Structural Bioinformatics (pt.2)

Farnam (PID: A17628539)

AlphaFold has changed the game for protein structure prediction and allow 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 output 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 by loading these structures up in Mol*

```
library(bio3d)
```

```
results_dir <- "HIVpr2_23119/"
```

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

# Print our PDB file names
basename(pdb_files)
```

```
[1] "HIVpr2_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb"
[2] "HIVpr2_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb"
[3] "HIVpr2_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb"
[4] "HIVpr2_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb"
[5] "HIVpr2_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:
HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb
HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb
HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb
HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb
HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb
.....

Extracting sequences

pdb/seq: 1   name: HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model
pdb/seq: 2   name: HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model
pdb/seq: 3   name: HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model
pdb/seq: 4   name: HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model
pdb/seq: 5   name: HIVpr2_23119//HIVpr2_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model
```

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

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

```
rd
```

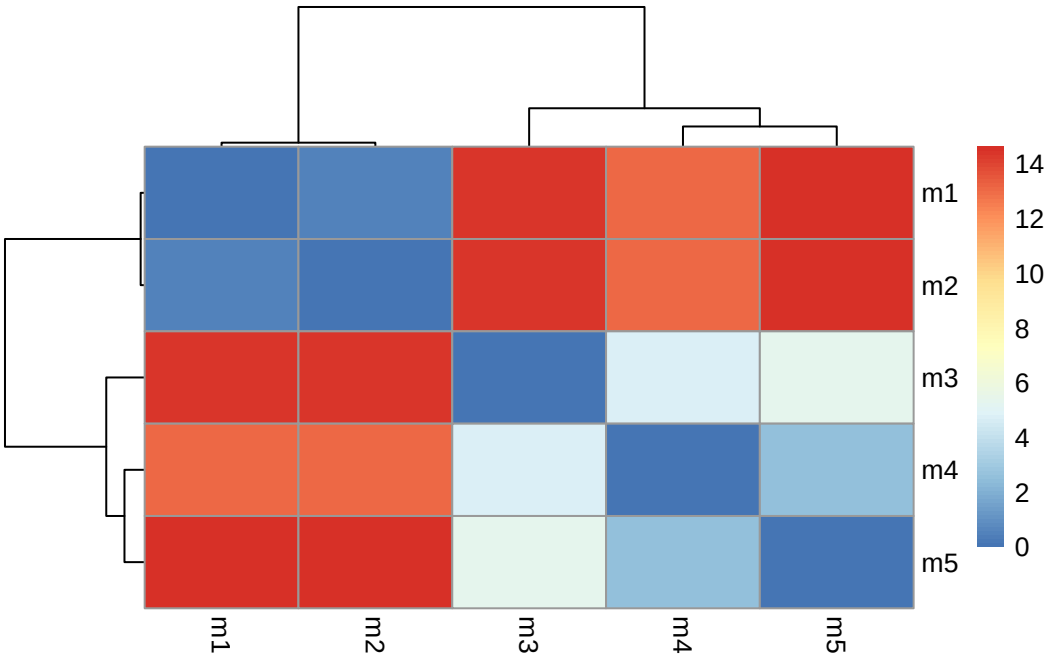
```
                                                                    HIVpr2_23119_unrelaxe
HIVpr2_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
HIVpr2_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
HIVpr2_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
HIVpr2_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
HIVpr2_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
                                                                    HIVpr2_23119_unrelaxe
HIVpr2_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
HIVpr2_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
HIVpr2_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
HIVpr2_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
HIVpr2_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
                                                                    HIVpr2_23119_unrelaxe
HIVpr2_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
```

```
HIVpr2_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
HIVpr2_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
HIVpr2_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
HIVpr2_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
                                                                HIVpr2_23119_unrelaxe
HIVpr2_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
HIVpr2_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
HIVpr2_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
HIVpr2_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
HIVpr2_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
                                                                HIVpr2_23119_unrelaxe
HIVpr2_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000
HIVpr2_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000
HIVpr2_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000
HIVpr2_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000
HIVpr2_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000
```

