AlphaFold Analysis

Emily Ignatoff (A16732102)

AlphaFold Analysis of find-a-gene sequence:

Here we analyze our AlphaFold structure prediction models for our HIV sequence:

```
results_dir <- "hivpr_monomer_94b5b/"
```

First, I will read in each generated pdb file from the results directory:

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

basename(pdb_files)
```

```
[1] "hivpr_monomer_94b5b_unrelaxed_rank_001_alphafold2_ptm_model_5_seed_000.pdb"
[2] "hivpr_monomer_94b5b_unrelaxed_rank_002_alphafold2_ptm_model_4_seed_000.pdb"
[3] "hivpr_monomer_94b5b_unrelaxed_rank_003_alphafold2_ptm_model_1_seed_000.pdb"
[4] "hivpr_monomer_94b5b_unrelaxed_rank_004_alphafold2_ptm_model_3_seed_000.pdb"
[5] "hivpr_monomer_94b5b_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pdb"
```

Now, I will superimpose each pdb on top of the others:

```
library(bio3d)
```

```
Warning: package 'bio3d' was built under R version 4.3.3
```

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

```
Reading PDB files:
hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_001_alphafold2_ptm_model_5_seed_000.pd
hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_002_alphafold2_ptm_model_4_seed_000.pd
hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_003_alphafold2_ptm_model_1_seed_000.pd
hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_004_alphafold2_ptm_model_3_seed_000.pd
hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_005_alphafold2_ptm_model_2_seed_000.pd
.....

Extracting sequences

pdb/seq: 1   name: hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_001_alphafold2_ptm_
pdb/seq: 2   name: hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_002_alphafold2_ptm_
pdb/seq: 3   name: hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_003_alphafold2_ptm_
pdb/seq: 4   name: hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_004_alphafold2_ptm_
pdb/seq: 5   name: hivpr_monomer_94b5b/hivpr_monomer_94b5b_unrelaxed_rank_005_alphafold2_ptm_
```

```r
pdbs
```

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

                              51        .         .         .         .         99 
[Truncated_Name:1]hivpr_mono   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:2]hivpr_mono   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:3]hivpr_mono   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:4]hivpr_mono   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:5]hivpr_mono   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
                               ************************************************* 
                              51        .         .         .         .         99 

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

Class:
  pdbs, fasta
```

```
Alignment dimensions:
  5 sequence rows; 99 position columns (99 non-gap, 0 gap)

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

Now, I will generate the structural difference between these coordinate sets using `rmsd()`:

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

```
Warning in rmsd(pdbs, fit = T): No indices provided, using the 99 non NA positions
```

```r
range(rd)
```

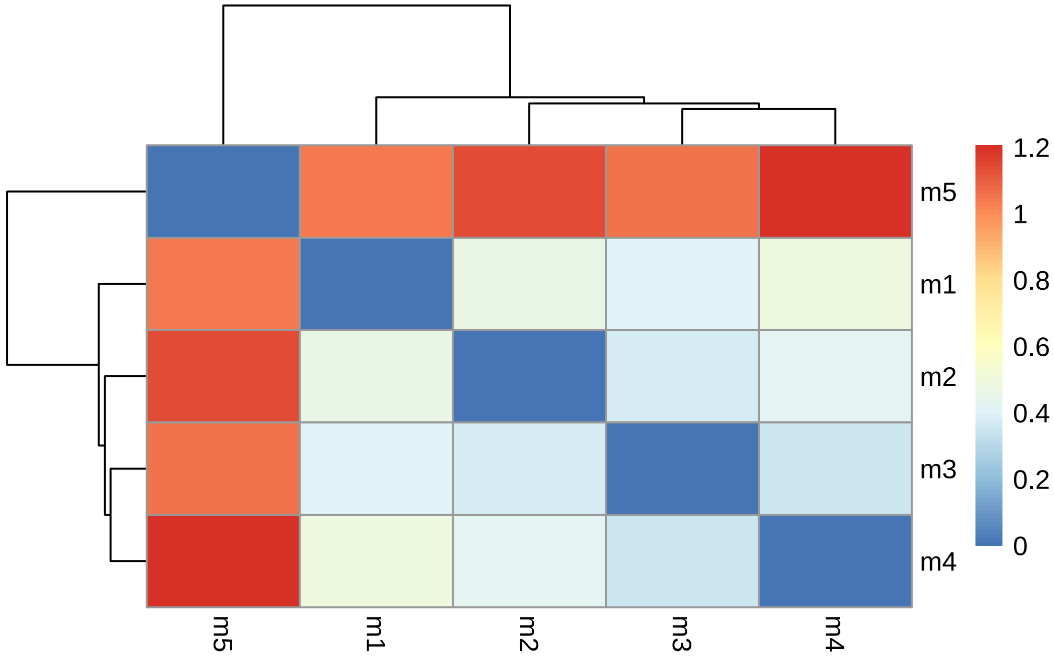
```
[1] 0.000 1.206
```

Heatmap of RMSD values:

```r
library(pheatmap)
```

```
Warning: package 'pheatmap' was built under R version 4.3.3
```

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



I will be using a human HSP70 protein from the PDB database as my reference protein for the following steps:

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

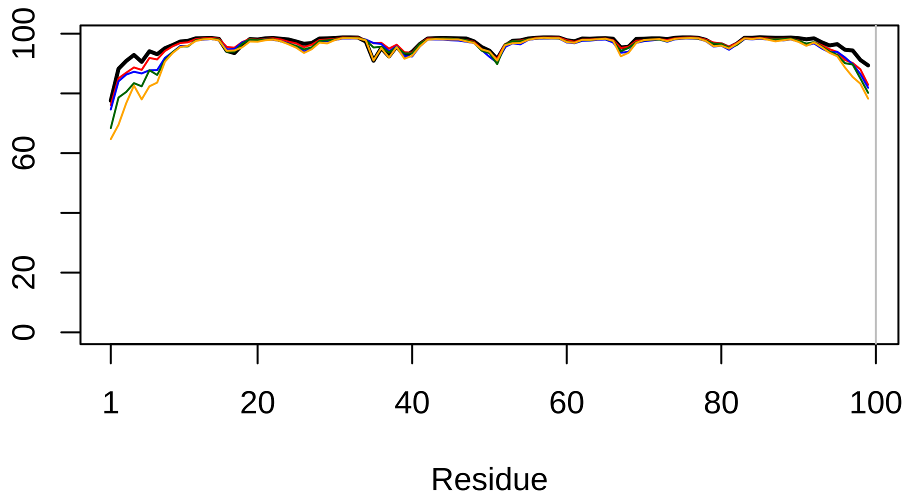
```
  Note: Accessing on-line PDB file
```

Now, I will plot the pLDDT values of each model against each other:

```
plotb3(pdbs$b[1,], typ="l", lwd=2, sse=pdb)
```

```
Warning in plotb3(pdbs$b[1, ], typ = "l", lwd = 2, sse = pdb): Length of input
'sse' does not equal the length of input 'x'; Ignoring 'sse'
```

```
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")
```



I will now find the the most "rigid" or consistent portion across models using the `core.find()` function:

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

```
 core size 98 of 99  vol = 3.66 
 core size 97 of 99  vol = 2.756 
 core size 96 of 99  vol = 2.236 
 core size 95 of 99  vol = 1.751 
 core size 94 of 99  vol = 1.386 
 core size 93 of 99  vol = 0.991 
 core size 92 of 99  vol = 0.769 
 core size 91 of 99  vol = 0.568 
 core size 90 of 99  vol = 0.422 
 FINISHED: Min vol ( 0.5 ) reached
```

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

