class10

AUTHOR
Jaewon Kim

The PDB database

Here we examine the size and composition of the main database of biomolecular structures - the PDB.

Get a CSV file from the PDB database and read it into R.

```
stat_summary <- read.csv("pdb stat summary.csv", row.names = 1)
stat_summary
```

	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

My pdbstats data frame has numbers with commas in them. This may cause us problems. Let's see

```
stat_summary$X.ray #chr with numbers
```

```
[1] "161,663" "9,348"   "8,404"   "2,758"   "164"     "11"
```

I can turn this snipet into a function that I can use for every column in the table

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

commasum(stat_summary$X.ray)
```

```
[1] 182348
```

Apply accross all columns

```
totals <- apply(stat_summary, 2, commasum)
totals
```

```
          X.ray             EM            NMR Multiple.methods
         182348          18817          14173            230
        Neutron          Other          Total
             79             37         215684
```

```
(totals/totals["Total"])*100
```

```
          X.ray             EM            NMR Multiple.methods
    84.54405519     8.72433746     6.57118748     0.10663749
        Neutron          Other          Total
     0.03662766     0.01715473   100.00000000
```

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

From the code above, 84.544% of structures are solved by X-ray and 8.724% by EM.

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

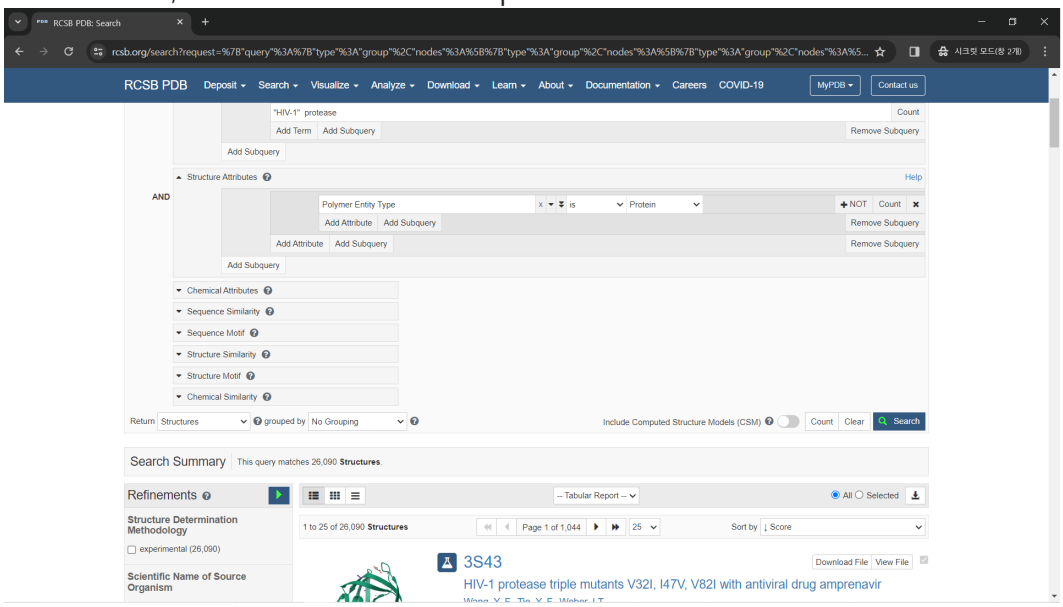
```
total.row <- apply(stat_summary, 1, commasum)
(total.row/sum(total.row))*100
```

```
      Protein (only) Protein/Oligosaccharide              Protein/NA
         86.65362289             5.35922924             5.85161625
Nucleic acid (only)                  Other Oligosaccharide (only)
          2.02982141             0.09551010             0.01020011
```

Therefore, 86.654% of structures in PDB are protein.

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

There are 26,090 structures of HIV-1 protease in current PDB.



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

Oxygen atoms is larger than hydrogen atoms that size of hydrogen atoms are insignificant compared to oxygen and rest of protein structure (Also prof. said hydrogen are too small to be

visible in size during lecture?). Therefore, only oxygen molecule is displayed.

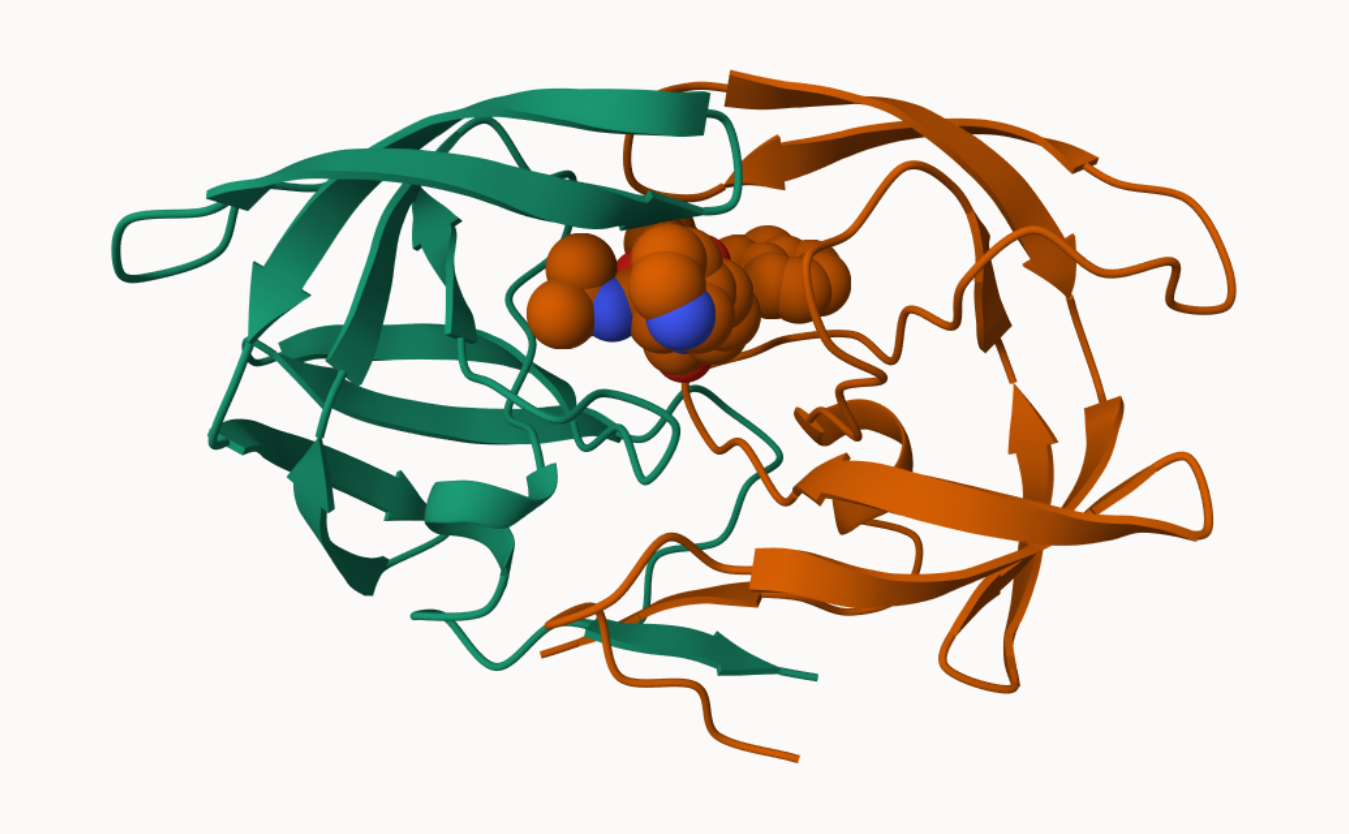
Q5: There is a critical "conserved" water molecule in the binding site. Can you identify this water molecule? What residue number does this water molecule have

HOH308 is forming bond with both ligand and I50 in protein.

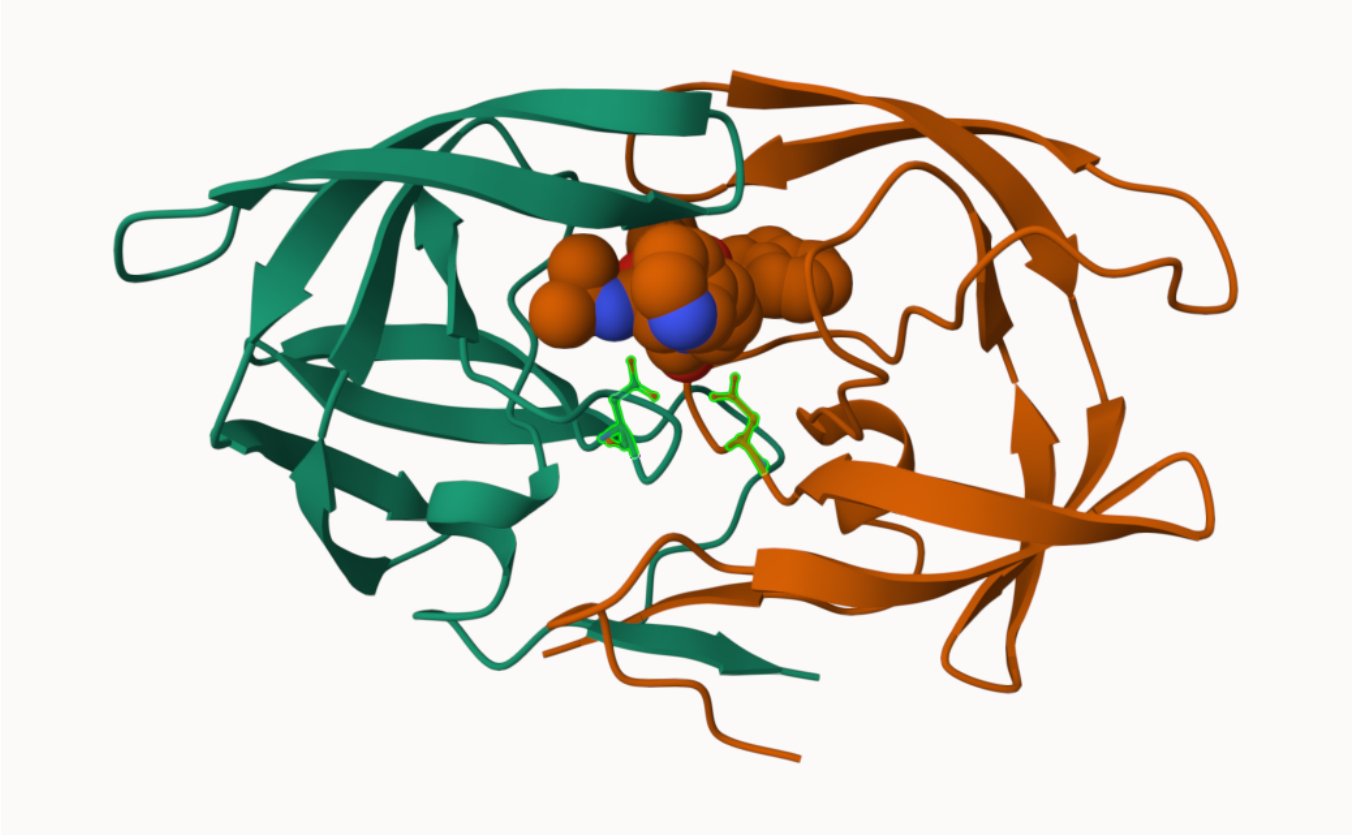
Visualizing Protein Structure

We will learn the basics of Mol* (mol-star) homepage: https://molstar.org/viewer/

We will play with PDB code 1HSG



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.



HIV-Pr with a bound inhibitor showing the two important ASP 25

Discussion Topic: Can you think of a way in which indinavir, or even larger ligands and substrates, could enter the binding site?

If question is asking "How do ligans enter binding site" : Two chains rotates (like twist) that binding site opens up, making larger space for ligand to enter. As ligand form bond with protein, two chains untwist and closes its binding site.

If question is asking "is there way to modify protein so that larger ligand can bind" : Maybe modify protein usign bump and hole method so that larger ligands have a space to bind.

Back to R and working with PDB structure

Predict the dynmics (flexibility) of an important protein:

```
library(bio3d)

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

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

```
hiv
```

```

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 amino acid residues are there in this pdb object?

There are 198 aa residues.

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

There are 128 non-protein residues, where their types are HOH (water) and MK1 (inhibitor)

Q9: How many protein chains are in this structure?

There are two chains, chain A and B.

```
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>
```

```
pdbseq(hiv)
```

```
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
"P" "Q" "I" "T" "L" "W" "Q" "R" "P" "L" "V" "T" "I" "K" "I" "G" "G" "Q" "L" "K"
 21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40
"E" "A" "L" "L" "D" "T" "G" "A" "D" "D" "T" "V" "L" "E" "E" "M" "S" "L" "P" "G"
 41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60
"R" "W" "K" "P" "K" "M" "I" "G" "G" "I" "G" "G" "F" "I" "K" "V" "R" "Q" "Y" "D"
 61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80
```

```
"Q" "I" "L" "I" "E" "I" "C" "G" "H" "K" "A" "I" "G" "T" "V" "L" "V" "G" "P" "T"
 81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99   1
"P" "V" "N" "I" "I" "G" "R" "N" "L" "L" "T" "Q" "I" "G" "C" "T" "L" "N" "F" "P"
  2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21
"Q" "I" "T" "L" "W" "Q" "R" "P" "L" "V" "T" "I" "K" "I" "G" "G" "Q" "L" "K" "E"
 22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41
"A" "L" "L" "D" "T" "G" "A" "D" "D" "T" "V" "L" "E" "E" "M" "S" "L" "P" "G" "R"
 42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61
"W" "K" "P" "K" "M" "I" "G" "G" "I" "G" "G" "F" "I" "K" "V" "R" "Q" "Y" "D" "Q"
 62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81
"I" "L" "I" "E" "I" "C" "G" "H" "K" "A" "I" "G" "T" "V" "L" "V" "G" "P" "T" "P"
 82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99
"V" "N" "I" "I" "G" "R" "N" "L" "L" "T" "Q" "I" "G" "C" "T" "L" "N" "F"
```

Here we will do a normal mode analysis (nma) to predict functional motion of a kinase

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

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

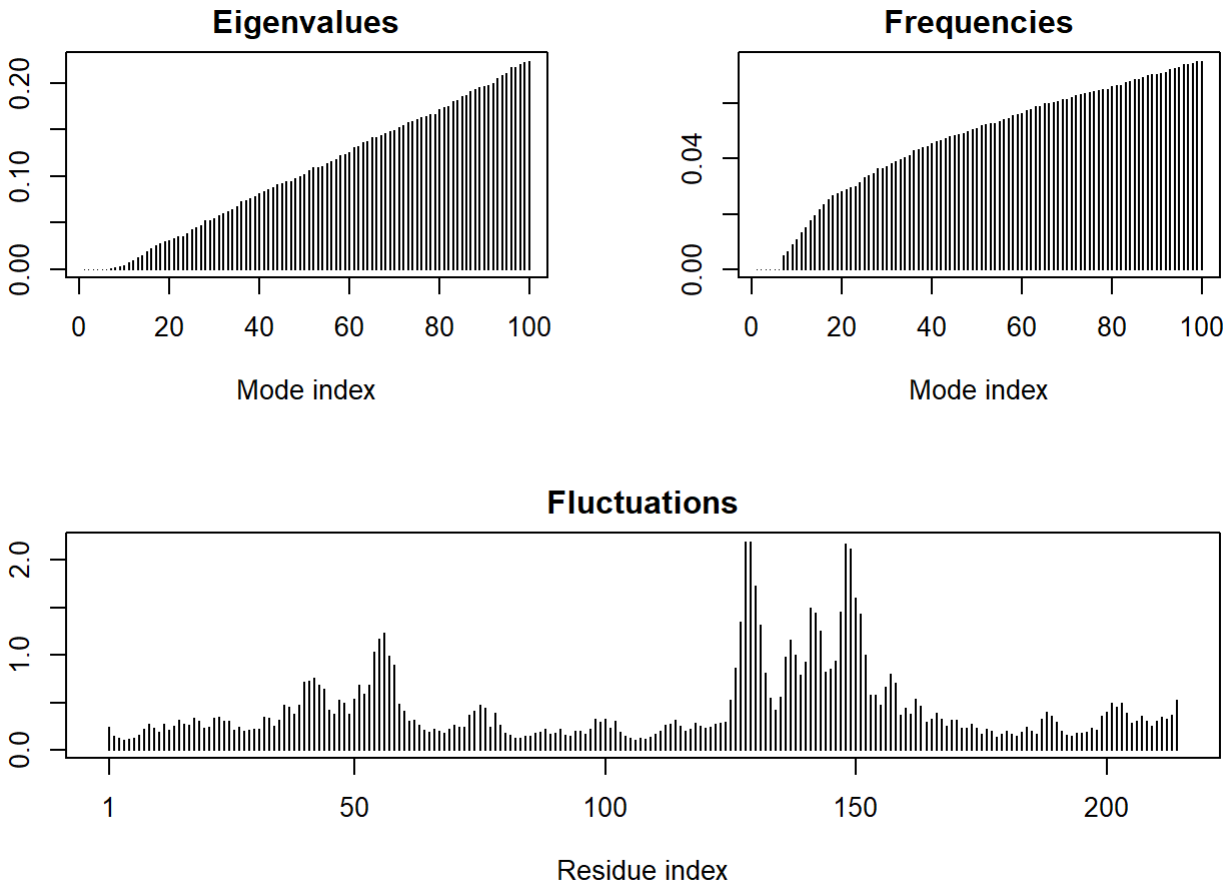
```
adk
```

```

 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
```

```
modes <- nma(adk)
```

```
 Building Hessian...        Done in 0.01 seconds.
 Diagonalizing Hessian...   Done in 0.23 seconds.
```

```
plot(modes)
```



Make a "movie" called a trajectory of the predicted motions:

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

Then I can open this file in Mol*...