```
range(rd)
```

```
[1]  0.000 14.631
```

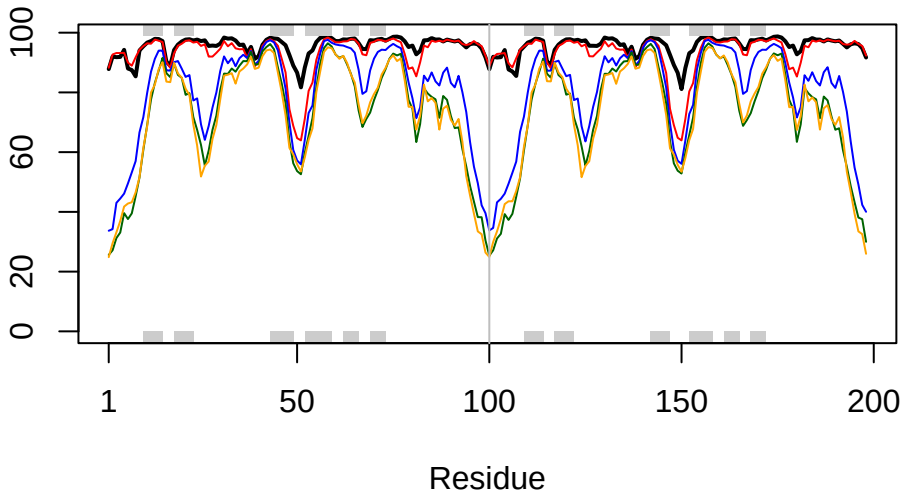
```
library(pheatmap)

