Class 11

Gavin Ambrose PID: A18548522

Table of contents

Background

We saw lat day that the main repository for bio molecular structure (the PDB database) only has ~250,000 entries.

UniprotKB (the main protein sequence database) has over 200 million entries!

AlphaFold

In this hands-on session we will utilize AlphaFold to predict protein structure from sequence (Jumper et al. 2021).

Without the aid of such approaches, it can take years of expensive laboratory work to determine the structure of just one protein. With AlphaFold we can now accurately compute a typical protein structure in as little as ten minutes.

The EBI AlphaFold database

The EBI alphafold database contains lots of computed structure models. It is increasingly likely that the structure you are interested in is already in this database at https://alphafold.ebi.ac.uk/

There are 3 major outputs on AlphaFold:

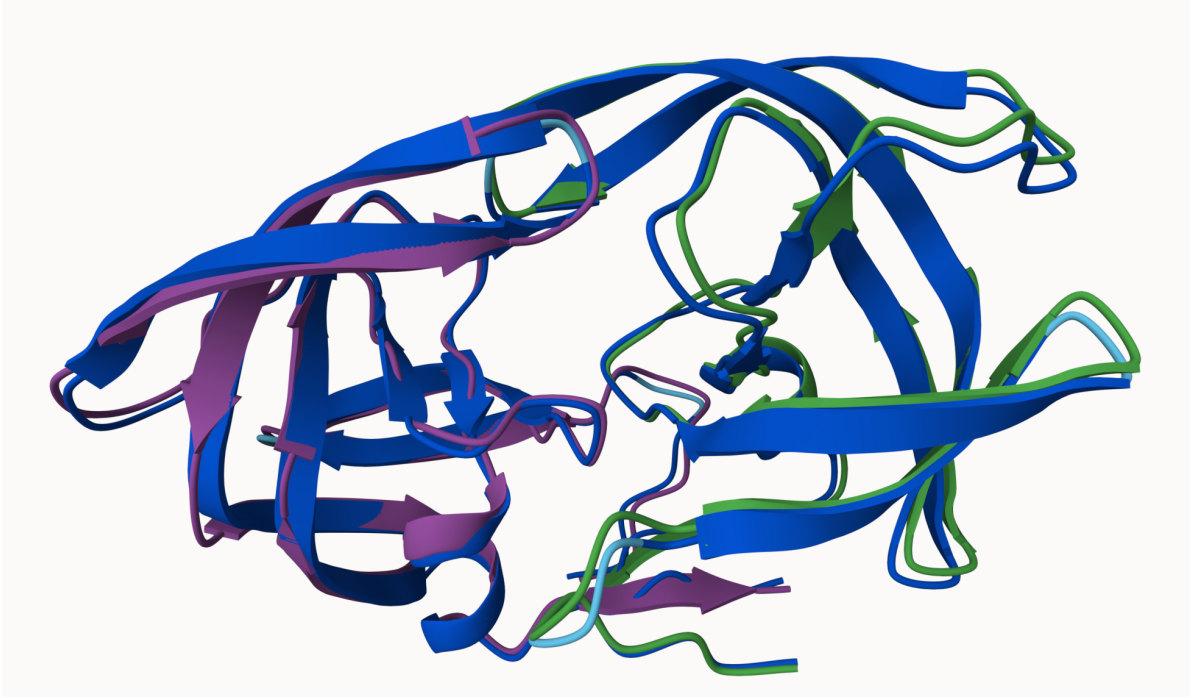
1. A model of structure in **PDB** format
2. a **plDDT score** - that tells us how confident the model is for a given residue in your protein (High values are good; above 70)
3. a **PAE** score that tells the quality of the protein packing.

If you can't find a matching entry for your sequence you are interestin in AFDB, you can run AlphaFold yourself...

Running Alphafold

We will use CollabFoldto run alphafold on our own https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb

Figure from alphafold here!



Interpreting Results

We can read all the alphafold results into R and do more quantitative analysis than just viwing structures in Mol-star

Read all the pdb models

```
library(bio3d)
```

```
Warning: package 'bio3d' was built under R version 4.4.3
```

```
#p <- read.pdb("hivpr_23119.result/")
```

```
pdb_files <- list.files("hivpr_23119.result/hivpr_23119/", pattern = ".pdb", full.names = T)

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

```
Reading PDB files:
hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_4_
hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_1_
hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_5_
hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_
hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_
.....

