Class 10 Structural Bioinformatics

Eli Haddad A16308227

1) Introduction to the RCSB Protein Data Bank (PDB)

First let's see what is in the PDB database - the main repository of protein structures.

Downloaded composition stats from: https://www.rcsb.org/stats/summary

For context: Release 2023_24 of 13-Sep-2023 of UniProtKB/TrEMBL contains 251600,768 sequence entries. The PDB only contains 183,201

```r
stats <- read.csv("Data Export Summary.csv", row.names = 1)
stats
```

```
                          X.ray     EM    NMR Multiple.methods Neutron Other
Protein (only)          158,844 11,759 12,296              197      73    32
Protein/Oligosaccharide   9,260  2,054     34                8       1     0
Protein/NA                8,307  3,667    284                7       0     0
Nucleic acid (only)       2,730    113  1,467               13       3     1
Other                       164      9     32                0       0     0
Oligosaccharide (only)       11      0      6                1       0     4
                          Total
Protein (only)          183,201
Protein/Oligosaccharide  11,357
Protein/NA               12,265
Nucleic acid (only)       4,327
Other                       205
Oligosaccharide (only)       22
```

There is a problem there due to the commas in the numbers. This causes R to treat them as characters.

```r
x <- stats$X.ray
x
```

```
[1] "158,844" "9,260"   "8,307"   "2,730"   "164"     "11"
```

```r
as.numeric(gsub(",","", x))
```

```
[1] 158844   9260   8307   2730    164     11
```

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

```r
rm.comma(stats$EM)
```

```
[1] 11759  2054  3667   113     9     0
```

I can use `apply()` to fix the whole table…

```r
pdbstats <- apply(stats, 2, rm.comma)
rownames(pdbstats) <- rownames(stats)
head(pdbstats)
```

```
                         X.ray    EM   NMR Multiple.methods Neutron Other
Protein (only)          158844 11759 12296              197      73    32
Protein/Oligosaccharide   9260  2054    34                8       1     0
Protein/NA                8307  3667   284                7       0     0
Nucleic acid (only)       2730   113  1467               13       3     1
Other                      164     9    32                0       0     0
Oligosaccharide (only)      11     0     6                1       0     4
                         Total
Protein (only)          183201
Protein/Oligosaccharide  11357
Protein/NA               12265
Nucleic acid (only)       4327
Other                      205
Oligosaccharide (only)      22
```

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

```
totals <- apply(pdbstats, 2, sum)
round(totals / totals["Total"] * 100, 2)
```

```
           X.ray               EM              NMR Multiple.methods 
           84.83             8.33             6.68             0.11 
         Neutron            Other            Total 
            0.04             0.02           100.00 
```

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

```
round(pdbstats[, "Total"] / sum(pdbstats[,"Total"]) * 100, 2)
```

```
         Protein (only) Protein/Oligosaccharide              Protein/NA 
                  86.67                    5.37                    5.80 
    Nucleic acid (only)                   Other  Oligosaccharide (only) 
                   2.05                    0.10                    0.01 
```

> 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?

SKIPPED for time!!

Protein structures in PDB as a function of UniProt sequences.

```
round((pdbstats[1,"Total"] /251600768) * 100, 2)
```

```
[1] 0.07
```

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

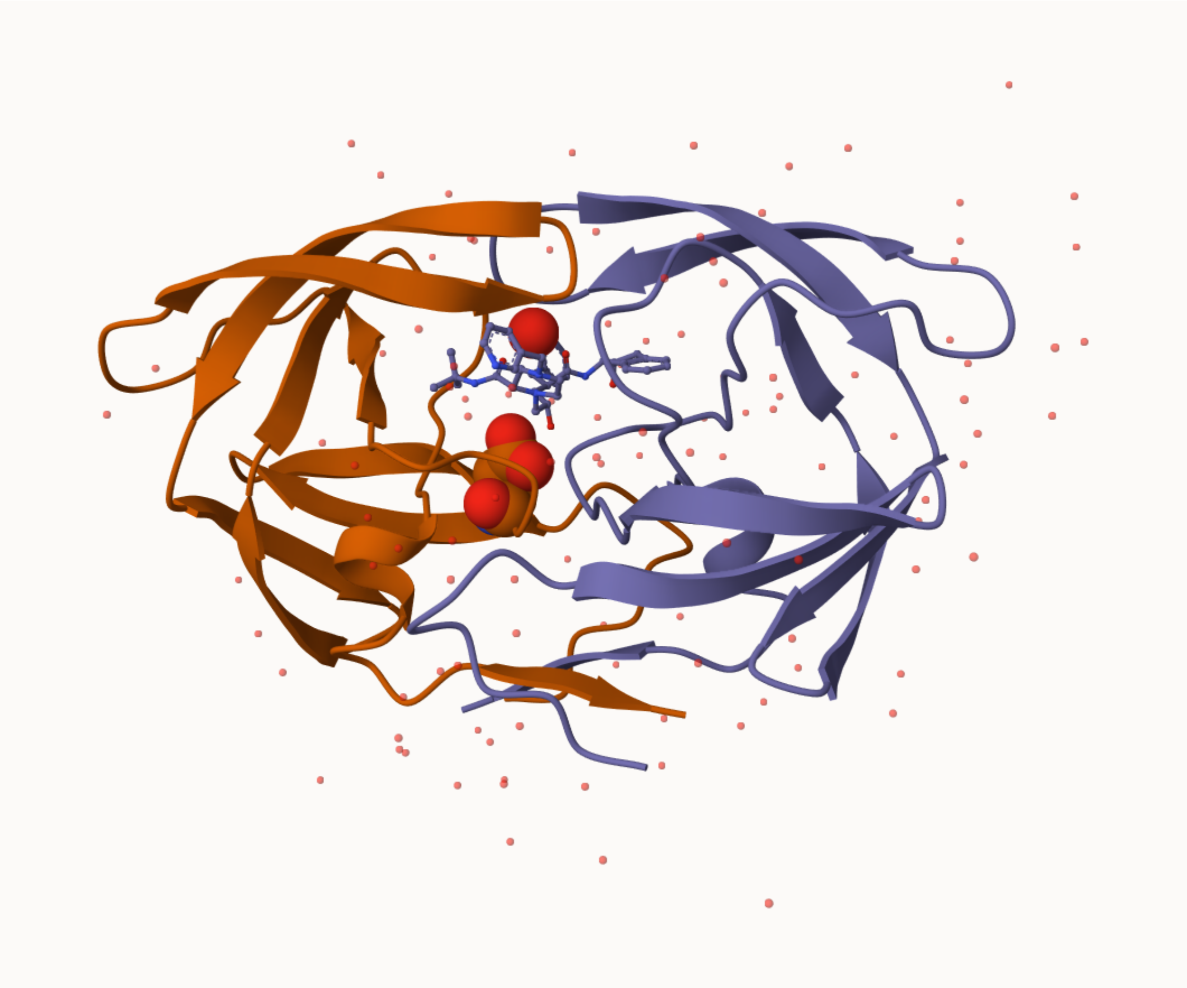
The resolution (2 Angstroms) is not small enough in order to capture the small size of the hydrogen atoms. That is why hydrogen is not seen anywhere in these structures. You need 1 Angstrom or better to see these smaller atoms.