```
# 91 positions (cumulative volume <= 0.5 Angstrom^3) 
  start end length
1     3   3      1
2     7  96     90
```

Writing the superimposed coordinates to a new file for use in Mol*

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

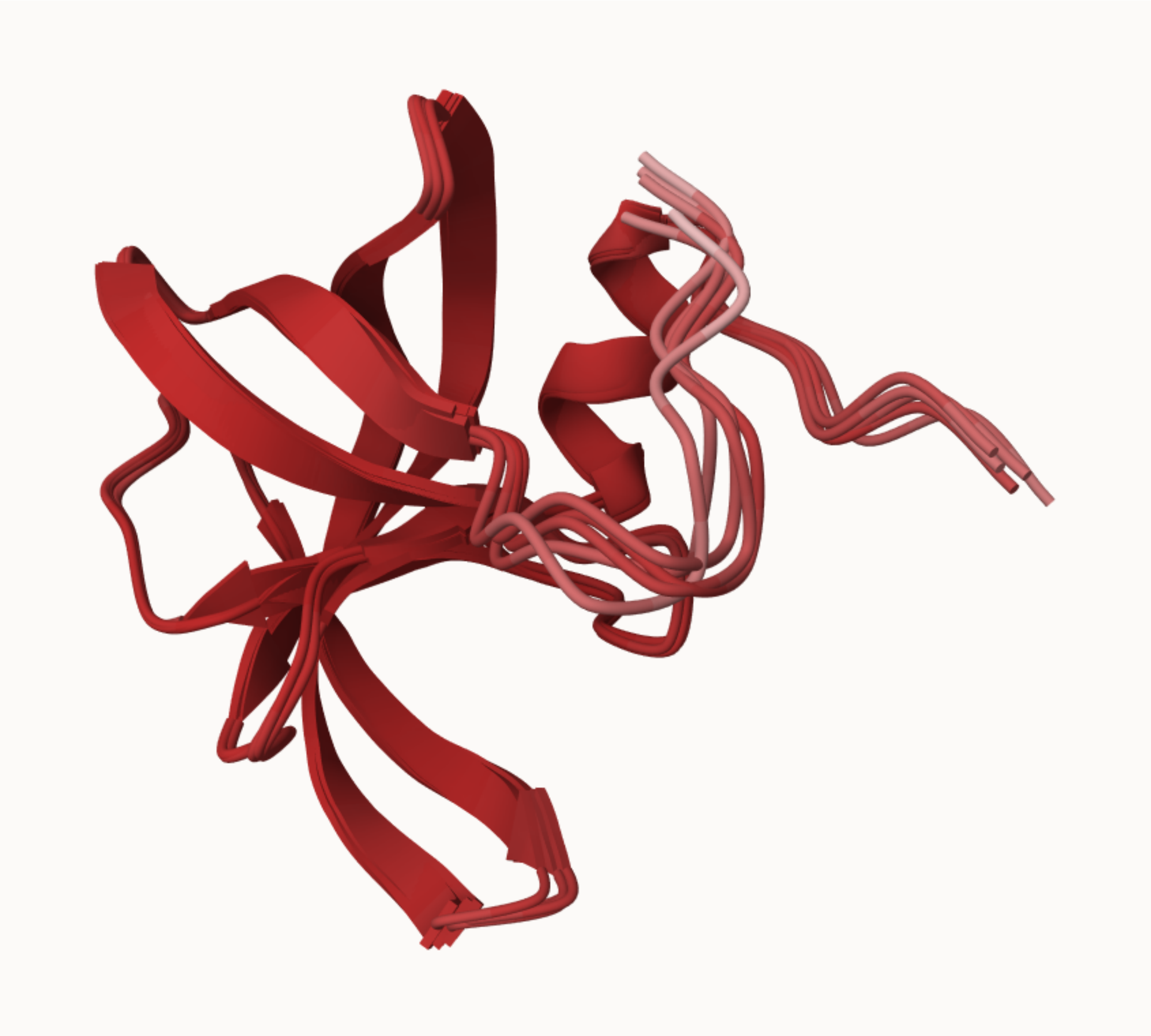


Figure 1: Uncertainty/disorder of strucutres in Mol*

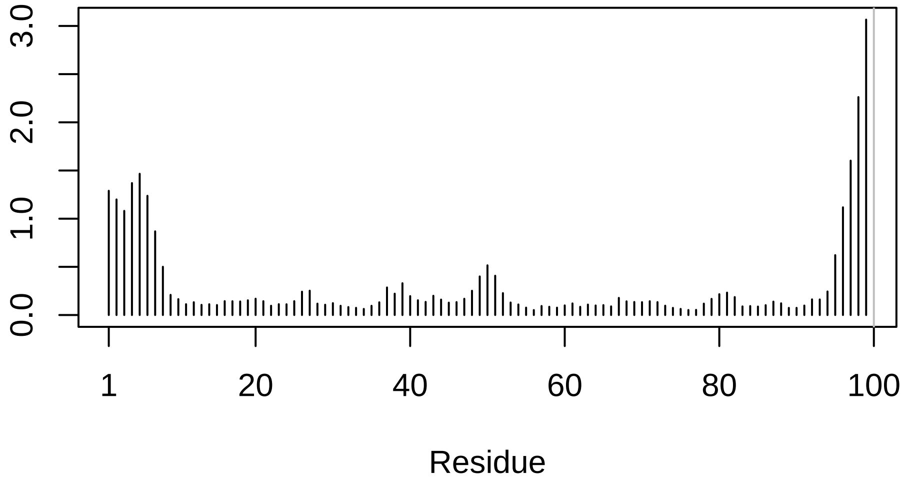
I will now use RMSF to observe the conformational variance across the structure:

```
rf <- rmsf(xyz)

plotb3(rf, sse=pdb)
```

```
Warning in plotb3(rf, sse = pdb): Length of input 'sse' does not equal the
length of input 'x'; Ignoring 'sse'
```

```
abline(v=100, col="gray", ylab="RMSF")
```



> Note: I am unsure why the sse function is not working using an existing PDB file as my reference protein :(

Predicted Alignment Error for Domains

```
library(jsonlite)
```

```
Warning: package 'jsonlite' was built under R version 4.3.3
```

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

```r
pae1 <- read_json(pae_files[1],simplifyVector = TRUE)
pae2 <- read_json(pae_files[2],simplifyVector = TRUE)
pae3 <- read_json(pae_files[3],simplifyVector = TRUE)
pae4 <- read_json(pae_files[4],simplifyVector = TRUE)
pae5 <- read_json(pae_files[5],simplifyVector = TRUE)

attributes(pae1)
```

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

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

```
[1] 77.56 88.25 90.88 92.88 90.56 94.12
```

Which model has the best (lowest) max PAE?

```r
pae1$max_pae
```

```
[1] 17.67188
```

```r
pae2$max_pae
```

```
[1] 16.9375
```

```r
pae3$max_pae
```

```
[1] 19.25
```

```r
pae4$max_pae
```

```
[1] 21.48438
```

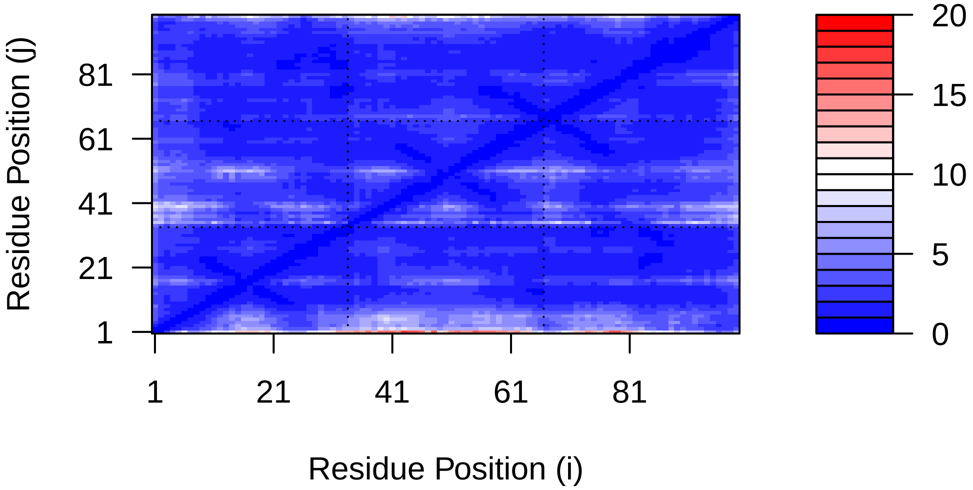
```r
pae5$max_pae
```