```
modes1 <- nma(read.pdb("1hsg"))
```

```
  Note: Accessing on-line PDB file

Warning in get.pdb(file, path = tempdir(), verbose = FALSE):
C:\Users\louis\AppData\Local\Temp\RtmpEtEyMb/1hsg.pdb exists. Skipping download

Warning in nma.pdb(read.pdb("1hsg")): Possible multi-chain structure or missing in-structure
residue(s) present
  Fluctuations at neighboring positions may be affected.

 Building Hessian...        Done in 0.02 seconds.
 Diagonalizing Hessian...   Done in 0.18 seconds.
```

```
mktrj(modes1, file = "1hsg_m7.pdb")
```

install.packages("bio3d") install.packages("devtools") install.packages("BiocManager") BiocManager::install("msa") devtools::install_bitbucket("Grantlab/bio3d-view")

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

msa package is found only on BioConductor.

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

bio3d-view (Grantlab/bio3d-view) is found only on bitbucket.

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

True.

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

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

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

```
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 are in this sequence, i.e. how long is this sequence?

There are 214 aa in this sequence.

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

```
#head(hits$pdb.id)
#Takes to long

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

```
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 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%
  |
```

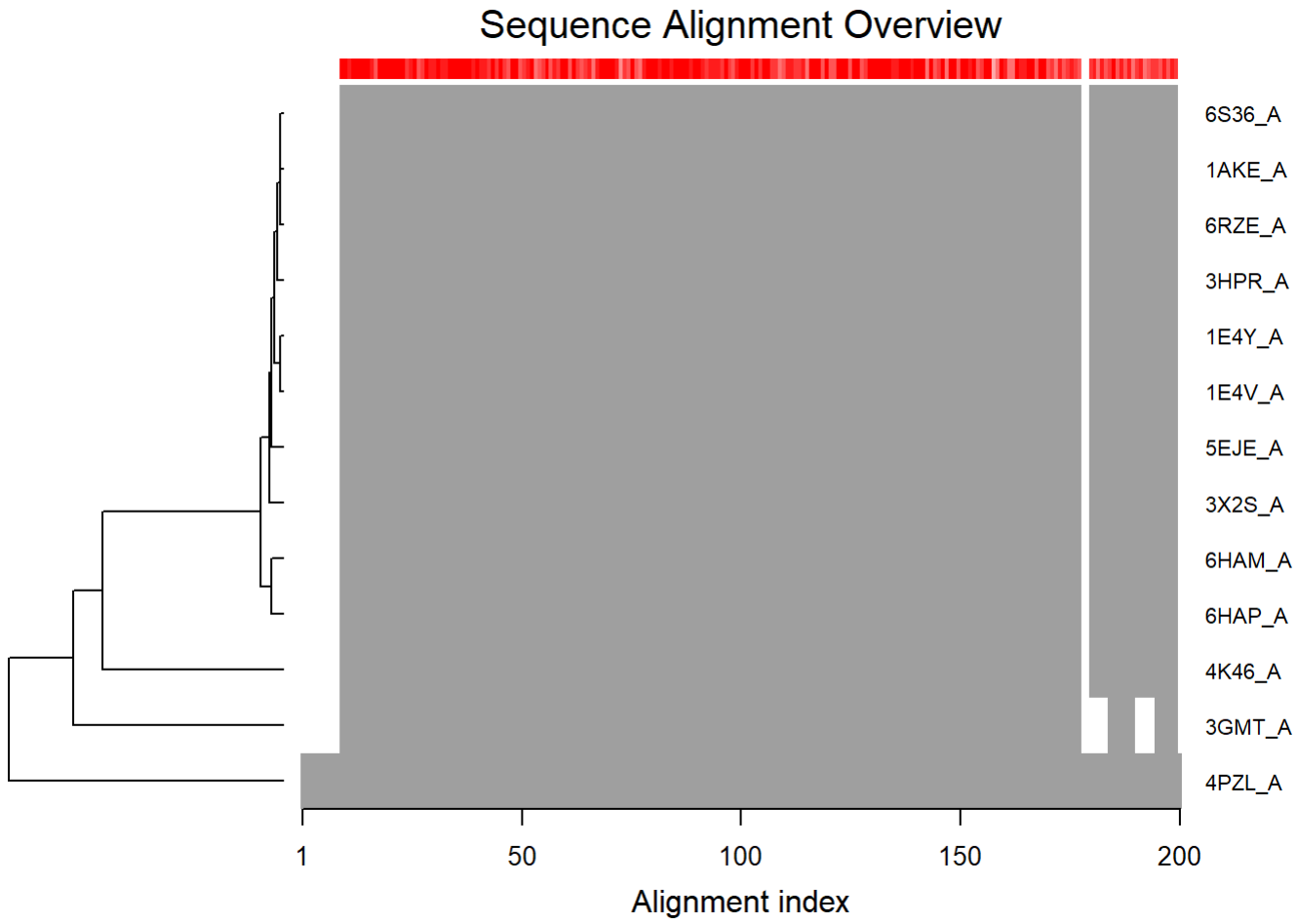
```
  |===========                                                     |  15%
  |
  |================                                                |  23%
  |
  |======================                                          |  31%
  |
  |============================                                    |  38%
  |
  |================================                                |  46%
  |
  |====================================                            |  54%
  |
  |========================================                        |  62%
  |
  |============================================                    |  69%
  |
  |================================================                |  77%
  |
  |========================================================        |  85%
  |
  |============================================================    |  92%
  |
  |================================================================| 100%
```

```
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
```

```
ids <- basename.pdb(pdbs$id)
plot(pdbs, labels=ids)
```




```
#Why is this not working? Please commentttt
library(bio3d.view)
library(rgl)
```