> 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 called HOH 308. It is important for making direct interactions with the ligand and with the protein.

> 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.

Here is a lovely figure of HIP-Pr with the catalytic ASP residues, the MK1 compound and the all important water 308



The bio3d package for structural bioinformatics

```
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
```

```
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>
```

Predicting functional motions of a single structure

Let's finish today with a bioinformatics calculation to predict the functional motions of a PDB structure.

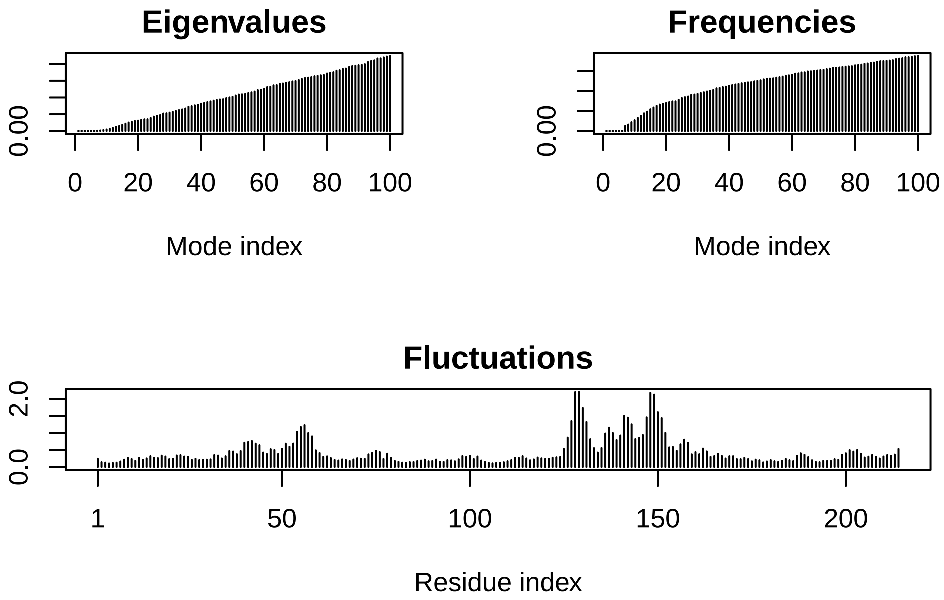
```
adk <- read.pdb("6s36")
```

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

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

```
 Building Hessian...        Done in 0.013 seconds.
 Diagonalizing Hessian...   Done in 0.275 seconds.
```

```
plot(m)
```



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

4. Comparative structure analysis of Adenylate Kinase

We need some packages for todays class. These include `bio3d` and `msa`

The `msa` package is from BioConductor. These packages focus on genomics type work and are managed by the `BiocManager` package.

Install `install.packages("BiocManager")` and then `BiocManager:install("msa")` all entered in the R "brain" console.

```r
library(bio3d)

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
```

Now I can search for the PDB database for related sequences:

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

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

```r
#attributes(b)
#head(b)
```

These are the related structures in the PDB database that we found via a BLAST search…

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

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

The `msa` package is found only in BioConductor and not CRAN.

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

The `Grantlab/bio3d-view` is installed using devtools, so this is not found on BioConductor or CRAN.

> 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?

There are 214 amino acids in this sequence.

Side-note: Lets annotate these structures (in other words find out what they are, what species they are from, stuff about the experiment they were solve in etc.)

For this we can use the `pdb.annotate()`

```r
anno <- pdb.annotate(hits$pdb.id)
```

```r
#attributes(anno)
head(anno)
```

```
       structureId chainId macromoleculeType chainLength experimentalTechnique
1AKE_A        1AKE       A           Protein         214                 X-ray
6S36_A        6S36       A           Protein         214                 X-ray
6RZE_A        6RZE       A           Protein         214                 X-ray
3HPR_A        3HPR       A           Protein         214                 X-ray
1E4V_A        1E4V       A           Protein         214                 X-ray
5EJE_A        5EJE       A           Protein         214                 X-ray
       resolution       scopDomain                                        pfam
1AKE_A       2.00 Adenylate kinase Adenylate kinase, active site lid (ADK_lid)
6S36_A       1.60             <NA> Adenylate kinase, active site lid (ADK_lid)
6RZE_A       1.69             <NA> Adenylate kinase, active site lid (ADK_lid)
3HPR_A       2.00             <NA> Adenylate kinase, active site lid (ADK_lid)
1E4V_A       1.85 Adenylate kinase Adenylate kinase, active site lid (ADK_lid)
```

```
5EJE_A       1.90             <NA> Adenylate kinase, active site lid (ADK_lid)
               ligandId                                       ligandName
1AKE_A              AP5                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
6S36_A CL (3),NA,MG (2)    CHLORIDE ION (3),SODIUM ION,MAGNESIUM ION (2)
6RZE_A    NA (3),CL (2)                  SODIUM ION (3),CHLORIDE ION (2)
3HPR_A              AP5                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
1E4V_A              AP5                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
5EJE_A           AP5,CO BIS(ADENOSINE)-5'-PENTAPHOSPHATE,COBALT (II) ION
                                       source
1AKE_A                       Escherichia coli
6S36_A                       Escherichia coli
6RZE_A                       Escherichia coli
3HPR_A                  Escherichia coli K-12
1E4V_A                       Escherichia coli
5EJE_A Escherichia coli O139:H28 str. E24377A

1AKE_A STRUCTURE OF THE COMPLEX BETWEEN ADENYLATE KINASE FROM ESCHERICHIA COLI AND THE INHIBI
6S36_A
6RZE_A
3HPR_A
1E4V_A
5EJE_A                                                                                  Crys
                                                    citation rObserved  rFree
1AKE_A               Muller, C.W., et al. J Mol Biol (1992)    0.1960     NA
6S36_A                Rogne, P., et al. Biochemistry (2019)    0.1632 0.2356
6RZE_A                Rogne, P., et al. Biochemistry (2019)    0.1865 0.2350
3HPR_A Schrank, T.P., et al. Proc Natl Acad Sci U S A (2009)    0.2100 0.2432
1E4V_A                 Muller, C.W., et al. Proteins (1993)    0.1960     NA
5EJE_A Kovermann, M., et al. Proc Natl Acad Sci U S A (2017)    0.1889 0.2358
        rWork spaceGroup
1AKE_A 0.1960  P 21 2 21
6S36_A 0.1594    C 1 2 1
6RZE_A 0.1819    C 1 2 1
3HPR_A 0.2062  P 21 21 2
1E4V_A 0.1960  P 21 2 21
5EJE_A 0.1863  P 21 2 21
```

