafont', 'fansi', 'farver',
  'fontawesome', 'forcats', 'formatR', 'fs', 'gargle', 'generics',
  'GenomeInfoDb', 'gert', 'ggbeeswarm', 'ggforce', 'ggplot2', 'ggrepel',
  'ggsignif', 'ggstatsplot', 'gh', 'gitcreds', 'glue', 'gmp', 'googlesheets4',
  'gplots', 'gtable', 'gtools', 'haven', 'highr', 'hms', 'htmlwidgets', 'httr',
  'hwriter', 'insight', 'irlba', 'isoband', 'jpeg', 'jsonlite', 'knitr',
  'latticeExtra', 'lifecycle', 'limma', 'locfit', 'lubridate', 'magrittr',
  'maps', 'markdown', 'MatrixModels', 'matrixStats', 'mc2d', 'mixsqp',
  'modelr', 'msigdbr', 'openssl', 'packrat', 'paletteer', 'palmerpenguins',
  'parameters', 'patchwork', 'pbapply', 'performance', 'pillar', 'plotly',
  'plotwidgets', 'plyr', 'PMCMRplus', 'png', 'polyclip', 'prismatic', 'proj4',
  'ps', 'purrr', 'ragg', 'RColorBrewer', 'Rcpp', 'RcppArmadillo', 'RcppEigen',
  'RCurl', 'readr', 'readxl', 'reprex', 'reshape', 'restfulr', 'rmarkdown',
  'Rmpfr', 'rprojroot', 'rsconnect', 'RSQLite', 'rstudioapi', 'Rttf2pt1',
  'rvest', 'S4Vectors', 'sass', 'scales', 'sourcetools', 'statsExpressions',
  'stringi', 'stringr', 'sys', 'systemfonts', 'systemPipeR', 'tibble', 'tidyr',
  'tidyselect', 'tidyverse', 'tinytex', 'tmod', 'tweenr', 'tzdb', 'utf8',
  'uuid', 'vctrs', 'viridisLite', 'vroom', 'whisker', 'WRS2', 'xfun', 'XML',
  'yaml', 'zip'
```

```
# devtools::install_bitbucket("Grantlab/bio3d-view")
```

```
Warning: package 'msa' was built under R version 4.1.1
```

```
Loading required package: Biostrings
```

```
Warning: package 'Biostrings' was built under R version 4.1.1

Loading required package: BiocGenerics

Warning: package 'BiocGenerics' was built under R version 4.1.1

Attaching package: 'BiocGenerics'

The following objects are masked from 'package:dplyr':

    combine, intersect, setdiff, union

The following objects are masked from 'package:stats':

    IQR, mad, sd, var, xtabs

The following objects are masked from 'package:base':

    anyDuplicated, append, as.data.frame, basename, cbind, colnames,
    dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
    grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
    order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
    rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
    union, unique, unsplit, which.max, which.min

Loading required package: S4Vectors

Warning: package 'S4Vectors' was built under R version 4.1.2

Loading required package: stats4

Attaching package: 'S4Vectors'

The following objects are masked from 'package:dplyr':

    first, rename

The following objects are masked from 'package:base':

    expand.grid, I, unname
```

```
Loading required package: IRanges

Warning: package 'IRanges' was built under R version 4.1.1

Attaching package: 'IRanges'

The following object is masked from 'package:bio3d':

    trim

The following objects are masked from 'package:dplyr':

    collapse, desc, slice

The following object is masked from 'package:grDevices':

    windows

Loading required package: XVector

Warning: package 'XVector' was built under R version 4.1.1

Loading required package: GenomeInfoDb

Warning: package 'GenomeInfoDb' was built under R version 4.1.1

Attaching package: 'Biostrings'

The following object is masked from 'package:bio3d':

    mask

The following object is masked from 'package:base':

    strsplit
```

```r
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?

214 amino acids.