```
[1] 20.95312
```

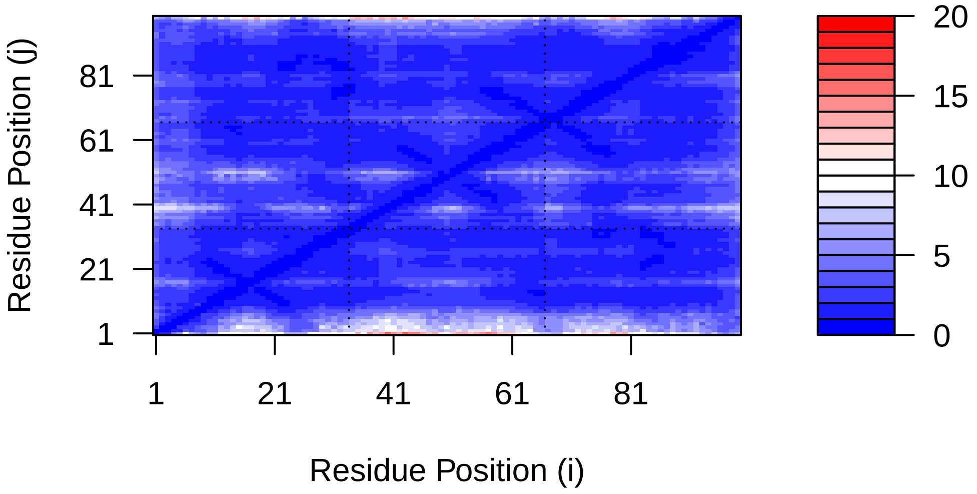
PAE 2 is lowest in this case!

I will plot each of these PAE in the same range:

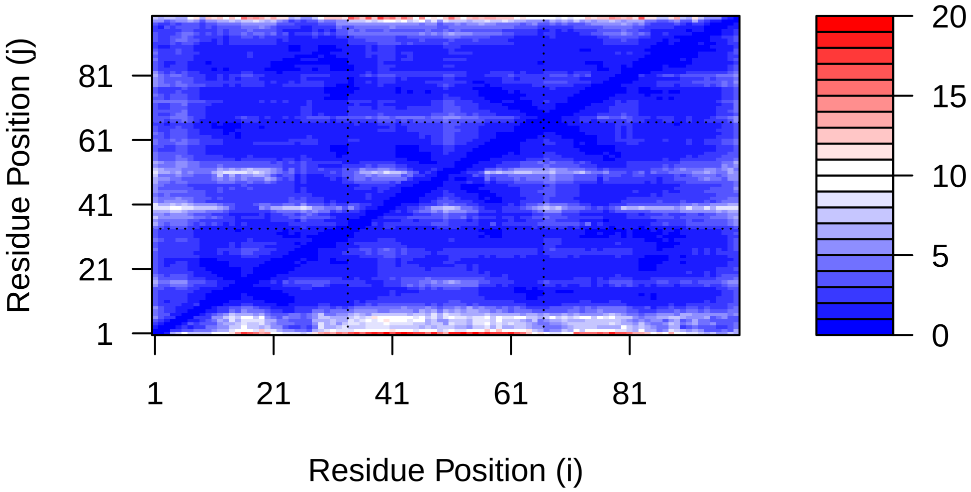
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
           grid.col = "black",
          zlim= c(0,20))
```



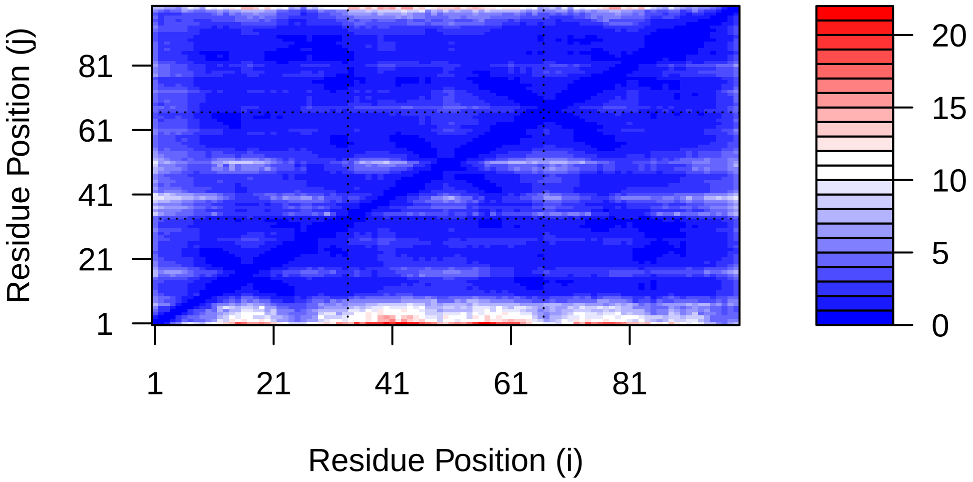
```
plot.dmat(pae2$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black",
          zlim= c(0,20))
```