Now we can download all these structures for further analysis with the `get.pdb()` function.

```
# Download related 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%
```

Now we have all these related structures we can Align and Superpose

```
# 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
```

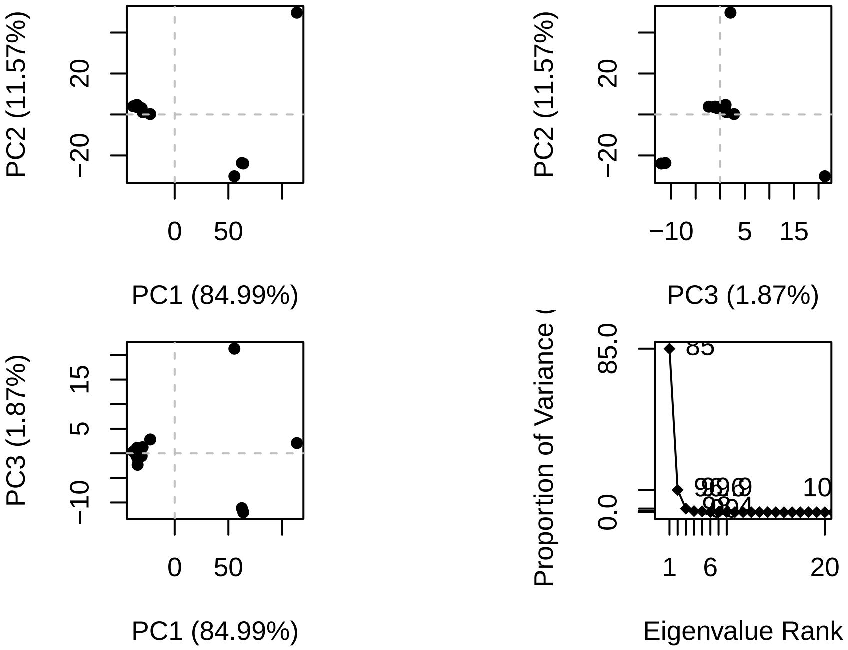
```
[Truncated_Name:1]1AKE_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:2]6S36_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:3]6RZE_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:4]3HPR_A.pdb    ----------MRIILLGAPGAGKGTQAQFIMEKYGIPQIS
[Truncated_Name:5]1E4V_A.pdb    ----------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
[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
[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
```