```
view.pdbs(pdbs)
```

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

```
[1] "Escherichia coli"
[2] "Escherichia coli K-12"
[3] "Escherichia coli O139:H28 str. E24377A"
[4] "Escherichia coli str. K-12 substr. MDS42"
[5] "Photobacterium profundum"
[6] "Burkholderia pseudomallei 1710b"
[7] "Francisella tularensis subsp. tularensis SCHU S4"
```

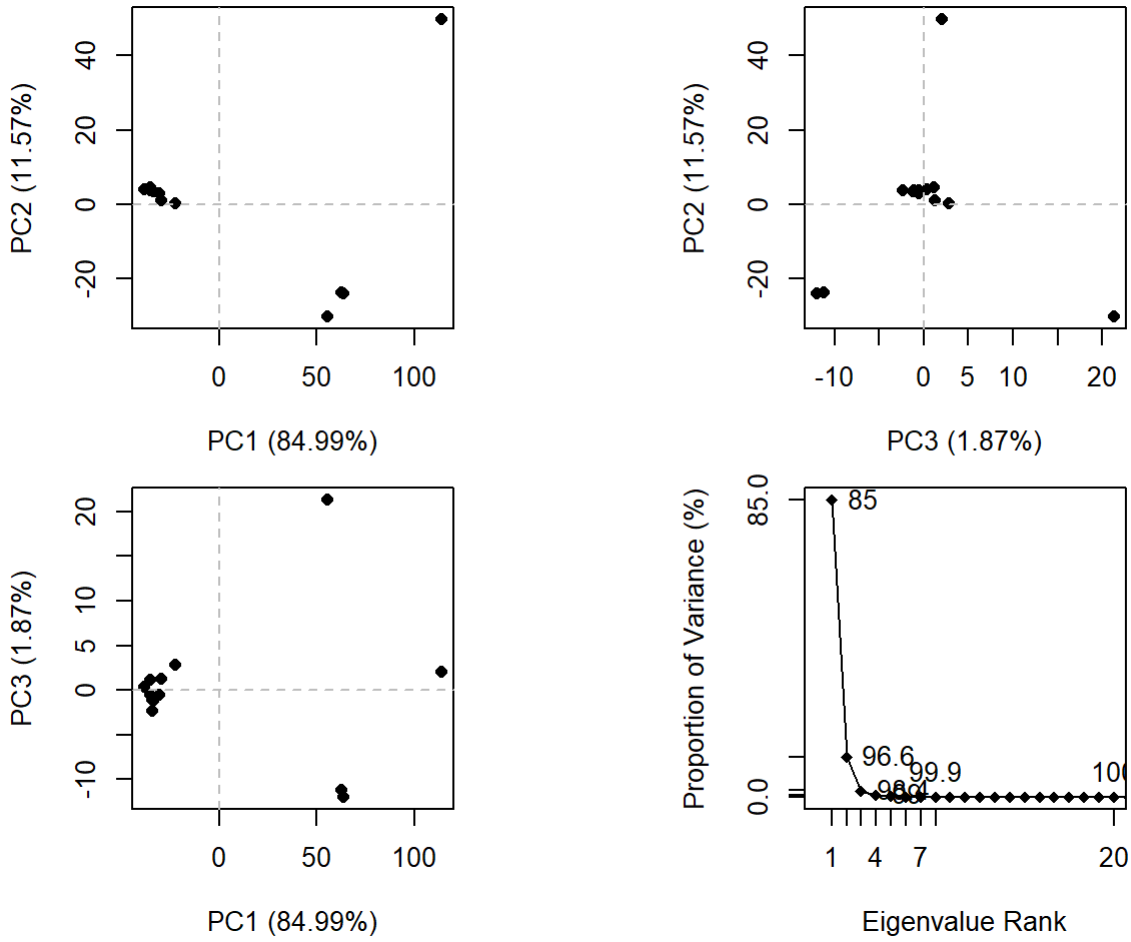
```
anno
```

```
       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
1E4Y_A        1E4Y       A           Protein         214                X-ray
3X2S_A        3X2S       A           Protein         214                X-ray
6HAP_A        6HAP       A           Protein         214                X-ray
6HAM_A        6HAM       A           Protein         214                X-ray
4K46_A        4K46       A           Protein         214                X-ray
3GMT_A        3GMT       A           Protein         230                X-ray
4PZL_A        4PZL       A           Protein         242                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 (ADK)
6RZE_A       1.69             <NA>                     Adenylate kinase (ADK)
3HPR_A       2.00             <NA> Adenylate kinase, active site lid (ADK_lid)
1E4V_A       1.85 Adenylate kinase                     Adenylate kinase (ADK)
5EJE_A       1.90             <NA>                     Adenylate kinase (ADK)
1E4Y_A       1.85 Adenylate kinase                     Adenylate kinase (ADK)
3X2S_A       2.80             <NA>                     Adenylate kinase (ADK)
6HAP_A       2.70             <NA> Adenylate kinase, active site lid (ADK_lid)
6HAM_A       2.55             <NA> Adenylate kinase, active site lid (ADK_lid)
4K46_A       2.01             <NA> Adenylate kinase, active site lid (ADK_lid)
3GMT_A       2.10             <NA>                     Adenylate kinase (ADK)
4PZL_A       2.10             <NA>                     Adenylate kinase (ADK)
              ligandId
1AKE_A             AP5
6S36_A NA,MG (2),CL (3)
6RZE_A    CL (2),NA (3)
3HPR_A             AP5
1E4V_A             AP5
5EJE_A          AP5,CO
1E4Y_A             AP5
3X2S_A   JPY (2),AP5,MG
```

```
6HAP_A              AP5
6HAM_A              AP5
4K46_A      ADP,AMP,PO4
3GMT_A          SO4 (2)
4PZL_A       CA,GOL,FMT
                                                                                ligandName
1AKE_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
6S36_A                                  SODIUM ION,MAGNESIUM ION (2),CHLORIDE ION (3)
6RZE_A                                                   CHLORIDE ION (2),SODIUM ION (3)
3HPR_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
1E4V_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
5EJE_A                                BIS(ADENOSINE)-5'-PENTAPHOSPHATE,COBALT (II) ION
1E4Y_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
3X2S_A N-(pyren-1-ylmethyl)acetamide (2),BIS(ADENOSINE)-5'-PENTAPHOSPHATE,MAGNESIUM ION
6HAP_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
6HAM_A                                                  BIS(ADENOSINE)-5'-PENTAPHOSPHATE
4K46_A                   ADENOSINE-5'-DIPHOSPHATE,ADENOSINE MONOPHOSPHATE,PHOSPHATE ION
3GMT_A                                                                  SULFATE ION (2)
4PZL_A                                                  CALCIUM ION,GLYCEROL,FORMIC ACID
                                                 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
1E4Y_A                                 Escherichia coli
3X2S_A         Escherichia coli str. K-12 substr. MDS42
6HAP_A           Escherichia coli O139:H28 str. E24377A
6HAM_A                             Escherichia coli K-12
4K46_A                          Photobacterium profundum
3GMT_A                  Burkholderia pseudomallei 1710b
4PZL_A Francisella tularensis subsp. tularensis SCHU S4

structureTitle
1AKE_A STRUCTURE OF THE COMPLEX BETWEEN ADENYLATE KINASE FROM ESCHERICHIA COLI AND THE
INHIBITOR AP5A REFINED AT 1.9 ANGSTROMS RESOLUTION: A MODEL FOR A CATALYTIC TRANSITION STATE
6S36_A
Crystal structure of E. coli Adenylate kinase R119K mutant
6RZE_A
Crystal structure of E. coli Adenylate kinase R119A mutant
3HPR_A
Crystal structure of V148G adenylate kinase from E. coli, in complex with Ap5A
1E4V_A
Mutant G10V of adenylate kinase from E. coli, modified in the Gly-loop
5EJE_A
Crystal structure of E. coli Adenylate kinase G56C/T163C double mutant in complex with Ap5a
1E4Y_A
Mutant P9L of adenylate kinase from E. coli, modified in the Gly-loop
3X2S_A
Crystal structure of pyrene-conjugated adenylate kinase
6HAP_A
Adenylate kinase
6HAM_A
```