Search against pdb database for related structures:

```r
#b <- blast.pdb(aa)
hits <- NULL
hits$pdb.id <- c('1AKE_A','6S36_A','6RZE_A','3HPR_A','1E4V_A','5EJE_A','1E4Y_A','3X2S_A','
```

Plot PDB Blast Hits:

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

Plot showing similar results to BLAST search result in plots (E value, identity, length, etc). Notice -log(Evalue) is plotted, so the highest values (black points) are what we want. The output automatically shows a cutoff point (dashed line). 16 hits passed.

Our top hits

```
hits$pdb.id
```

```
 [1] "1AKE_A" "6S36_A" "6RZE_A" "3HPR_A" "1E4V_A" "5EJE_A" "1E4Y_A" "3X2S_A"
 [9] "6HAP_A" "6HAM_A" "4K46_A" "3GMT_A" "4PZL_A"
```

Downloading structures

```
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%
```

Align and superposition

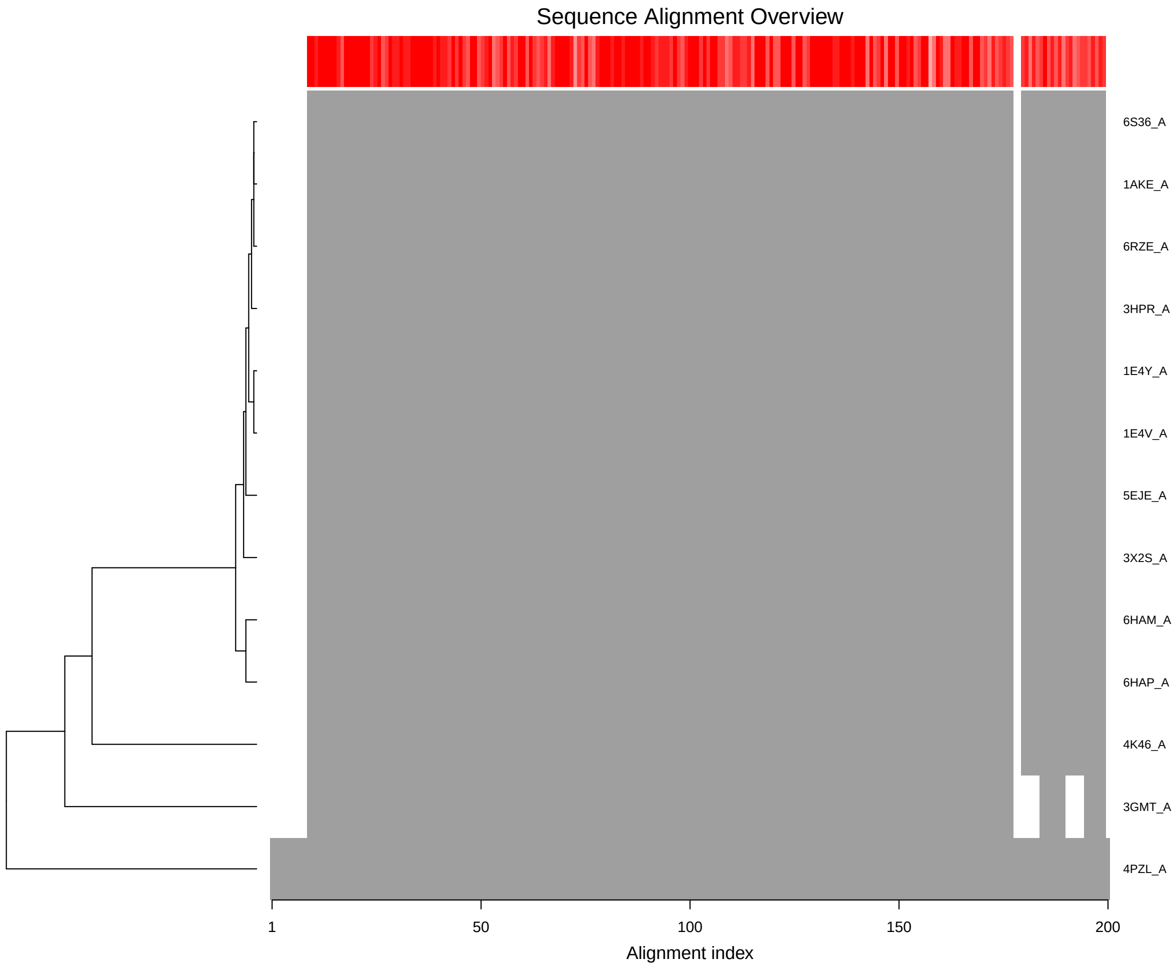
```
pdbs <- pdbaln(files, fit = TRUE, exefile="msa")
```

```
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
```