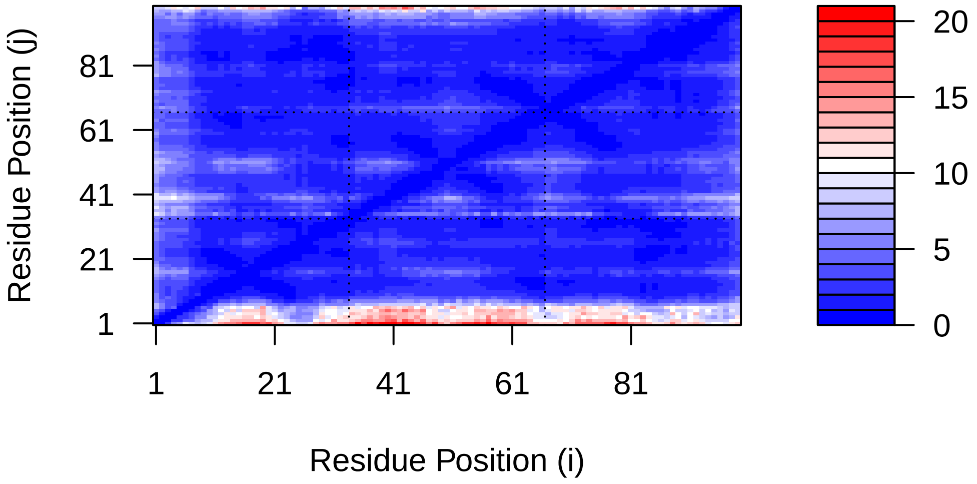
```
plot.dmat(pae3$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black")
```



```r
plot.dmat(pae4$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)",
          grid.col = "black")
```



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



Residue Conservation from Alignment File

Generate alignment file:

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

```
[1] "hivpr_monomer_94b5b/hivpr_monomer_94b5b.a3m"
```

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

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

How many sequences in this alignment?

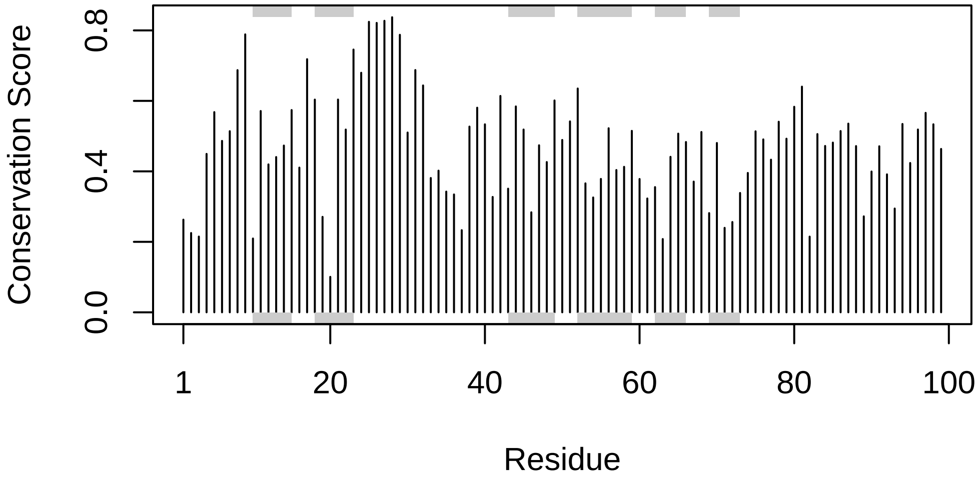
```
dim(aln$ali)
```

```
[1] 5378  132
```

I will score residue conservation using the `conserv()` function:

```
sim <- conserv(aln)
```