```
Adenylate kinase
4K46_A
Crystal Structure of Adenylate Kinase from Photobacterium profundum
3GMT_A
Crystal structure of adenylate kinase from burkholderia pseudomallei
4PZL_A                                                                    The
crystal structure of adenylate kinase from Francisella tularensis subsp. tularensis SCHU S4
                                                  citation rObserved   rFree
1AKE_A                Muller, C.W., et al. J Mol Biol (1992)   0.19600      NA
6S36_A                 Rogne, P., et al. Biochemistry (2019)   0.16320 0.23560
6RZE_A                 Rogne, P., et al. Biochemistry (2019)   0.18650 0.23500
3HPR_A  Schrank, T.P., et al. Proc Natl Acad Sci U S A (2009)   0.21000 0.24320
1E4V_A                  Muller, C.W., et al. Proteins (1993)   0.19600      NA
5EJE_A  Kovermann, M., et al. Proc Natl Acad Sci U S A (2017)   0.18890 0.23580
1E4Y_A                  Muller, C.W., et al. Proteins (1993)   0.17800      NA
3X2S_A               Fujii, A., et al. Bioconjug Chem (2015)   0.20700 0.25600
6HAP_A              Kantaev, R., et al. J Phys Chem B (2018)   0.22630 0.27760
6HAM_A              Kantaev, R., et al. J Phys Chem B (2018)   0.20511 0.24325
4K46_A                   Cho, Y.-J., et al. To be published    0.17000 0.22290
3GMT_A Buchko, G.W., et al. Biochem Biophys Res Commun (2010)   0.23800 0.29500
4PZL_A                       Tan, K., et al. To be published    0.19360 0.23680
         rWork spaceGroup
1AKE_A 0.19600  P 21 2 21
6S36_A 0.15940    C 1 2 1
6RZE_A 0.18190    C 1 2 1
3HPR_A 0.20620  P 21 21 2
1E4V_A 0.19600  P 21 2 21
5EJE_A 0.18630  P 21 2 21
1E4Y_A 0.17800   P 1 21 1
3X2S_A 0.20700 P 21 21 21
6HAP_A 0.22370    I 2 2 2
6HAM_A 0.20311       P 43
4K46_A 0.16730 P 21 21 21
3GMT_A 0.23500   P 1 21 1
4PZL_A 0.19130       P 32
```

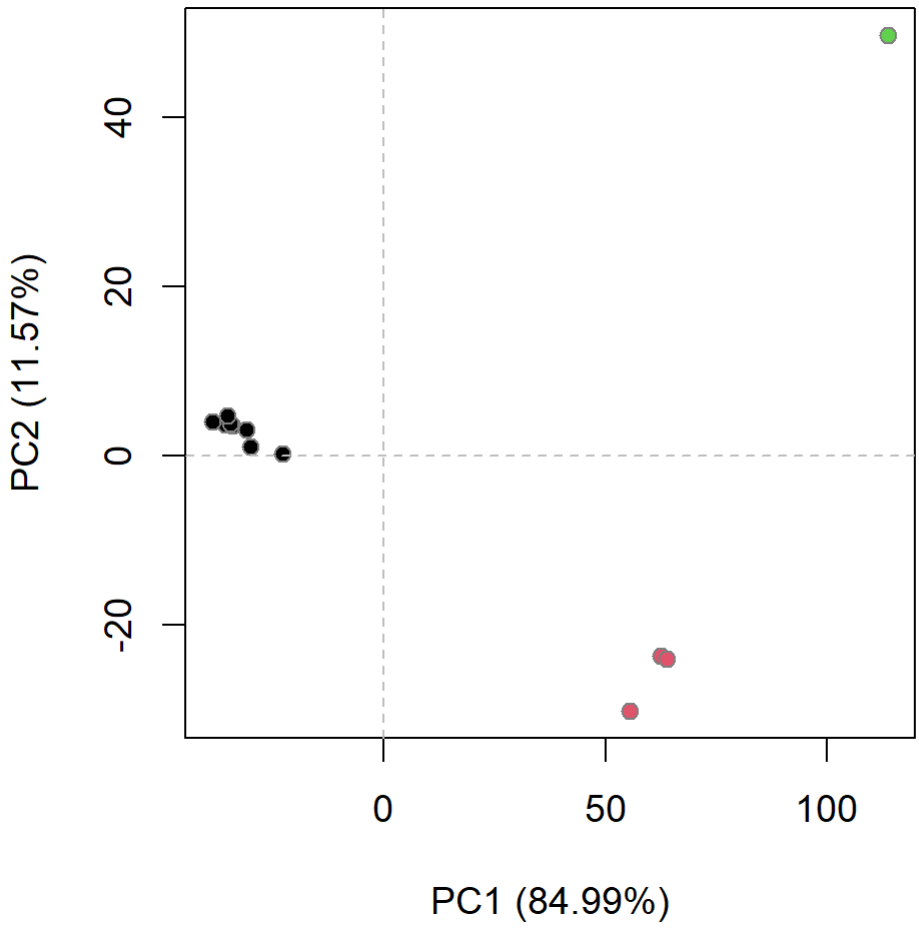
```
pc.xray <- pca(pdbs)
plot(pc.xray)
```




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

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

```
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)
```




```
library(ggplot2)
library(ggrepel)

df <- data.frame(PC1=pc.xray$z[,1],
                 PC2=pc.xray$z[,2],
                 col=as.factor(grps.rd),
                 ids=ids)

p <- ggplot(df) +
  aes(PC1, PC2, col=col, label=ids) +
  geom_point(size=2) +
  geom_text_repel(max.overlaps = 20) +
  theme(legend.position = "none")
p
```

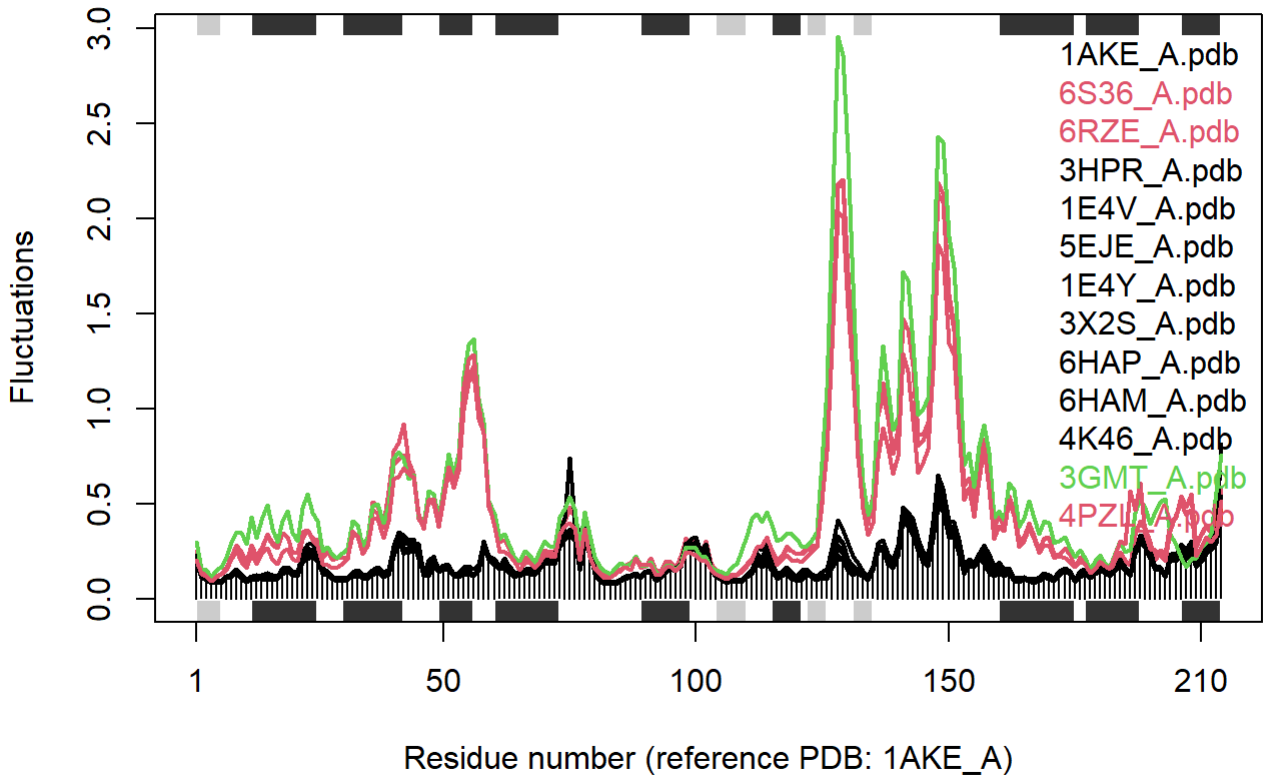

