# Class 12: Structural Bioinformatics (pt2. Focus on new AlphaFold2)

Winnie Zhou (A16673200)

## Part 1 PCA of ADK

We will perform principal component analysis (PCA) on the complete collection of Adenylate kinase structures in the protein data-bank (PDB).

Adenylate kinase (ADK) is a ubiquitous enzyme that functions to maintain the equilibrium between cytoplasmic nucleotides by catalyzing the reversible transfer of a phosphoryl group from ATP to AMP.

Here we analyze all currently available ADK structures in the PDB to reveal detailed features and mechanistic principles of these essential shape changing transitions.

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

"msa" is found only on BioConductor.

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

"Grantlab/bio3d-view" 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

## Search and retrieve ADK structures

```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
```

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

Now we can use this sequence as a query to BLAST search the PDB to find similar sequences and structures.

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

We can now plot a summary of the search results

```
#hits <- plot(b)
#head(hits$pdb.id)
```

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

### Align and superpose structures

We can use `pdbaln()` function to align and superpose the identified PDB structures.

```
library(bio3d)
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)
```

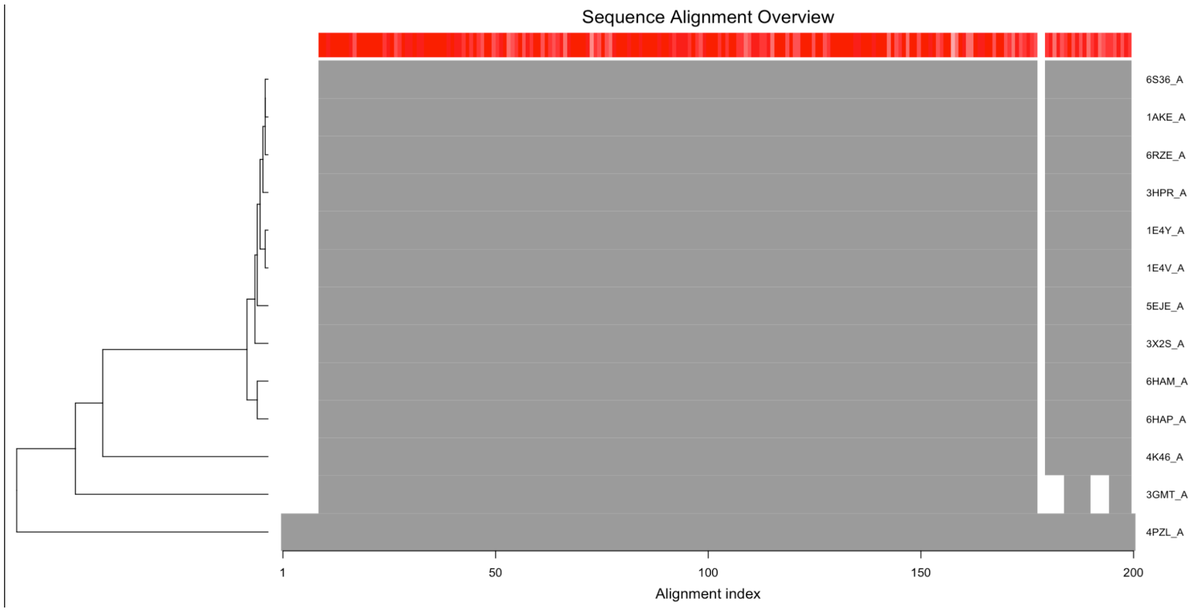


Figure 1: Plot of PDB structures (figure margins were too large so here's an image of the created plot)