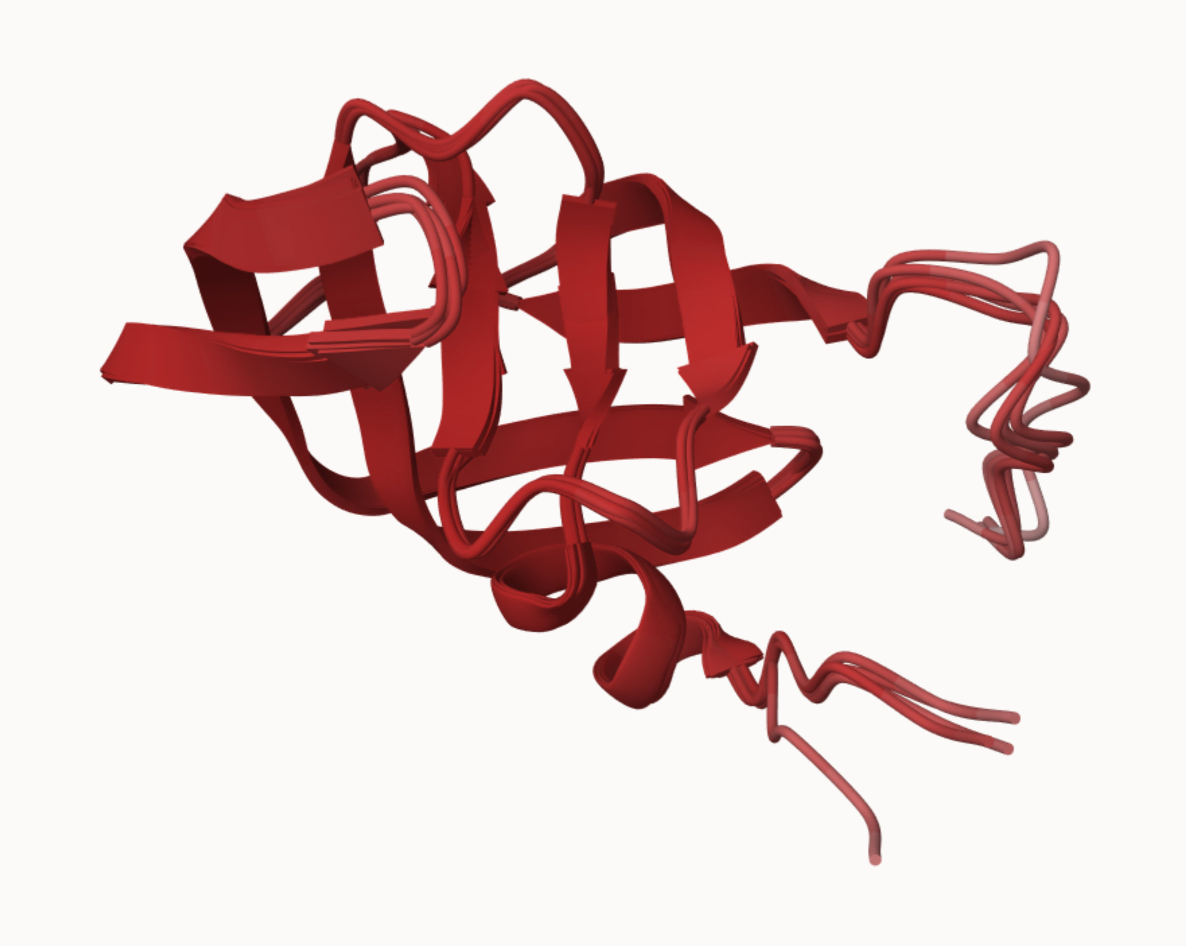
Principal Component Analysis

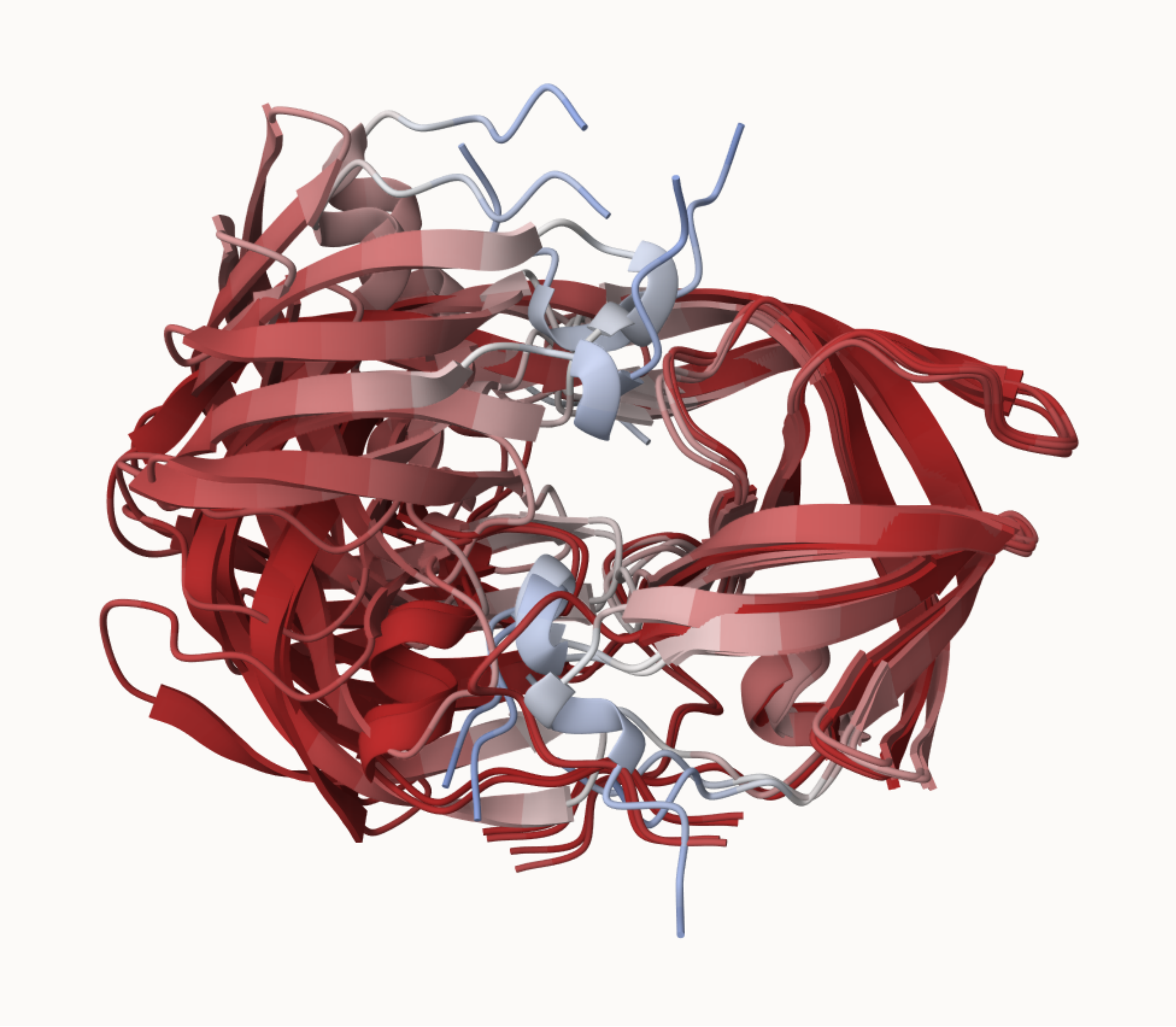
```
# Perform PCA
pc.xray <- pca(pdbs)
plot(pc.xray)
```



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

Monomer





Dimer

# 8. Custom analysis of resulting models

```
results_dir <- "hivpr_dimer_23119"
pdb_files <- list.files(path=results_dir,
                        pattern="*.pdb",
                        full.names = TRUE)
pdb_files
```

```
[1] "hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_se
[2] "hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_se
[3] "hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_se
[4] "hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_se
[5] "hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_se
```

```r
library(bio3d)

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

```
Reading PDB files:
hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_00
hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_00
hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_00
hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_00
hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_00
.....

Extracting sequences

pdb/seq: 1   name: hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_001_alphafold2_multimer
pdb/seq: 2   name: hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_002_alphafold2_multimer
pdb/seq: 3   name: hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_003_alphafold2_multimer
pdb/seq: 4   name: hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_004_alphafold2_multimer
pdb/seq: 5   name: hivpr_dimer_23119/hivpr_dimer_23119_unrelaxed_rank_005_alphafold2_multimer
```

```r
pdbs
```

```
                               1        .         .         .         .         50 
[Truncated_Name:1]hivpr_dime   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:2]hivpr_dime   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:3]hivpr_dime   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:4]hivpr_dime   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:5]hivpr_dime   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
                               ************************************************** 
                               1        .         .         .         .         50 

                              51        .         .         .         .         100 
[Truncated_Name:1]hivpr_dime   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:2]hivpr_dime   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:3]hivpr_dime   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:4]hivpr_dime   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:5]hivpr_dime   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
                               ************************************************** 
                              51        .         .         .         .         100 
```

```
                             101       .         .         .         .         150
[Truncated_Name:1]hivpr_dime   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:2]hivpr_dime   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:3]hivpr_dime   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:4]hivpr_dime   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:5]hivpr_dime   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
                               **************************************************
                             101       .         .         .         .         150

                             151       .         .         .         .       198
[Truncated_Name:1]hivpr_dime   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:2]hivpr_dime   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:3]hivpr_dime   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:4]hivpr_dime   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:5]hivpr_dime   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
                               ************************************************
                             151       .         .         .         .       198

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

Class:
  pdbs, fasta

Alignment dimensions:
  5 sequence rows; 198 position columns (198 non-gap, 0 gap)