```
modes <- nma(pdbs)
```

```
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%
```

```
plot(modes, pdbs, col=grps.rd)
```

```
Extracting SSE from pdbs$sse attribute
```



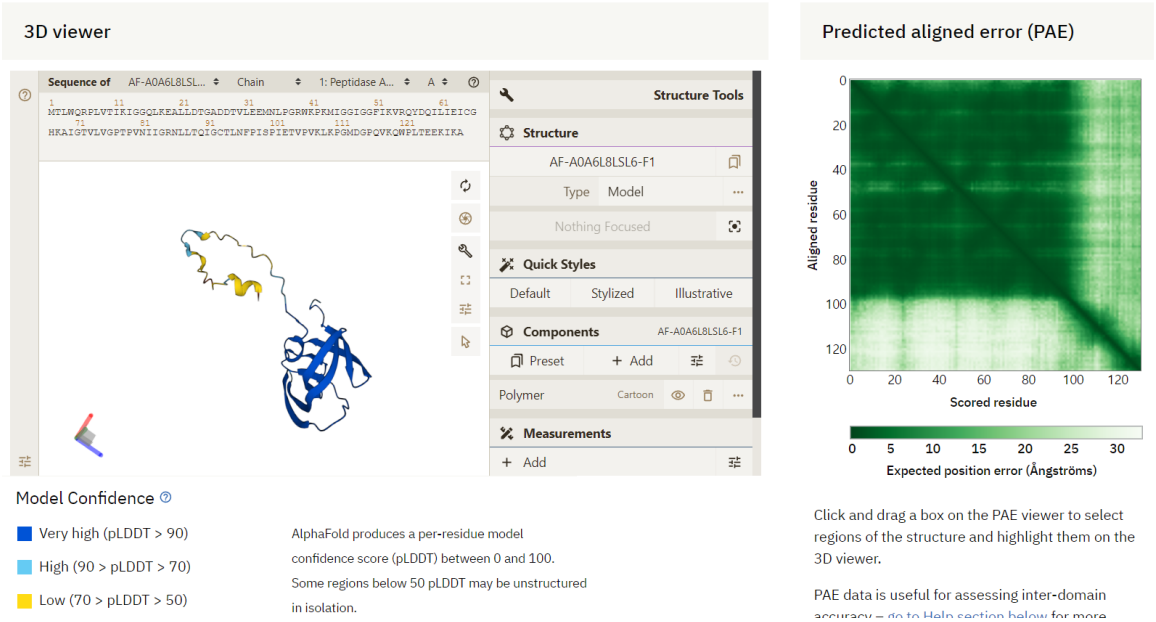
Q14. What do you note about this plot? Are the black and colored lines similar or different? Where do you think they differ most and why?

Colored line and black lines are different in flucuations around residue 1~50 and 120~170. Fluctuation differs the most around residue 130. This is because corresponding area is where protein moves (open and close activation site) to enable substrate binding.

class11

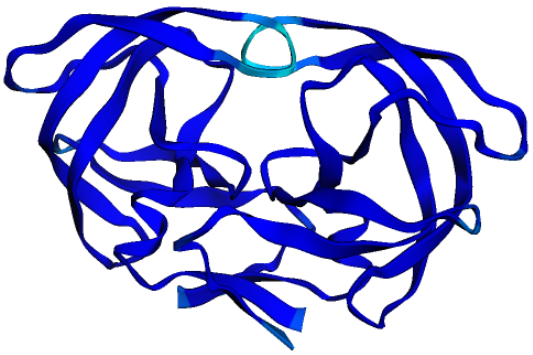
Jaewon Kim

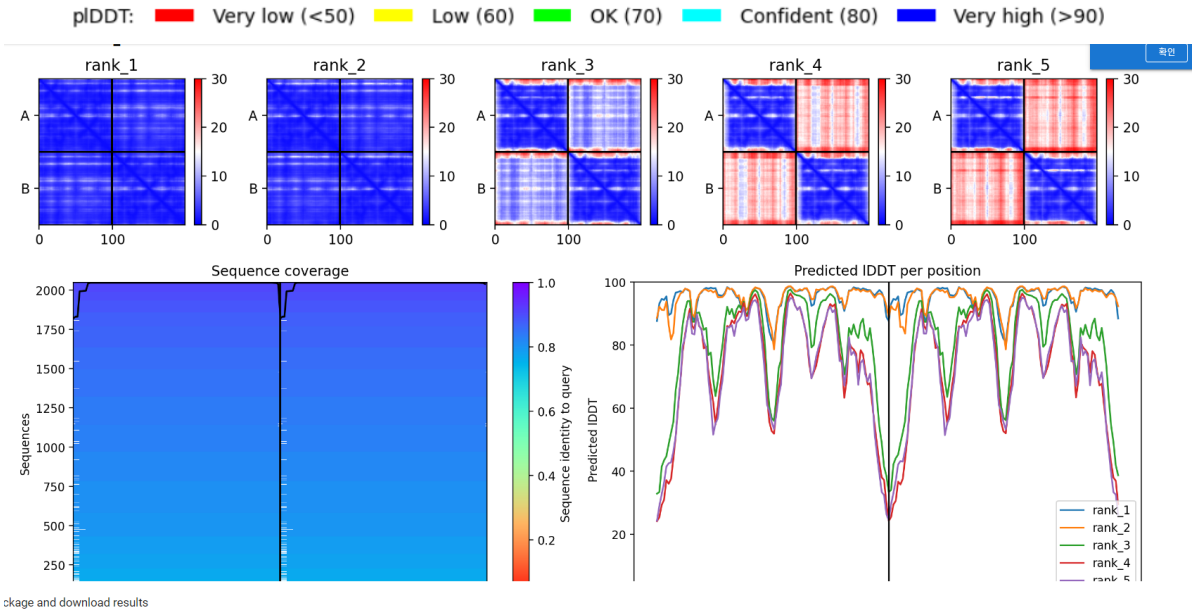
Alphafold has changed the game for protein structure prediction and allows anyone with sufficient bioinformatics skills to predict the structure of virtually any protein.



we ran alphafold via googlecolab at: https://github.com/sokrypton/ColabFold

In particular we used their AlphaFold2_mmseqs2 version that uses mmseqs2 rather than HMMMer for sequence search.





The main outputs include a set of **PDB structure files** along with matching **JSON format files** that tell us how good the resulting models might be.

Let's start my loading these structures up in Mol*

```
library(bio3d)

# Change this for YOUR results dir name
```

```
results_dir <- "C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.resu

# File names for all PDB models
pdb.files <- list.files(path= results_dir, pattern="*.pdb", full.names = TRUE)

# Print our PDB file names
basename(pdb.files)
```

```
[1] "HIV1prhomodimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb"
[2] "HIV1prhomodimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb"
[3] "HIV1prhomodimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb"
[4] "HIV1prhomodimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb"
[5] "HIV1prhomodimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb"
```

```
# Read all data from Models
#  and superpose/fit coords
pdbs <- pdbaln(pdb.files, fit=TRUE, exefile="msa")
```

```
Reading PDB files:
C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.result/HIV1prhomodimer
C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.result/HIV1prhomodimer
C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.result/HIV1prhomodimer
C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.result/HIV1prhomodimer
C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.result/HIV1prhomodimer
.....

Extracting sequences

pdb/seq: 1   name: C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.resu
pdb/seq: 2   name: C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.resu
pdb/seq: 3   name: C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.resu
pdb/seq: 4   name: C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.resu
pdb/seq: 5   name: C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.resu
```

```
pdbs
```

```
                                1        .        .        .        .         50
[Truncated_Name:1]HIV1prhomo   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:2]HIV1prhomo   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
```

```
[Truncated_Name:3]HIV1prhomo   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:4]HIV1prhomo   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:5]HIV1prhomo   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
                               **************************************************
                               1        .         .         .         .         50

                              51        .         .         .         .         100
[Truncated_Name:1]HIV1prhomo   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:2]HIV1prhomo   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:3]HIV1prhomo   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:4]HIV1prhomo   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:5]HIV1prhomo   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
                               **************************************************
                              51        .         .         .         .         100

                             101        .         .         .         .         150
[Truncated_Name:1]HIV1prhomo   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:2]HIV1prhomo   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:3]HIV1prhomo   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:4]HIV1prhomo   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:5]HIV1prhomo   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
                               **************************************************
                             101        .         .         .         .         150

                             151        .         .         .         .       198
[Truncated_Name:1]HIV1prhomo   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:2]HIV1prhomo   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:3]HIV1prhomo   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:4]HIV1prhomo   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:5]HIV1prhomo   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
```

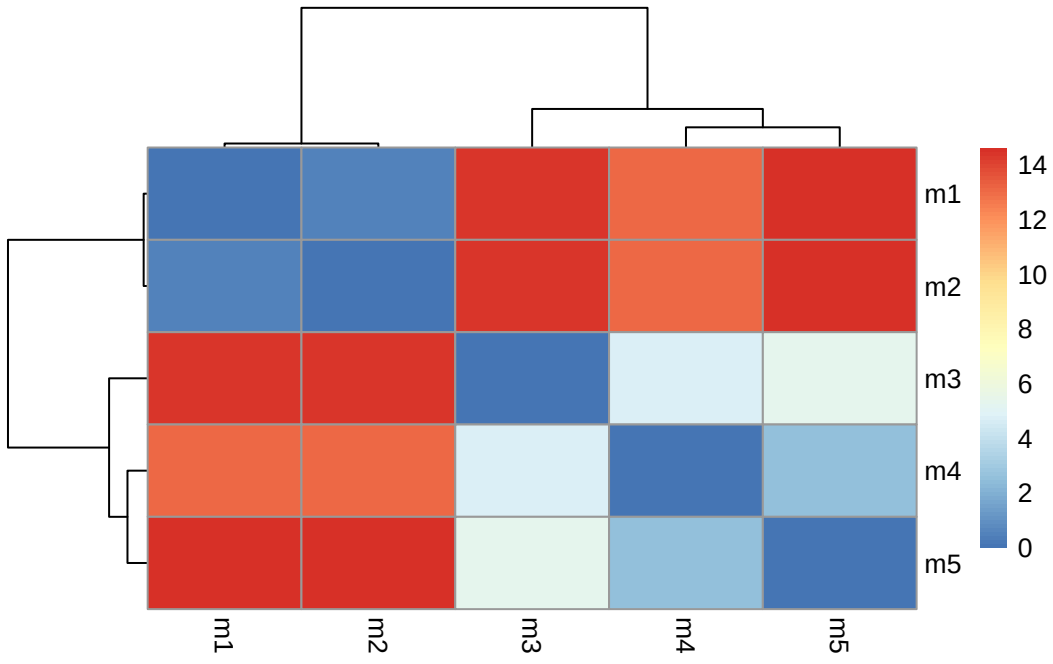
```
#install.packages("pheatmap")
library(pheatmap)

rd <- rmsd(pdbs, fit=T)
```

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