## Annotate collected PDB structures

We can annotate the PDB files using `pdb.annotate()` function.

```
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 (ADK)
6S36_A       1.60             <NA>                   Adenylate kinase (ADK)
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 (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 (ADK)
```

```
6HAM_A       2.55             <NA>                      Adenylate kinase (ADK)
4K46_A       2.01             <NA> Adenylate kinase, active site lid (ADK_lid)
3GMT_A       2.10             <NA> Adenylate kinase, active site lid (ADK_lid)
4PZL_A       2.10             <NA> Adenylate kinase, active site lid (ADK_lid)
               ligandId
1AKE_A              AP5
6S36_A CL (3),NA,MG (2)
6RZE_A    NA (3),CL (2)
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,FMT,GOL
                                                                             ligandName
1AKE_A                                                 BIS(ADENOSINE)-5'-PENTAPHOSPHATE
6S36_A                                    CHLORIDE ION (3),SODIUM ION,MAGNESIUM ION (2)
6RZE_A                                                  SODIUM ION (3),CHLORIDE ION (2)
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,FORMIC ACID,GLYCEROL
                                                 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
```

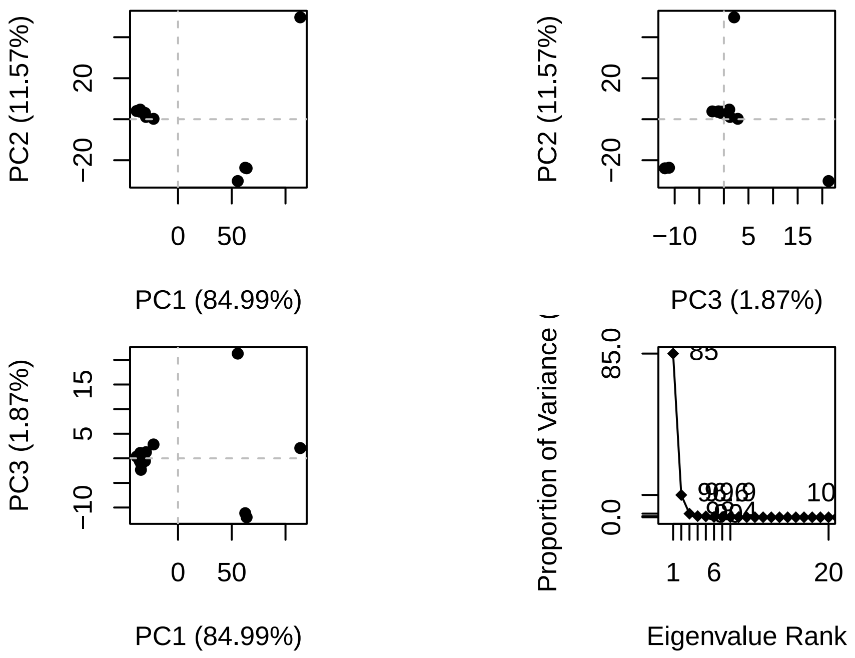
```
4K46_A                         Photobacterium profundum
3GMT_A                   Burkholderia pseudomallei 1710b
4PZL_A Francisella tularensis subsp. tularensis SCHU S4

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
1E4Y_A
3X2S_A
6HAP_A
6HAM_A
4K46_A
3GMT_A
4PZL_A                                                                              The crys
                                                     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
```

## Principal component analysis

We can now perform a PCA of the ADK structure data.

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



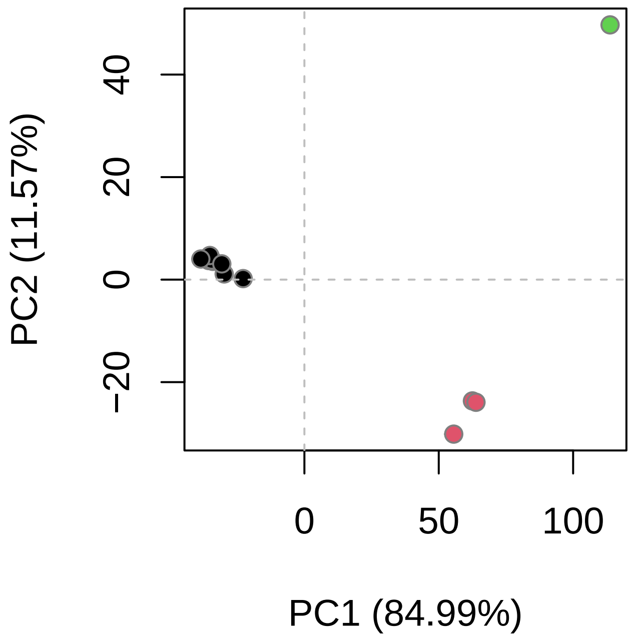
The `rmsd()` function calculates all pairwise RMSD values of the structural ensemble, facilitating clustering analysis based on the pairwise structural deviation.

```
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)
```



This plot shows a onformer plot – a low-dimensional representation of the conformational variability within the ensemble of PDB structures. Each dot on this plot represents one PDB structure.