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

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

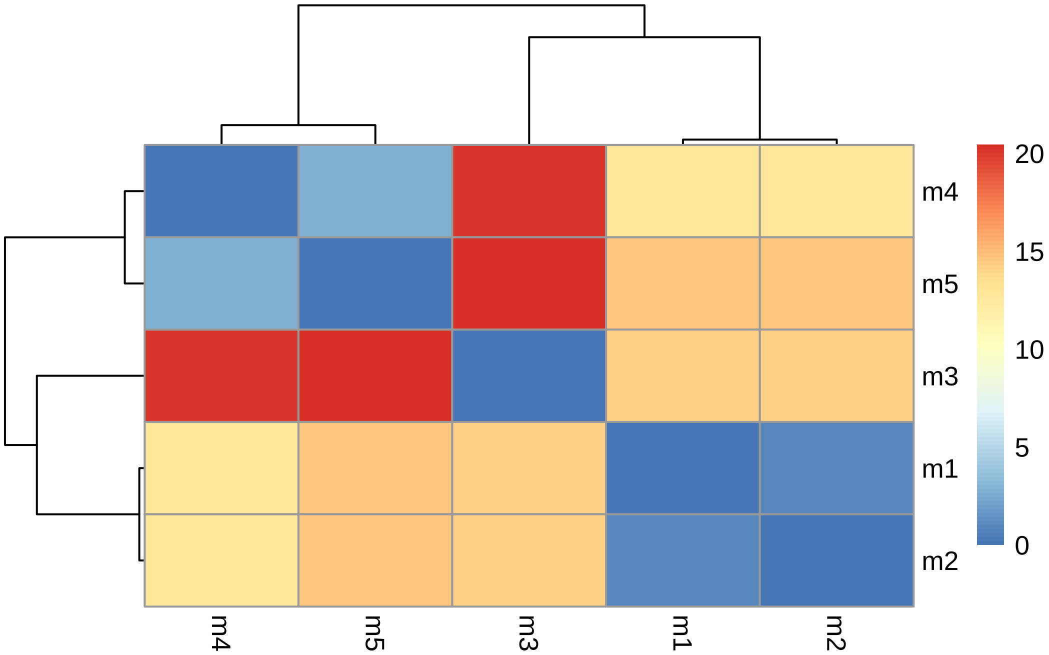
```
Warning in rmsd(pdbs): No indices provided, using the 198 non NA positions
```

```r
range(rd)
```

```
[1]  0.000 20.431
```

```r
library(pheatmap)
```

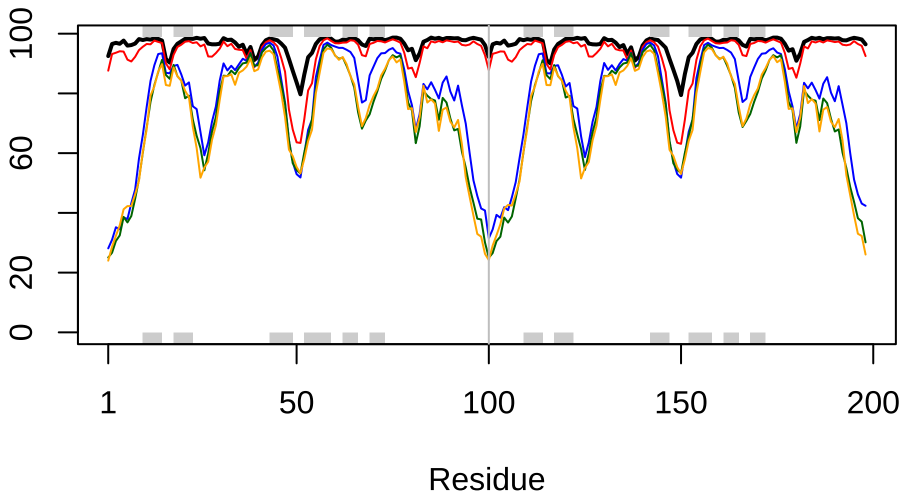
```
colnames(rd) <- paste0("m",1:5)
rownames(rd) <- paste0("m",1:5)
pheatmap(rd)
```



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

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
/var/folders/97/bkcl6mms5d74hwhqjg57_51m0000gn/T//Rtmpvao892/1hsg.pdb exists.
Skipping download
```

```
plotb3(pdbs$b, typ="l", lwd=2, sse=pdb)
points(pdbs$b[2,], typ="l", col="red")
points(pdbs$b[3,], typ="l", col="blue")
points(pdbs$b[4,], typ="l", col="darkgreen")
points(pdbs$b[5,], typ="l", col="orange")
abline(v=100, col="gray")
```