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




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

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

```
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 = 4573.887
core size 196 of 198  vol = 3931.77
core size 195 of 198  vol = 3708.443
core size 194 of 198  vol = 3498.324
core size 193 of 198  vol = 3305.211
core size 192 of 198  vol = 3146.93
core size 191 of 198  vol = 3049.552
core size 190 of 198  vol = 2969.133
core size 189 of 198  vol = 2891.554
core size 188 of 198  vol = 2829.335
core size 187 of 198  vol = 2771.216
core size 186 of 198  vol = 2724.777
core size 185 of 198  vol = 2701.216
core size 184 of 198  vol = 2698.229
core size 183 of 198  vol = 2711.762
core size 182 of 198  vol = 2807.147
core size 181 of 198  vol = 2887.614
core size 180 of 198  vol = 2966.39
core size 179 of 198  vol = 3013.423
core size 178 of 198  vol = 3039.454
```

```
core size 177 of 198  vol = 3032.53
core size 176 of 198  vol = 3031.31
core size 175 of 198  vol = 2997.817
core size 174 of 198  vol = 2962.274
core size 173 of 198  vol = 2888.678
core size 172 of 198  vol = 2802.227
core size 171 of 198  vol = 2740.279
core size 170 of 198  vol = 2677.494
core size 169 of 198  vol = 2613.689
core size 168 of 198  vol = 2544.162
core size 167 of 198  vol = 2486.156
core size 166 of 198  vol = 2416.503
core size 165 of 198  vol = 2352.246
core size 164 of 198  vol = 2291.378
core size 163 of 198  vol = 2229.064
core size 162 of 198  vol = 2164.937
core size 161 of 198  vol = 2087.506
core size 160 of 198  vol = 2023.689
core size 159 of 198  vol = 1945.296
core size 158 of 198  vol = 1875.586
core size 157 of 198  vol = 1796.387
core size 156 of 198  vol = 1724.287
core size 155 of 198  vol = 1668.221
core size 154 of 198  vol = 1595.319
core size 153 of 198  vol = 1526.594
core size 152 of 198  vol = 1452.503
core size 151 of 198  vol = 1392.525
core size 150 of 198  vol = 1327.898
core size 149 of 198  vol = 1266.131
core size 148 of 198  vol = 1214.055
core size 147 of 198  vol = 1170.624
core size 146 of 198  vol = 1133.152
core size 145 of 198  vol = 1096.74
core size 144 of 198  vol = 1044.472
core size 143 of 198  vol = 1008.986
core size 142 of 198  vol = 966.045
core size 141 of 198  vol = 923.606
core size 140 of 198  vol = 884.908
core size 139 of 198  vol = 843.34
core size 138 of 198  vol = 802.29
core size 137 of 198  vol = 771.688
core size 136 of 198  vol = 739.939
core size 135 of 198  vol = 712.765
```

```
core size 134 of 198  vol = 687.256
core size 133 of 198  vol = 657.949
core size 132 of 198  vol = 628.927
core size 131 of 198  vol = 595.344
core size 130 of 198  vol = 564.914
core size 129 of 198  vol = 530.679
core size 128 of 198  vol = 495.179
core size 127 of 198  vol = 462.53
core size 126 of 198  vol = 431.298
core size 125 of 198  vol = 408.352
core size 124 of 198  vol = 375.994
core size 123 of 198  vol = 361.786
core size 122 of 198  vol = 352.972
core size 121 of 198  vol = 330.943
core size 120 of 198  vol = 311.606
core size 119 of 198  vol = 285.832
core size 118 of 198  vol = 261.516
core size 117 of 198  vol = 244.41
core size 116 of 198  vol = 227.782
core size 115 of 198  vol = 209.712
core size 114 of 198  vol = 190.802
core size 113 of 198  vol = 172.654
core size 112 of 198  vol = 158.157
core size 111 of 198  vol = 144.23
core size 110 of 198  vol = 130.907
core size 109 of 198  vol = 117.624
core size 108 of 198  vol = 108.825
core size 107 of 198  vol = 102.367
core size 106 of 198  vol = 95.869
core size 105 of 198  vol = 87.982
core size 104 of 198  vol = 81.415
core size 103 of 198  vol = 74.499
core size 102 of 198  vol = 68.286
core size 101 of 198  vol = 65.785
core size 100 of 198  vol = 62.063
core size 99 of 198  vol = 58.444
core size 98 of 198  vol = 52.671
core size 97 of 198  vol = 47.57
core size 96 of 198  vol = 41.092
core size 95 of 198  vol = 33.66
core size 94 of 198  vol = 24.755
core size 93 of 198  vol = 18.77
core size 92 of 198  vol = 12.639
```

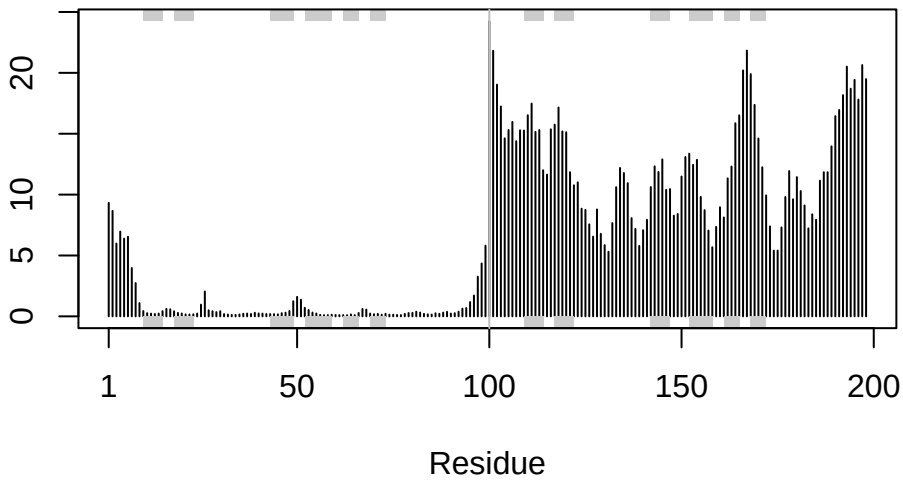
```
 core size 91 of 198  vol = 7.368
 core size 90 of 198  vol = 4.969
 core size 89 of 198  vol = 3.446
 core size 88 of 198  vol = 2.582
 core size 87 of 198  vol = 1.943
 core size 86 of 198  vol = 1.531
 core size 85 of 198  vol = 1.204
 core size 84 of 198  vol = 1.029
 core size 83 of 198  vol = 0.921
 core size 82 of 198  vol = 0.755
 core size 81 of 198  vol = 0.667
 core size 80 of 198  vol = 0.597
 core size 79 of 198  vol = 0.547
 core size 78 of 198  vol = 0.489
 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 relarive domain positions is in graphs of predicted aligned error (PAE) which are included in the downloadable zip file and analyzed in R above.

Predicted Alignment Error for domain.