## Part 2 Alphafold Dimer structure prediction

Section 8: Custom analysis of resulting models

We can move our AlphaFold results directory into our RStudio project directory.

```
results_dir <- "hivprdimer_23119"
```

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

```
[1] "hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb"
```

```
[2] "hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb"
[3] "hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb"
[4] "hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb"
[5] "hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb"
```

```
library(bio3d)
```

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

```
Reading PDB files:
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
.....

Extracting sequences

pdb/seq: 1   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_001_alphafold2_multimer_
pdb/seq: 2   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_002_alphafold2_multimer_
pdb/seq: 3   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_003_alphafold2_multimer_
pdb/seq: 4   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_004_alphafold2_multimer_
pdb/seq: 5   name: hivprdimer_23119/hivprdimer_23119_unrelaxed_rank_005_alphafold2_multimer_
```

```
pdbs
```

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

                               51        .         .         .         .         100
[Truncated_Name:1]hivprdimer    GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:2]hivprdimer    GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
```

```
[Truncated_Name:3]hivprdimer   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:4]hivprdimer   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:5]hivprdimer   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
                               **************************************************
                              51         .         .         .         .         100

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

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

RMSD is a standard measure of structural distance between coordinate sets. We can use the `rmsd()` function to calculate the RMSD between all pairs models.

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

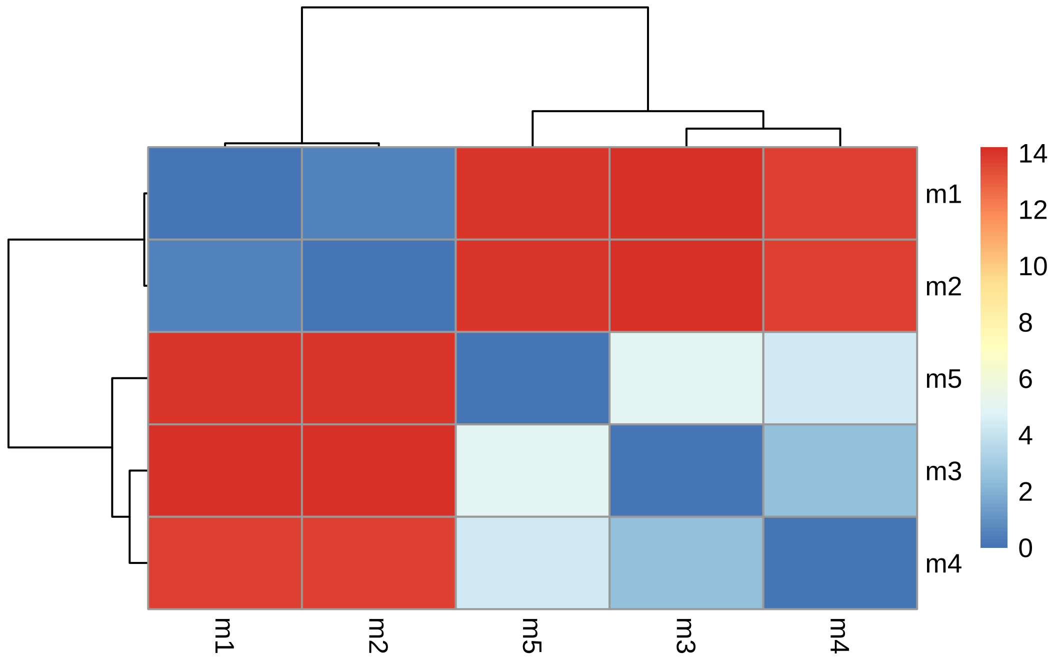
```
Warning in rmsd(pdbs, fit = T): No indices provided, using the 198 non NA positions
```

```
range(rd)
```

```
[1]  0.000 14.203
```