```
core <- core.find(pdbs)
```

```
core size 197 of 198  vol = 6154.839
core size 196 of 198  vol = 5399.676
core size 195 of 198  vol = 5074.795
core size 194 of 198  vol = 4802.518
core size 193 of 198  vol = 4520.256
core size 192 of 198  vol = 4305.362
core size 191 of 198  vol = 4089.792
core size 190 of 198  vol = 3886.145
core size 189 of 198  vol = 3758.321
core size 188 of 198  vol = 3620.18
core size 187 of 198  vol = 3496.698
core size 186 of 198  vol = 3389.985
core size 185 of 198  vol = 3320.114
core size 184 of 198  vol = 3258.683
core size 183 of 198  vol = 3208.591
core size 182 of 198  vol = 3156.736
core size 181 of 198  vol = 3141.668
core size 180 of 198  vol = 3136.574
core size 179 of 198  vol = 3155.52
core size 178 of 198  vol = 3185.362
core size 177 of 198  vol = 3204.487
core size 176 of 198  vol = 3211.978
core size 175 of 198  vol = 3234.993
core size 174 of 198  vol = 3244.062
core size 173 of 198  vol = 3237.845
core size 172 of 198  vol = 3218.77
core size 171 of 198  vol = 3180.743
core size 170 of 198  vol = 3130.369
core size 169 of 198  vol = 3067.881
core size 168 of 198  vol = 2989.546
core size 167 of 198  vol = 2928.272
core size 166 of 198  vol = 2851.193
core size 165 of 198  vol = 2780.877
core size 164 of 198  vol = 2708.433
core size 163 of 198  vol = 2636.516
core size 162 of 198  vol = 2563.25
core size 161 of 198  vol = 2478.024
core size 160 of 198  vol = 2404.793
core size 159 of 198  vol = 2330.997
core size 158 of 198  vol = 2250.477
core size 157 of 198  vol = 2159.432
core size 156 of 198  vol = 2070.759
core size 155 of 198  vol = 1983.579
core size 154 of 198  vol = 1917.913
core size 153 of 198  vol = 1842.556
core size 152 of 198  vol = 1775.398
core size 151 of 198  vol = 1695.133
core size 150 of 198  vol = 1632.173
core size 149 of 198  vol = 1570.391
core size 148 of 198  vol = 1497.238
core size 147 of 198  vol = 1434.802
core size 146 of 198  vol = 1367.706
core size 145 of 198  vol = 1302.596
core size 144 of 198  vol = 1251.985
core size 143 of 198  vol = 1207.976
core size 142 of 198  vol = 1167.112
core size 141 of 198  vol = 1118.27
core size 140 of 198  vol = 1081.664
core size 139 of 198  vol = 1029.75
core size 138 of 198  vol = 981.766
core size 137 of 198  vol = 944.446
core size 136 of 198  vol = 899.224
core size 135 of 198  vol = 859.402
core size 134 of 198  vol = 814.694
core size 133 of 198  vol = 771.862
core size 132 of 198  vol = 733.807
core size 131 of 198  vol = 702.053
core size 130 of 198  vol = 658.757
core size 129 of 198  vol = 622.574
core size 128 of 198  vol = 578.29
core size 127 of 198  vol = 543.07
core size 126 of 198  vol = 510.934
core size 125 of 198  vol = 481.595
core size 124 of 198  vol = 464.672
core size 123 of 198  vol = 451.721
core size 122 of 198  vol = 430.417
core size 121 of 198  vol = 409.141
core size 120 of 198  vol = 378.942
core size 119 of 198  vol = 348.325
core size 118 of 198  vol = 324.738
core size 117 of 198  vol = 312.394
core size 116 of 198  vol = 300.89
core size 115 of 198  vol = 279.976
core size 114 of 198  vol = 263.434
core size 113 of 198  vol = 250.263
core size 112 of 198  vol = 229.592
core size 111 of 198  vol = 209.929
core size 110 of 198  vol = 196.379
core size 109 of 198  vol = 180.628
core size 108 of 198  vol = 167.088
core size 107 of 198  vol = 155.875
core size 106 of 198  vol = 142.595
core size 105 of 198  vol = 128.924
core size 104 of 198  vol = 114.054
core size 103 of 198  vol = 100.936
core size 102 of 198  vol = 90.431
core size 101 of 198  vol = 81.972
core size 100 of 198  vol = 74.017
core size 99 of 198  vol = 66.855
core size 98 of 198  vol = 59.525
core size 97 of 198  vol = 52.263
core size 96 of 198  vol = 43.699
core size 95 of 198  vol = 35.813
core size 94 of 198  vol = 28.888
core size 93 of 198  vol = 20.692
core size 92 of 198  vol = 14.975
```

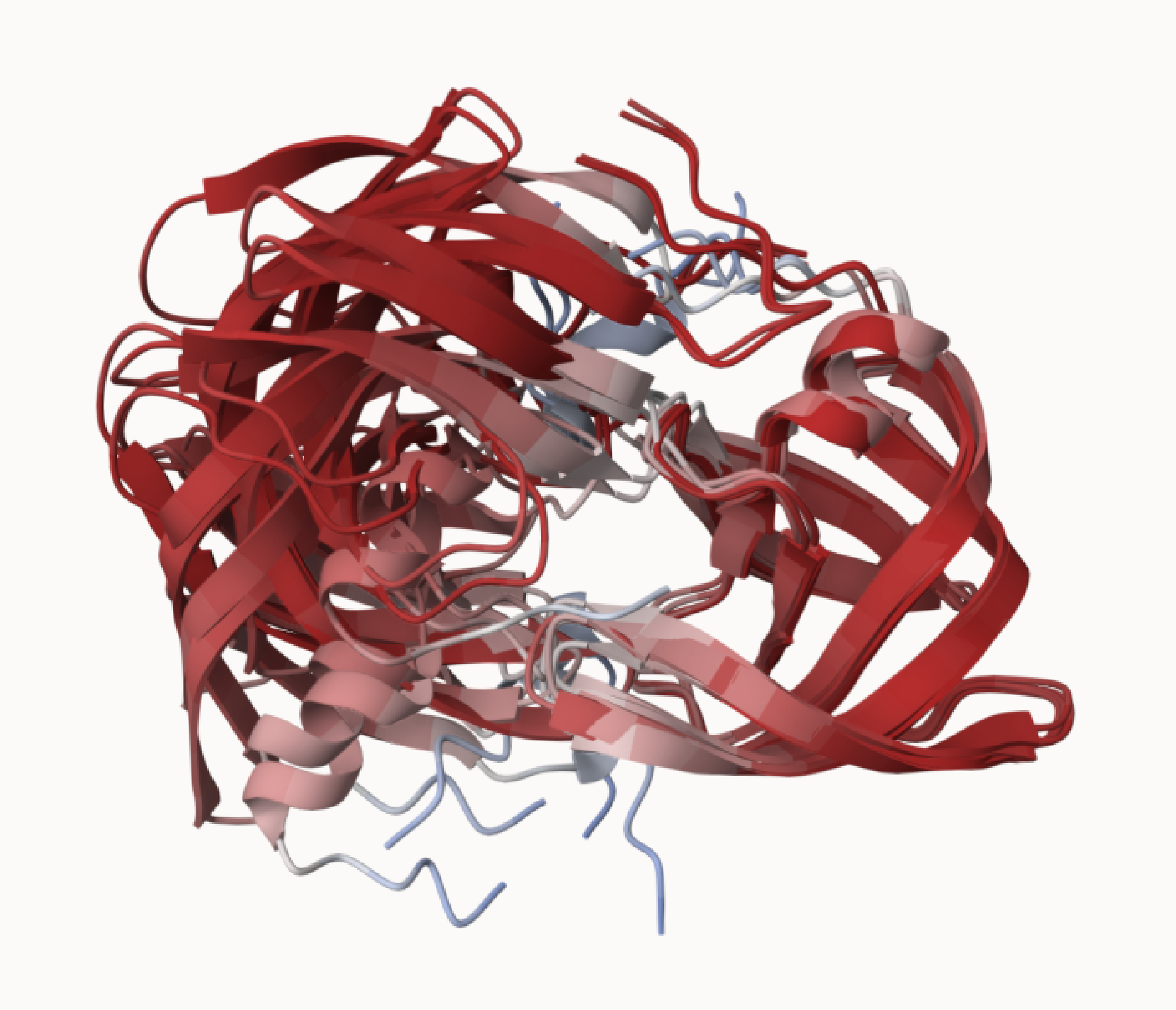
```
 core size 91 of 198  vol = 9.146
 core size 90 of 198  vol = 5.232
 core size 89 of 198  vol = 3.53
 core size 88 of 198  vol = 2.657
 core size 87 of 198  vol = 1.998
 core size 86 of 198  vol = 1.333
 core size 85 of 198  vol = 1.141
 core size 84 of 198  vol = 1.012
 core size 83 of 198  vol = 0.891
 core size 82 of 198  vol = 0.749
 core size 81 of 198  vol = 0.618
 core size 80 of 198  vol = 0.538
 core size 79 of 198  vol = 0.479
 FINISHED: Min vol ( 0.5 ) reached
```