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



Let's look at the sequences which will stand out in a consensus sequence

```
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(sim[1:length(unique(m1.pdb$atom$resno))], m1.pdb$atom$resno)
write.pdb(m1.pdb, o=occ, file="m1_conserv.pdb")
```

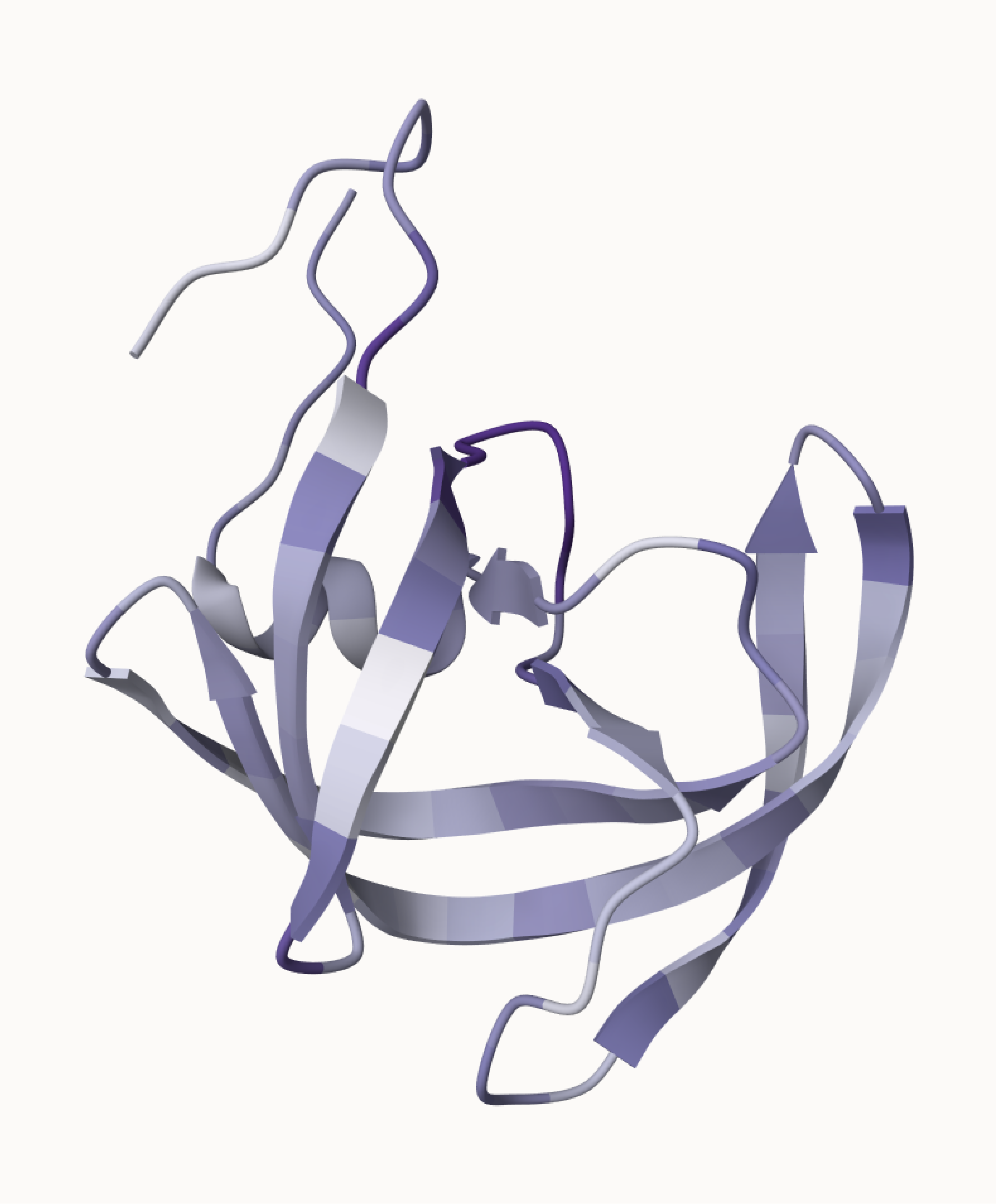


Figure 2: Color by Occupancy in Mol*