Let's draw a heatmap of these RMSD matrix values.

```
library(pheatmap)

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



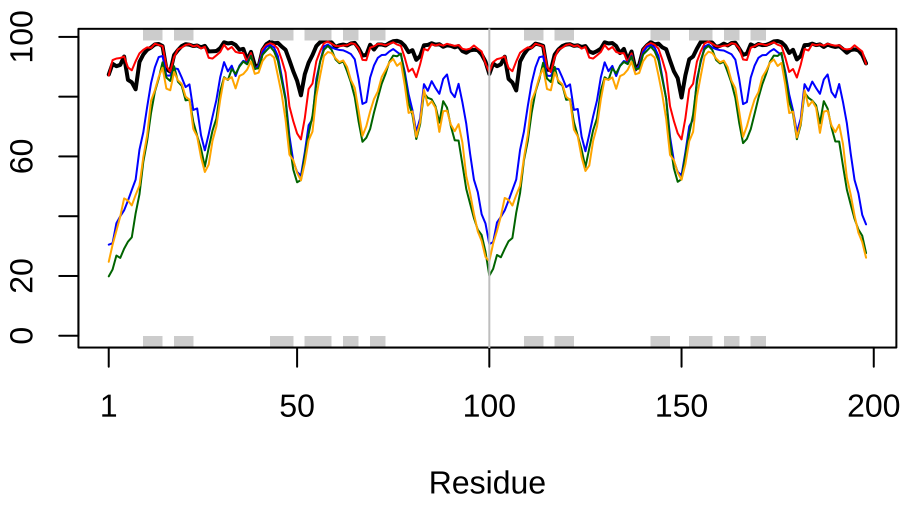
The heatmap shows us that 1 and 2 are more similar to each other than to the other models. Models 4 and 5 are also similar to each other and more similar to model 3 than to models 1 and 2.

Let's plot the pLDDT values across all models.

```
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")
```



We can improve the superposition of our models.

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