```
core.inds <- print(core, vol=0.5)
```

```
# 80 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1    10  25     16
2    27  48     22
3    53  94     42
```

```
xyz <- pdbfit(pdbs, core.inds, outpath="corefit_structures")
```

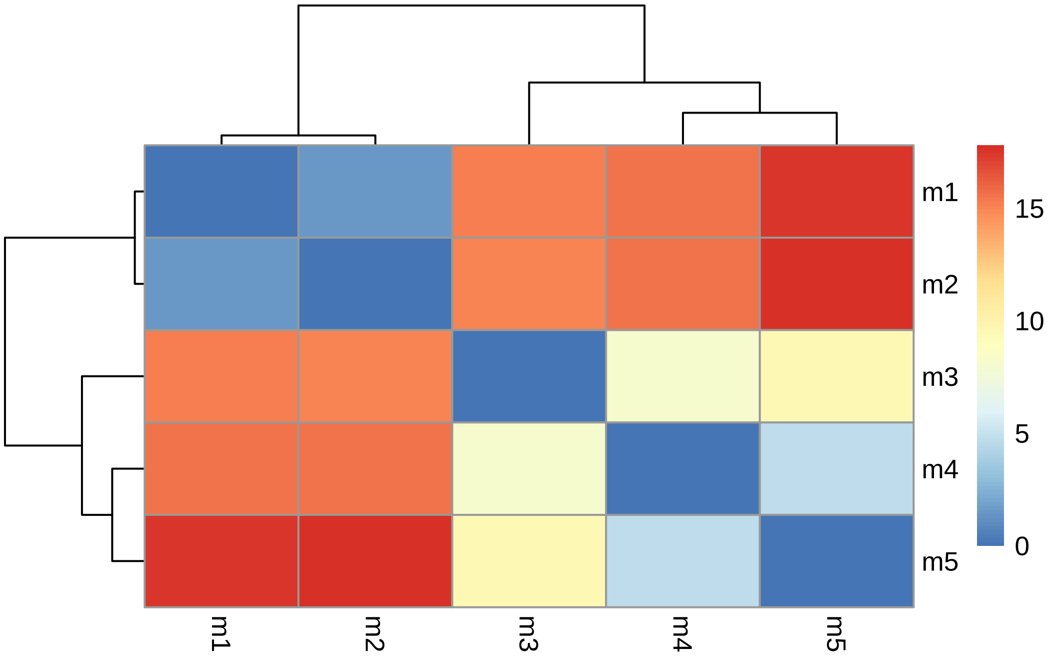
Core superposed structure



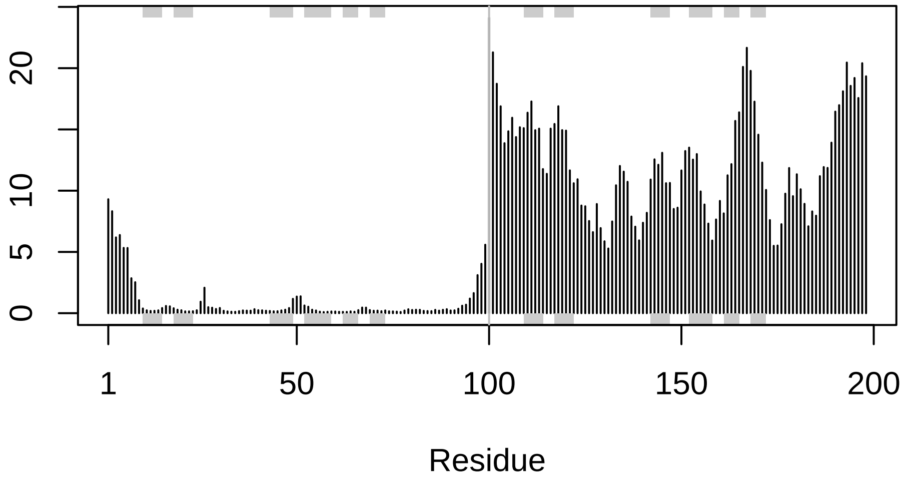
```
rd <- rmsd(xyz)
```

```
Warning in rmsd(xyz): No indices provided, using the 198 non NA positions
```

```
colnames(rd) <- paste0("m",1:5)
rownames(rd) <- paste0("m",1:5)
pheatmap(rd)
```



```
rf <- rmsf(xyz)

plotb3(rf, sse=pdb)
abline(v=100, col="gray", ylab="RMSF")
```



Predicted Alignment Error for domains

```
library(jsonlite)

pae_files <- list.files(path=results_dir,
                        pattern=".*model.*\\.json",
                        full.names = TRUE)
```

1st and 5th files for example

```
pae1 <- read_json(pae_files[1],simplifyVector = TRUE)
pae5 <- read_json(pae_files[5],simplifyVector = TRUE)

attributes(pae1)
```

```
$names
[1] "plddt"   "max_pae" "pae"     "ptm"     "iptm"
```

```
# Per-residue pLDDT scores
#  same as B-factor of PDB..
```

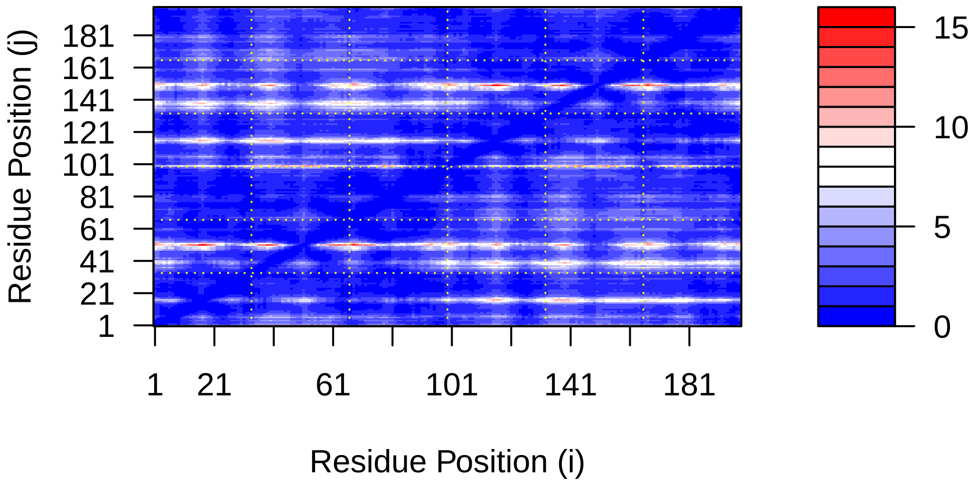
```
head(pae1$plddt)
```