```
range(rd)
```

```
[1]  0.000 14.631
```

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



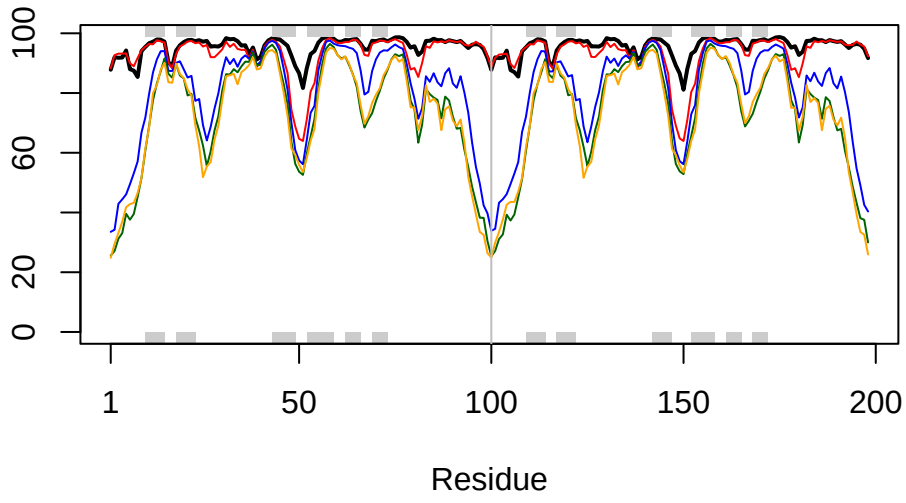

```
rd
```

```
       m1     m2     m3     m4     m5
m1  0.000  0.572 14.407 13.153 14.631
m2  0.572  0.000 14.423 13.060 14.548
m3 14.407 14.423  0.000  4.821  5.335
m4 13.153 13.060  4.821  0.000  2.496
m5 14.631 14.548  5.335  2.496  0.000
```

```
#Read ref. PDB

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

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

```
plotb3(pdbs$b[1,], typ="l", lwd=2, sse=pdb)
points(pdbs$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 = 4578.346
core size 196 of 198  vol = 3931.108
core size 195 of 198  vol = 3709.733
core size 194 of 198  vol = 3496.019
core size 193 of 198  vol = 3302.432
core size 192 of 198  vol = 3146.474
core size 191 of 198  vol = 3048.964
core size 190 of 198  vol = 2970.354
core size 189 of 198  vol = 2893.012
core size 188 of 198  vol = 2831.825
core size 187 of 198  vol = 2774.506
core size 186 of 198  vol = 2728.043
core size 185 of 198  vol = 2704.946
core size 184 of 198  vol = 2701.981
core size 183 of 198  vol = 2715.909
core size 182 of 198  vol = 2809.853
core size 181 of 198  vol = 2888.95
core size 180 of 198  vol = 2967.282
core size 179 of 198  vol = 3036.256
core size 178 of 198  vol = 3066.287
core size 177 of 198  vol = 3096.833
core size 176 of 198  vol = 3056.414
core size 175 of 198  vol = 3014.768
core size 174 of 198  vol = 2975.013
core size 173 of 198  vol = 2898.051
core size 172 of 198  vol = 2810.173
core size 171 of 198  vol = 2747.532
core size 170 of 198  vol = 2684.434
core size 169 of 198  vol = 2620.353
core size 168 of 198  vol = 2550.877
core size 167 of 198  vol = 2492.582
core size 166 of 198  vol = 2422.978
core size 165 of 198  vol = 2358.916
core size 164 of 198  vol = 2298.292
core size 163 of 198  vol = 2235.918
core size 162 of 198  vol = 2171.02
core size 161 of 198  vol = 2093.559
core size 160 of 198  vol = 2029.144
core size 159 of 198  vol = 1950.957
core size 158 of 198  vol = 1881.015
```

```
core size 157 of 198  vol = 1801.506
core size 156 of 198  vol = 1728.892
core size 155 of 198  vol = 1660.037
core size 154 of 198  vol = 1586.149
core size 153 of 198  vol = 1532.718
core size 152 of 198  vol = 1460.186
core size 151 of 198  vol = 1399.251
core size 150 of 198  vol = 1333.908
core size 149 of 198  vol = 1271.747
core size 148 of 198  vol = 1219.496
core size 147 of 198  vol = 1176.003
core size 146 of 198  vol = 1138.478
core size 145 of 198  vol = 1102.124
core size 144 of 198  vol = 1049.642
core size 143 of 198  vol = 1014.063
core size 142 of 198  vol = 970.575
core size 141 of 198  vol = 929.178
core size 140 of 198  vol = 889.104
core size 139 of 198  vol = 846.668
core size 138 of 198  vol = 805.8
core size 137 of 198  vol = 775.034
core size 136 of 198  vol = 743.09
core size 135 of 198  vol = 715.695
core size 134 of 198  vol = 689.788
core size 133 of 198  vol = 660.329
core size 132 of 198  vol = 630.966
core size 131 of 198  vol = 597.207
core size 130 of 198  vol = 566.989
core size 129 of 198  vol = 532.89
core size 128 of 198  vol = 496.208
core size 127 of 198  vol = 463.183
core size 126 of 198  vol = 431.893
core size 125 of 198  vol = 408.864
core size 124 of 198  vol = 376.61
core size 123 of 198  vol = 362.377
core size 122 of 198  vol = 353.633
core size 121 of 198  vol = 331.501
core size 120 of 198  vol = 312.518
core size 119 of 198  vol = 286.715
core size 118 of 198  vol = 262.336
core size 117 of 198  vol = 245.109
core size 116 of 198  vol = 228.342
core size 115 of 198  vol = 210.366
```

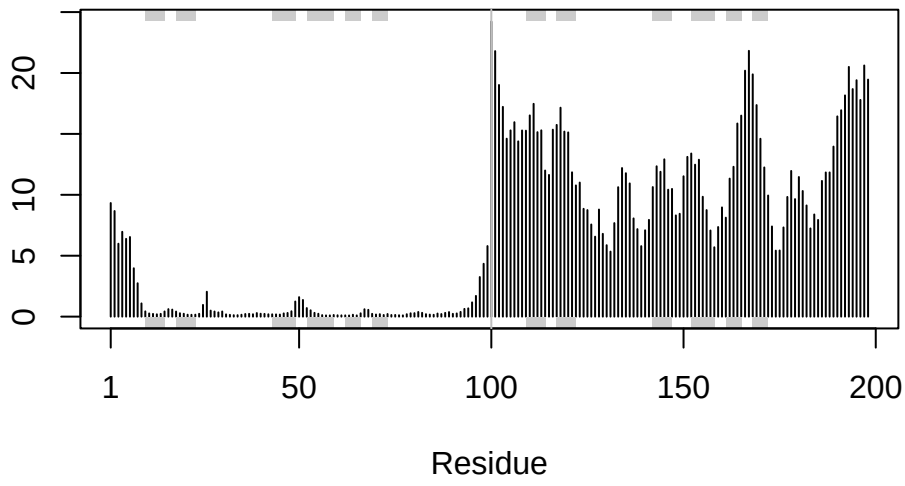
```
 core size 114 of 198  vol = 197.519
 core size 113 of 198  vol = 179.392
 core size 112 of 198  vol = 161.891
 core size 111 of 198  vol = 148.359
 core size 110 of 198  vol = 134.477
 core size 109 of 198  vol = 121.261
 core size 108 of 198  vol = 109.516
 core size 107 of 198  vol = 103.031
 core size 106 of 198  vol = 96.443
 core size 105 of 198  vol = 88.455
 core size 104 of 198  vol = 81.816
 core size 103 of 198  vol = 74.88
 core size 102 of 198  vol = 68.386
 core size 101 of 198  vol = 65.937
 core size 100 of 198  vol = 62.345
 core size 99 of 198  vol = 58.836
 core size 98 of 198  vol = 52.868
 core size 97 of 198  vol = 47.796
 core size 96 of 198  vol = 41.292
 core size 95 of 198  vol = 33.831
 core size 94 of 198  vol = 24.912
 core size 93 of 198  vol = 18.912
 core size 92 of 198  vol = 12.7
 core size 91 of 198  vol = 7.35
 core size 90 of 198  vol = 4.922
 core size 89 of 198  vol = 3.421
 core size 88 of 198  vol = 2.553
 core size 87 of 198  vol = 1.917
 core size 86 of 198  vol = 1.513
 core size 85 of 198  vol = 1.201
 core size 84 of 198  vol = 1.046
 core size 83 of 198  vol = 0.922
 core size 82 of 198  vol = 0.755
 core size 81 of 198  vol = 0.668
 core size 80 of 198  vol = 0.596
 core size 79 of 198  vol = 0.549
 core size 78 of 198  vol = 0.493
 FINISHED: Min vol ( 0.5 ) reached
```