```
core size 197 of 198  vol = 3124.961
core size 196 of 198  vol = 2920.851
core size 195 of 198  vol = 2759.61
core size 194 of 198  vol = 2621.721
core size 193 of 198  vol = 2506.06
core size 192 of 198  vol = 2431.57
core size 191 of 198  vol = 2389.645
core size 190 of 198  vol = 2361.021
core size 189 of 198  vol = 2369.16
core size 188 of 198  vol = 2363.057
core size 187 of 198  vol = 2376.959
core size 186 of 198  vol = 2355.91
core size 185 of 198  vol = 2346.191
core size 184 of 198  vol = 2306.353
core size 183 of 198  vol = 2255.869
core size 182 of 198  vol = 2190.649
core size 181 of 198  vol = 2116.308
core size 180 of 198  vol = 1992.733
core size 179 of 198  vol = 1949.969
core size 178 of 198  vol = 1893.838
core size 177 of 198  vol = 1829.766
core size 176 of 198  vol = 1752.857
core size 175 of 198  vol = 1678.022
core size 174 of 198  vol = 1604.938
core size 173 of 198  vol = 1562.773
core size 172 of 198  vol = 1530.882
core size 171 of 198  vol = 1484.989
core size 170 of 198  vol = 1440.135
core size 169 of 198  vol = 1398.99
core size 168 of 198  vol = 1357.399
core size 167 of 198  vol = 1314.429
core size 166 of 198  vol = 1266.817
core size 165 of 198  vol = 1225.544
core size 164 of 198  vol = 1188.065
core size 163 of 198  vol = 1156.619
core size 162 of 198  vol = 1091.277
core size 161 of 198  vol = 1058.55
core size 160 of 198  vol = 1010.855
core size 159 of 198  vol = 983.008
core size 158 of 198  vol = 957.153
core size 157 of 198  vol = 931.981
core size 156 of 198  vol = 905.995
core size 155 of 198  vol = 869.254
core size 154 of 198  vol = 843.978
core size 153 of 198  vol = 811.745
core size 152 of 198  vol = 783.826
core size 151 of 198  vol = 758.742
core size 150 of 198  vol = 728.577
core size 149 of 198  vol = 698.246
core size 148 of 198  vol = 678.143
core size 147 of 198  vol = 652.415
core size 146 of 198  vol = 636.917
core size 145 of 198  vol = 619.772
core size 144 of 198  vol = 603.091
core size 143 of 198  vol = 587.741
core size 142 of 198  vol = 573.444
core size 141 of 198  vol = 557.195
core size 140 of 198  vol = 539.015
core size 139 of 198  vol = 517.718
core size 138 of 198  vol = 496.614
core size 137 of 198  vol = 475.524
core size 136 of 198  vol = 459.064
core size 135 of 198  vol = 445.606
core size 134 of 198  vol = 430.657
core size 133 of 198  vol = 410.332
core size 132 of 198  vol = 401.044
core size 131 of 198  vol = 388.238
core size 130 of 198  vol = 375.86
core size 129 of 198  vol = 364.108
core size 128 of 198  vol = 350.533
core size 127 of 198  vol = 341.561
core size 126 of 198  vol = 328.474
core size 125 of 198  vol = 314.511
core size 124 of 198  vol = 300.607
core size 123 of 198  vol = 288.935
core size 122 of 198  vol = 277.27
core size 121 of 198  vol = 264.382
core size 120 of 198  vol = 254.352
core size 119 of 198  vol = 242.691
core size 118 of 198  vol = 231.991
core size 117 of 198  vol = 221.382
core size 116 of 198  vol = 212.788
core size 115 of 198  vol = 203.834
core size 114 of 198  vol = 194.898
core size 113 of 198  vol = 184.082
core size 112 of 198  vol = 172.93
core size 111 of 198  vol = 162.111
core size 110 of 198  vol = 151.154
core size 109 of 198  vol = 141.921
core size 108 of 198  vol = 131.714
core size 107 of 198  vol = 124.278
core size 106 of 198  vol = 118.708
core size 105 of 198  vol = 112.734
core size 104 of 198  vol = 106.464
core size 103 of 198  vol = 100.447
core size 102 of 198  vol = 92.93
```

```
 core size 101 of 198  vol = 84.911
 core size 100 of 198  vol = 77.129
 core size 99 of 198  vol = 70.021
 core size 98 of 198  vol = 62.159
 core size 97 of 198  vol = 54.55
 core size 96 of 198  vol = 47.345
 core size 95 of 198  vol = 42.479
 core size 94 of 198  vol = 37.149
 core size 93 of 198  vol = 29.658
 core size 92 of 198  vol = 22.749
 core size 91 of 198  vol = 14.984
 core size 90 of 198  vol = 7.932
 core size 89 of 198  vol = 4.439
 core size 88 of 198  vol = 3.189
 core size 87 of 198  vol = 2.468
 core size 86 of 198  vol = 1.901
 core size 85 of 198  vol = 1.633
 core size 84 of 198  vol = 1.295
 core size 83 of 198  vol = 1.019
 core size 82 of 198  vol = 0.867
 core size 81 of 198  vol = 0.722
 core size 80 of 198  vol = 0.618
 core size 79 of 198  vol = 0.532
 core size 78 of 198  vol = 0.506
 core size 77 of 198  vol = 0.474
 FINISHED: Min vol ( 0.5 ) reached
```

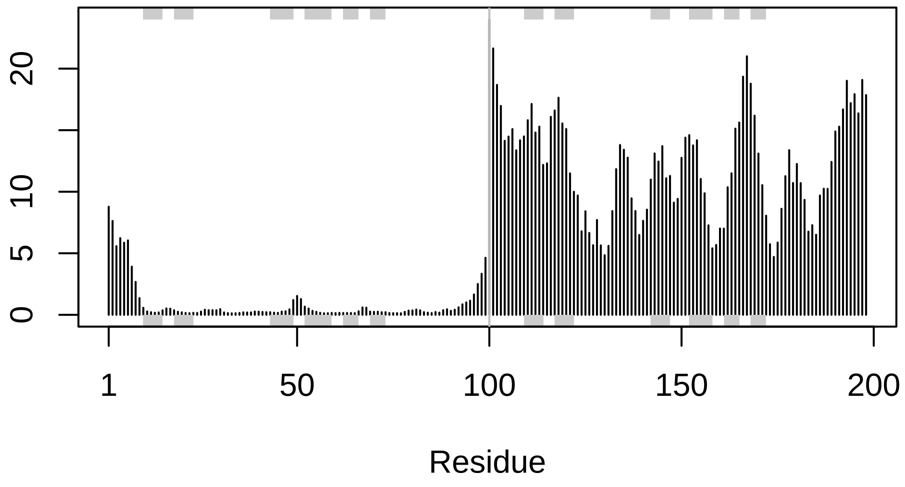
```
core.inds <- print(core, vol=0.5)
```