```
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
```

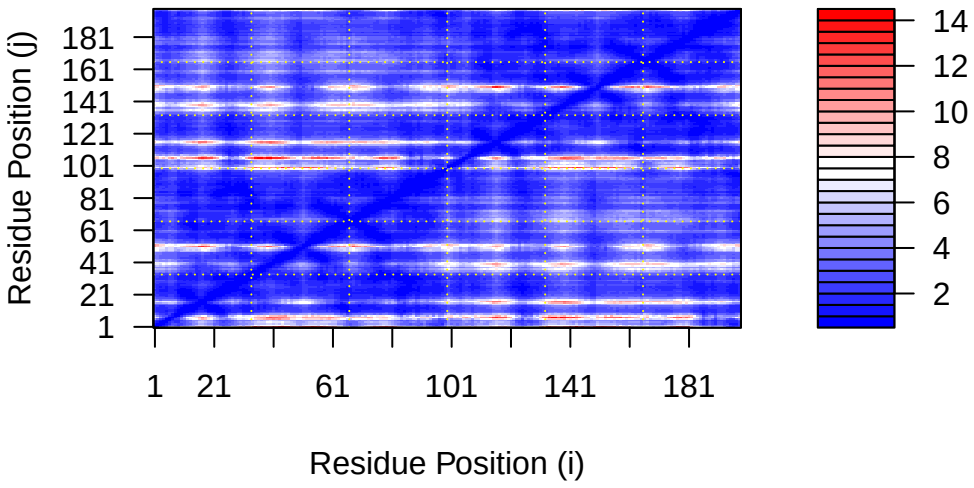
```
pae1$max_pae
```

```
[1] 14.09375
```

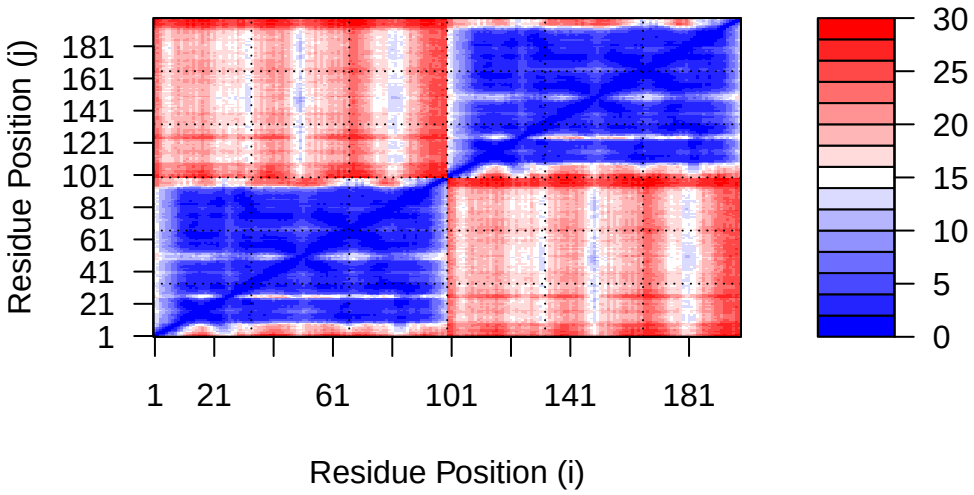
```
pae5$max_pae
```