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

# 79 positions (cumulative volume <= 0.5 Angstrom^3)
```

```
  start end length
1    10  25     16
2    28  48     21
3    53  94     42
```

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

```
rf <- rmsf(xyz)

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



If the predicted model has more than one domain, each domain may have high confidence, yet the relative positions of the domains may not. The estimated reliability of relative domain positions is in graphs of predicted aligned error (PAE) which are included in the downloadable zip file and analyzed in R above.

```
library(jsonlite)

# Listing of all PAE JSON files
pae_files <- list.files(path=results_dir, pattern=".*model.*\\.json", full.names = TRUE)
```

```
  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] 87.81 92.00 91.81 91.88 94.25 88.00
```

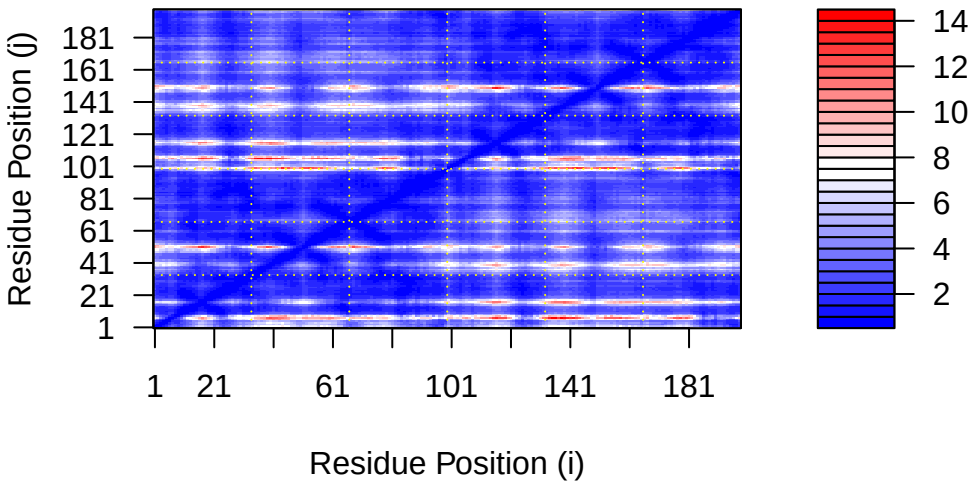
```
  pae1$max_pae
```

```
[1] 14.09375
```

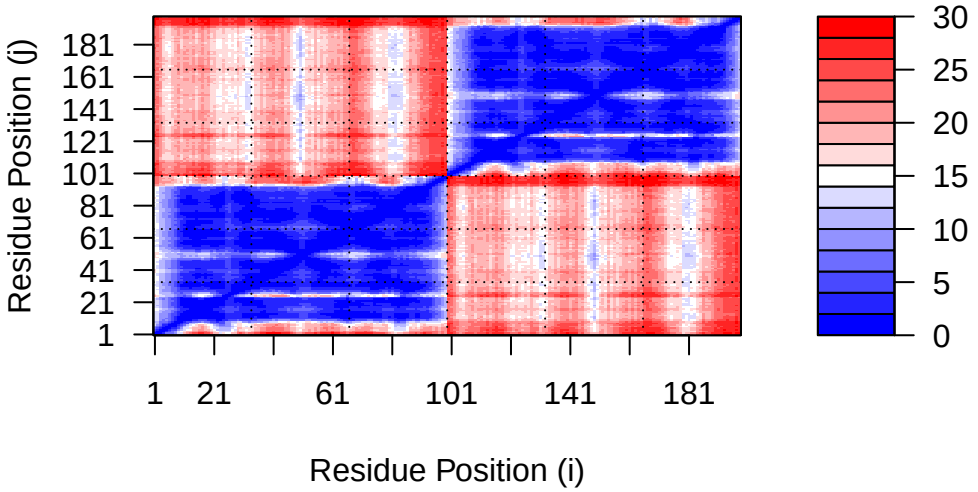
```
  pae5$max_pae
```

```
[1] 29.29688
```

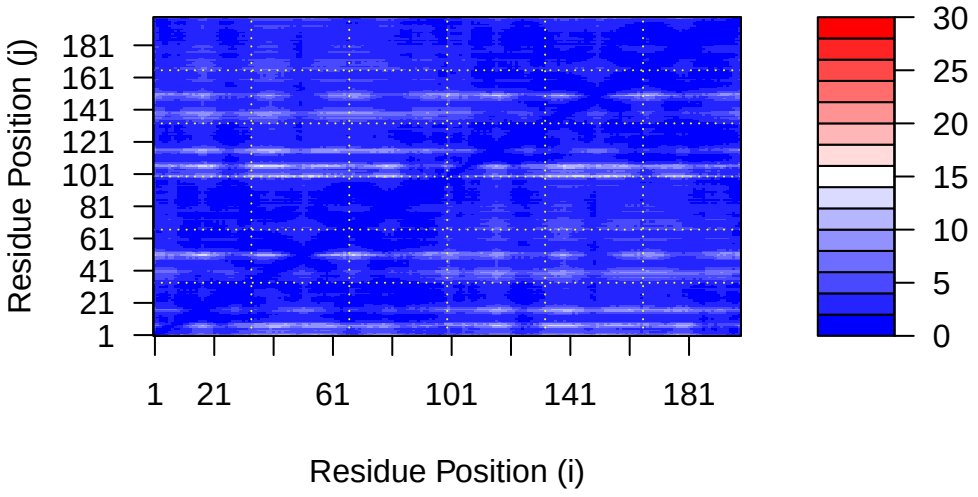
```
  plot.dmat(pae1$pae, xlab="Residue Position (i)", ylab="Residue Position (j)")
```




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



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




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

```
[1] "C:/Users/louis/Downloads/bimm143/R code/class11/HIV1prhomodimer_23119.result/HIV1prhomoc
```

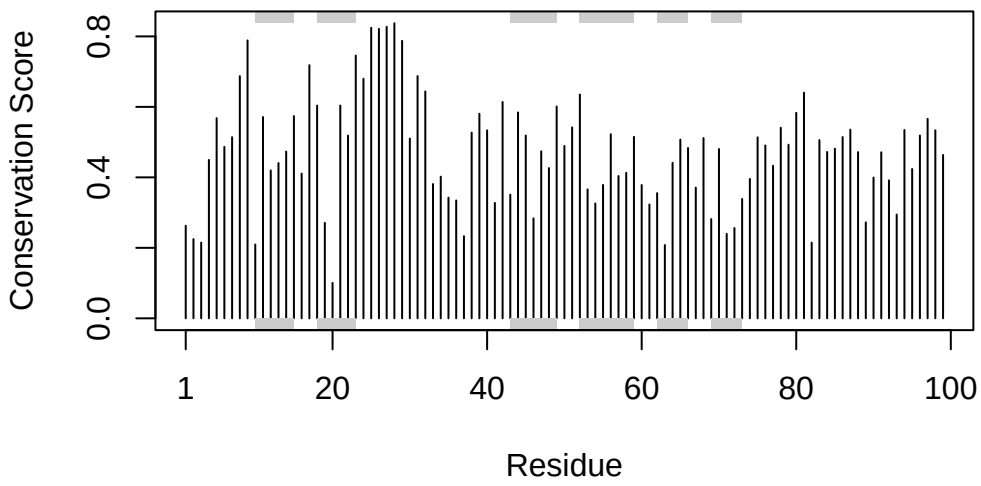
```
aln <- read.fasta(aln_file[1], to.upper = TRUE)
```

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

```
dim(aln$ali) #number of sequences in alignment
```

```
[1] 5378  132
```

```
sim <- conserv(aln)

plotb3(sim[1:99], sse=trim.pdb(pdb, chain="A"), ylab="Conservation Score")
```




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

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

```
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")
```