```
# 78 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1    10  48     39
2    53  66     14
3    68  92     25
```

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

We can examine the RMSF (root mean square fluctuations) between positions of the structure. RMSF is an often used measure of conformational variance along the structure.

```r
rf <- rmsf(xyz)

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



The first chain is very similar across the different models.

## Lets predict the alignment error for domains

```r
library(jsonlite)

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

```r
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"
```

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

```
[1] 87.38 91.00 90.19 90.62 93.44 85.62
```

The lower the PAE scores the better. Let's look at the max PAE scores of the first and fifth files.

```
pae1$max_pae
```

```
[1] 15.875
```

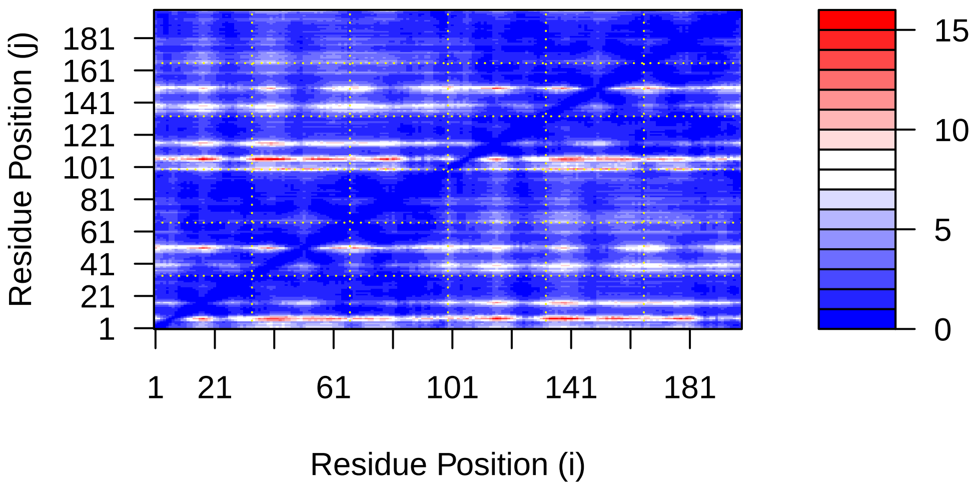
```
pae5$max_pae
```