```
[1] 92.50 96.56 96.94 96.62 97.69 96.00
```

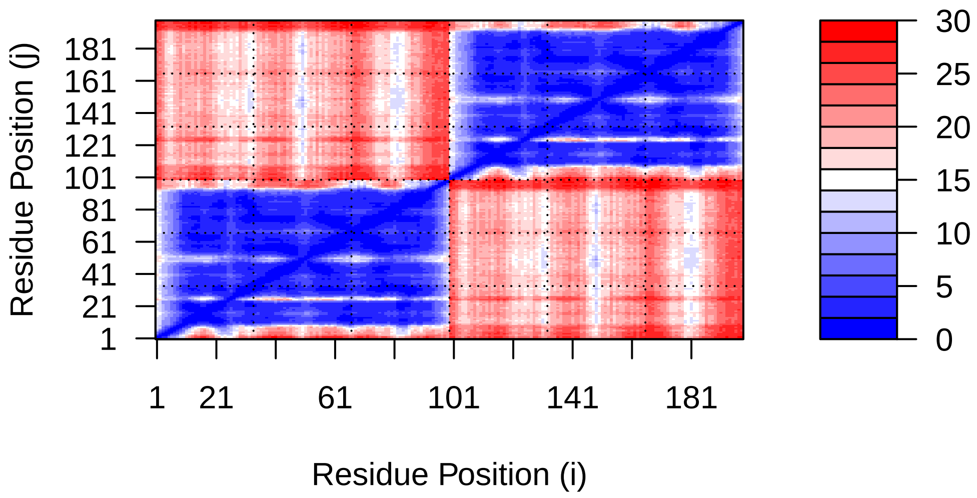
```
pae1$max_pae
```

```
[1] 15.54688
```

```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```

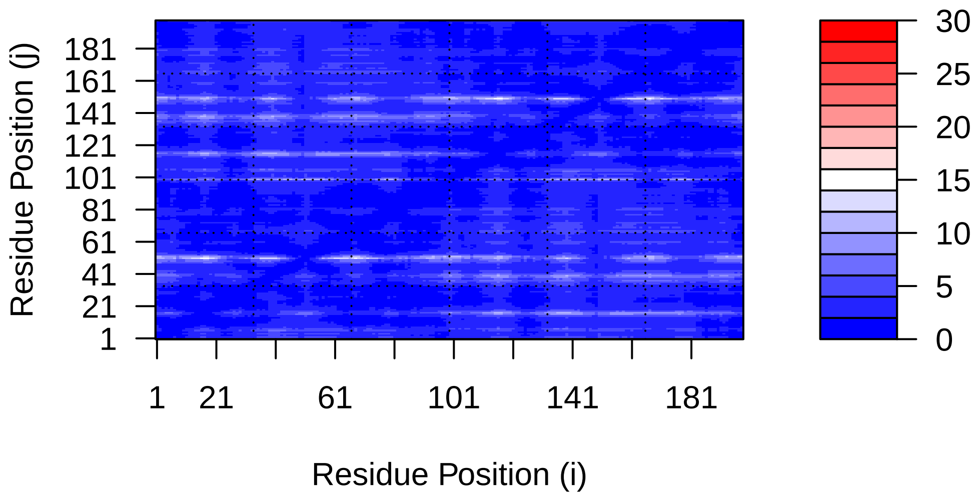


```
plot.dmat(pae5$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black",
          zlim=c(0,30))
```



Model 1 plot with the same Z-range

```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black",
          zlim=c(0,30))
```



Residue conservation from alignment file

```
aln_file <- list.files(path=results_dir,
                       pattern=".a3m$",
                        full.names = TRUE)
aln_file
```

```
[1] "hivpr_dimer_23119/hivpr_dimer_23119.a3m"
```

```
aln <- read.fasta(aln_file[1], to.upper = TRUE)
```

```
[1] " ** Duplicated sequence id's: 101 **"
[2] " ** Duplicated sequence id's: 101 **"
```

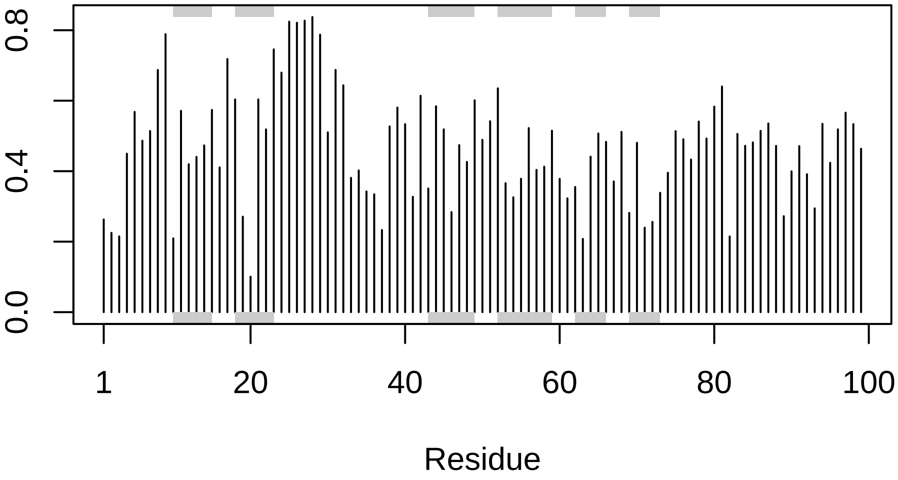
Amount of sequences in this alignment

```
dim(aln$ali)
```

```
[1] 5378  132
```

Score of residue conservation

```
sim <- conserv(aln)
plotb3(sim[1:99], sse=trim.pdb(pdb, chain="A"))
```



Conserved Active Site residue D25, T26, G27, A28

```
con <- consensus(aln, cutoff = 0.9)
con$seq
```

```
  [1] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [19] "-" "-" "-" "-" "-" "-" "D" "T" "G" "A" "-" "-" "-" "-" "-" "-" "-" "-"
 [37] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [55] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [73] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
 [91] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[109] "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-" "-"
[127] "-" "-" "-" "-" "-" "-"
```

Mapping the conservation score to the Occupancy column of a PDB file

```
m1.pdb <- read.pdb(pdb_files[1])
occ <- vec2resno(c(sim[1:99], sim[1:99]), m1.pdb$atom$resno)
write.pdb(m1.pdb, o=occ, file="m1_conserv.pdb")
```