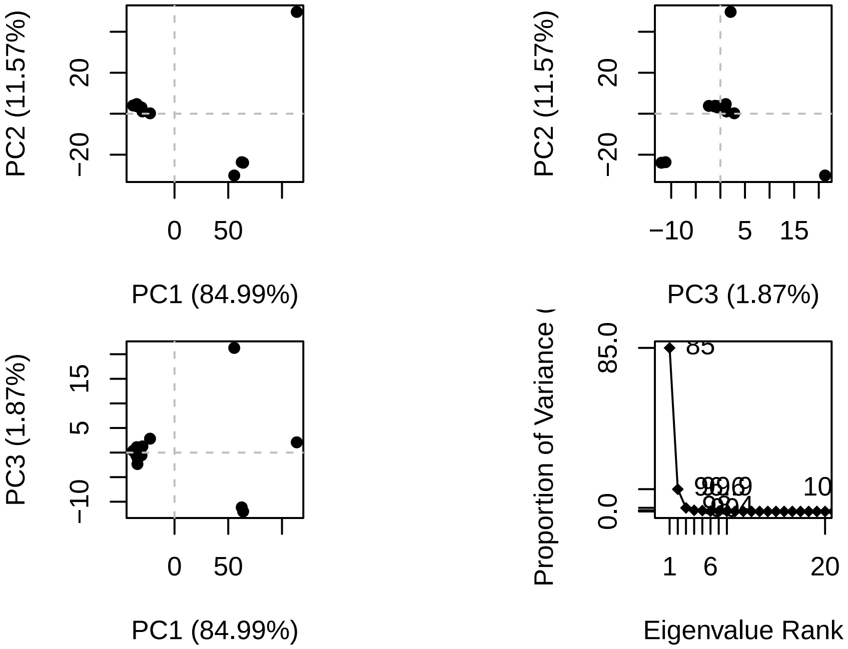
Drawing it:

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

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



Do PCA

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

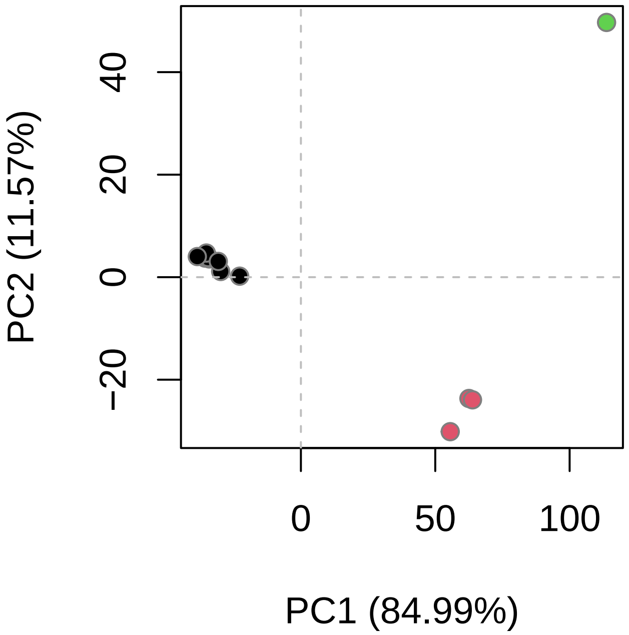


Trajectory Animation

```r
rd <- rmsd(pdbs)
```

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

```r
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 PC

```
pc1 <- mktrj(pc.xray, pc=1, file="pc_1.pdb")
```

Load the output on Mol. *The resulting animation has a dotted line in one portion representing some sequence that is missing in one of the models. Mol* doesn't just want to guess/average based on the other structures, so it puts a dotted line instead.