```
[1] 29.23438
```

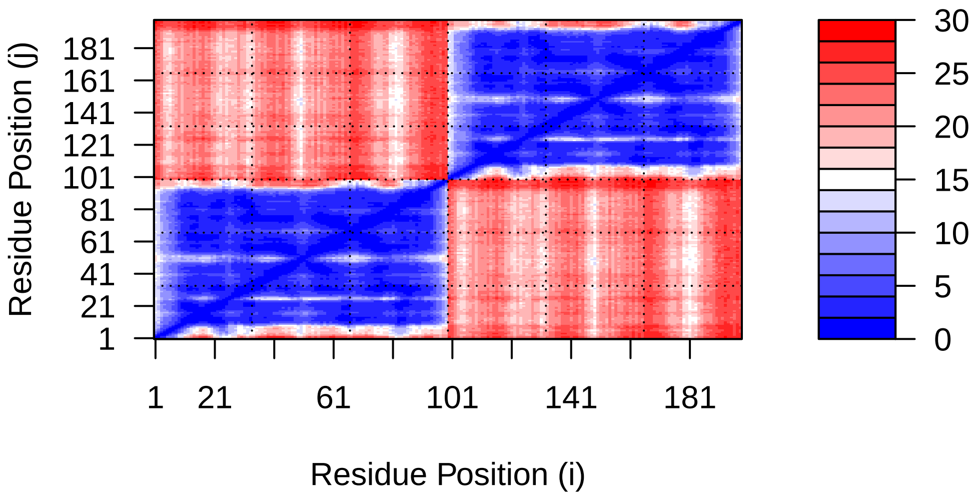
Model 1 has a lower PAE score than 5, so it is a better model.

We can plot the number of residues (N) by N PAE scores with ggplot or with functions from the Bio3D package:

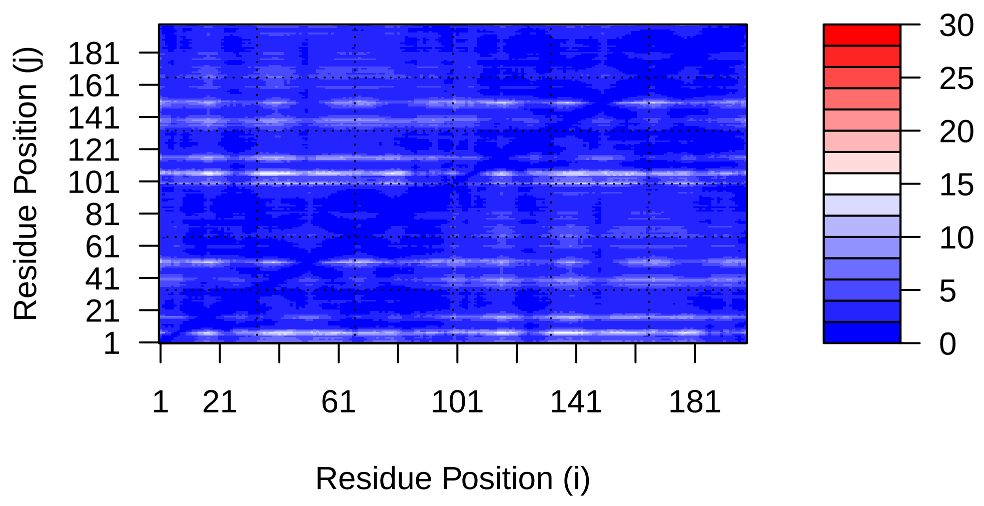
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```



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



```
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] "hivprdimer_23119/hivprdimer_23119.a3m"
```

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

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

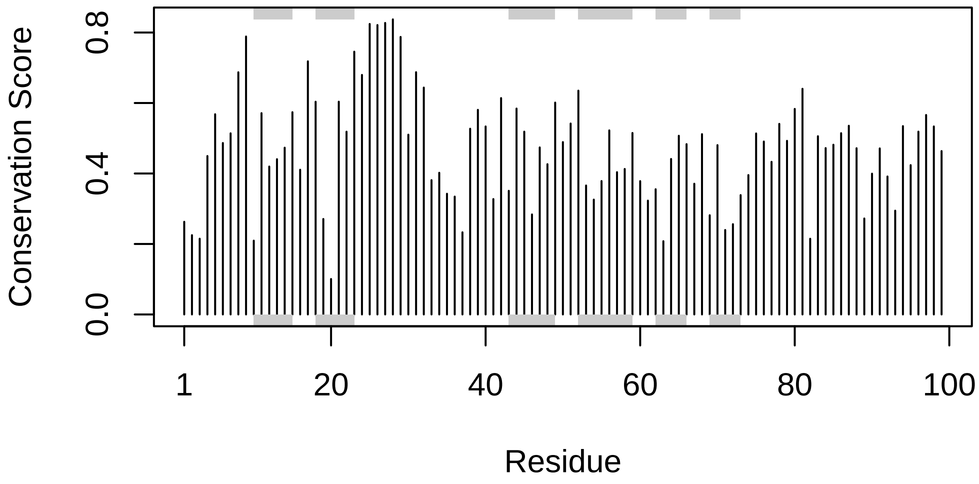
Q. How many sequences are in this alignment?

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