```
[1] 29.29688
```

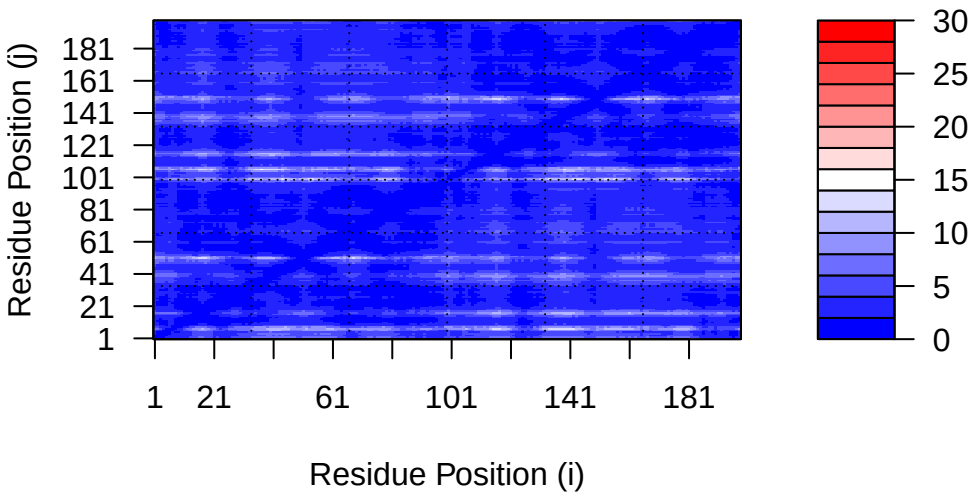
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```



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




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




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

```
[1] "HIVpr2_23119//HIVpr2_23119.a3m"
```

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

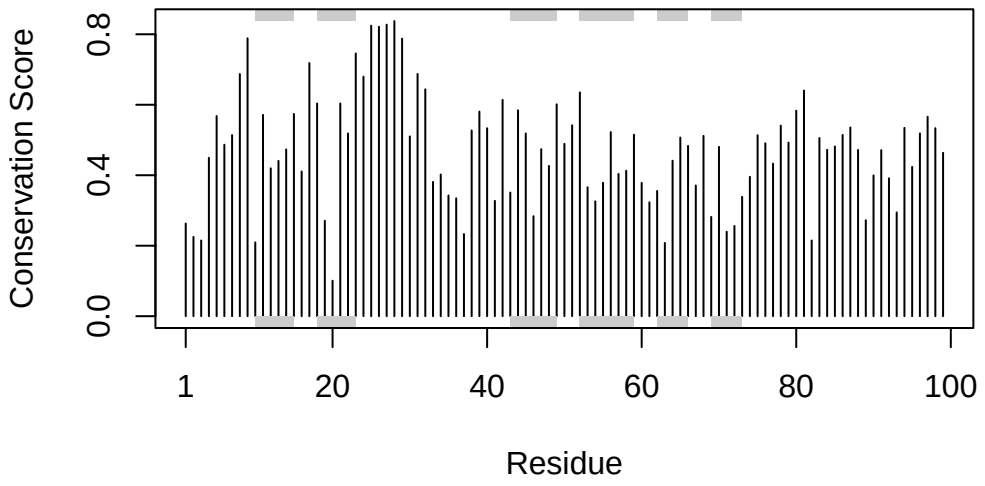
```
[1] " ** Duplicated sequence id's: 101 **"
[2] " ** Duplicated sequence id's: 101 **"
```

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

```
[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")
```