```
[1] 5378  132
```

We can use `conserv()` function to score residue conservation in the alignment.

```
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] "-" "-" "-" "-" "-" "-"
```

With a high cutoff of 0.9, the conserved Active Site residues D25, T26, G27, A28 will stand out.

We can visualize these sites by mapping this conservation score to the occupancy column of a PDB file for viewing in molecular viewer programs such as Mol*.

```
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")
```

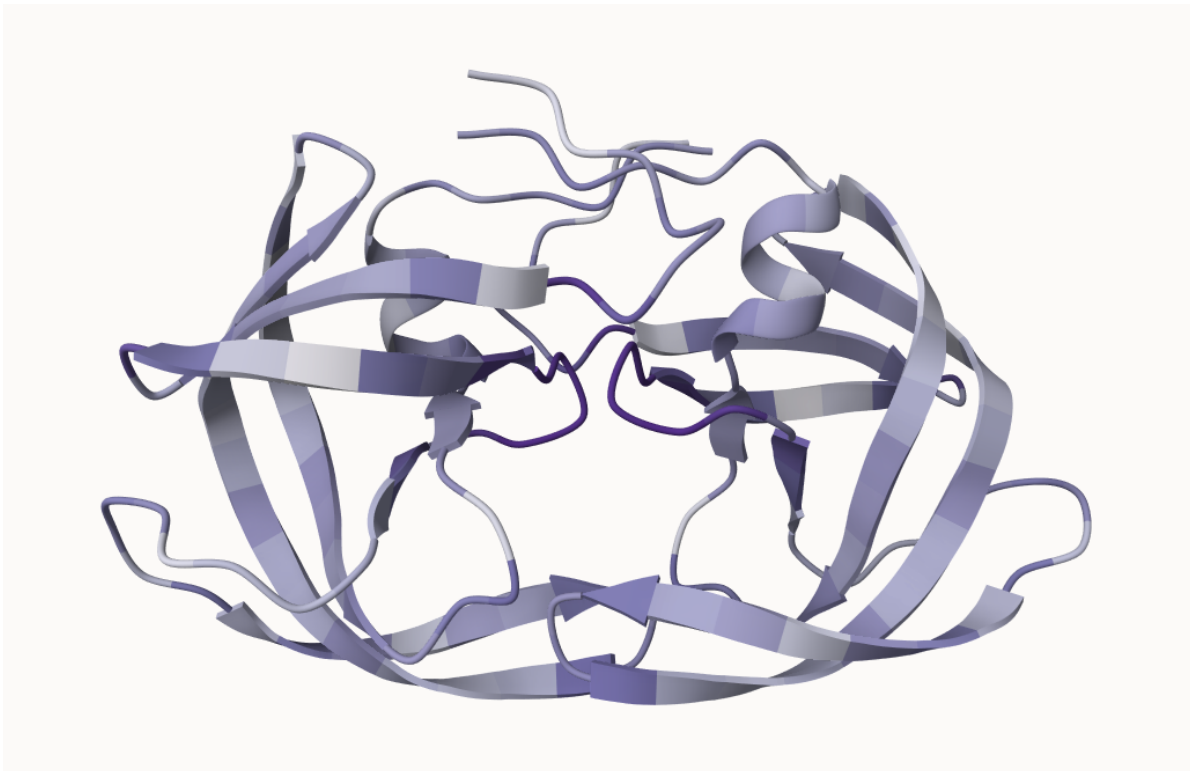


Figure 2: This is the best ranked dimer model.