Extracting sequences

pdb/seq: 1   name: hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_001_alphafold2_m
pdb/seq: 2   name: hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_002_alphafold2_m
pdb/seq: 3   name: hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_003_alphafold2_m
pdb/seq: 4   name: hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_004_alphafold2_m
pdb/seq: 5   name: hivpr_23119.result/hivpr_23119/hivpr_23119_unrelaxed_rank_005_alphafold2_m
```

```
pdbs
```

```
                                 1        .         .         .         50
[Truncated_Name:1]hivpr_2311   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:2]hivpr_2311   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:3]hivpr_2311   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:4]hivpr_2311   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
```

```
[Truncated_Name:5]hivpr_2311   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
                               **************************************************
                               1        .         .         .         .         50

                              51        .         .         .         .         100
[Truncated_Name:1]hivpr_2311   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:2]hivpr_2311   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:3]hivpr_2311   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:4]hivpr_2311   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:5]hivpr_2311   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
                               **************************************************
                              51        .         .         .         .         100

                             101        .         .         .         .         150
[Truncated_Name:1]hivpr_2311   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:2]hivpr_2311   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:3]hivpr_2311   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:4]hivpr_2311   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:5]hivpr_2311   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
                               **************************************************
                             101        .         .         .         .         150

                             151        .         .         .         .       198
[Truncated_Name:1]hivpr_2311   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:2]hivpr_2311   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:3]hivpr_2311   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:4]hivpr_2311   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:5]hivpr_2311   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
```

How similar or different are my models

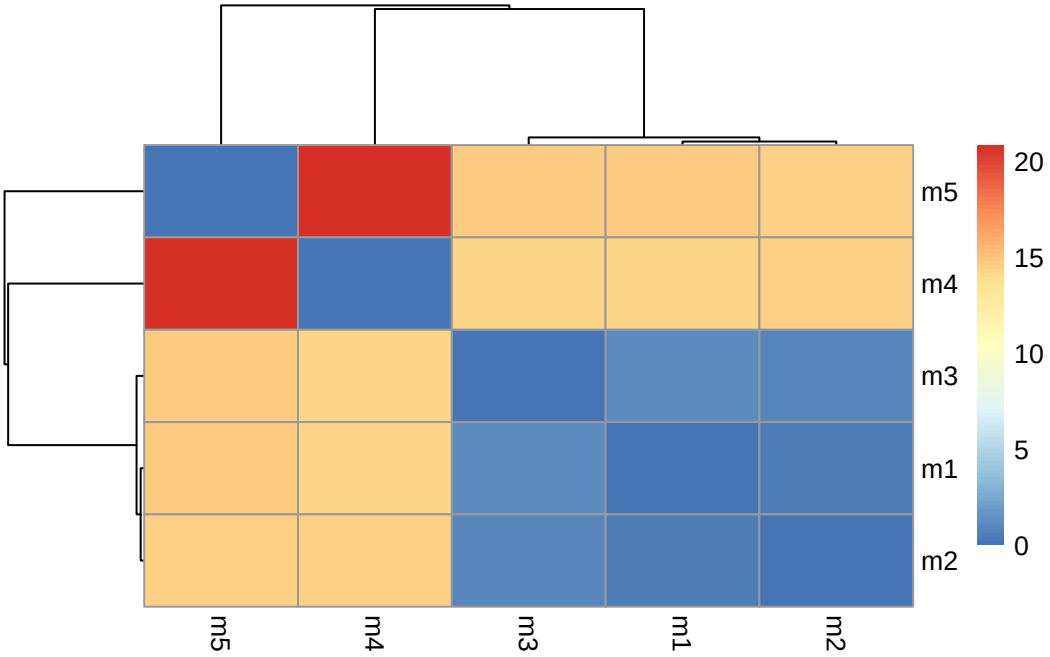
```
rd <- rmsd(pdbs)
```

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

```
library(pheatmap)
```

```
Warning: package 'pheatmap' was built under R version 4.4.3
```

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



Homework is to look at the PAE and plDDT:

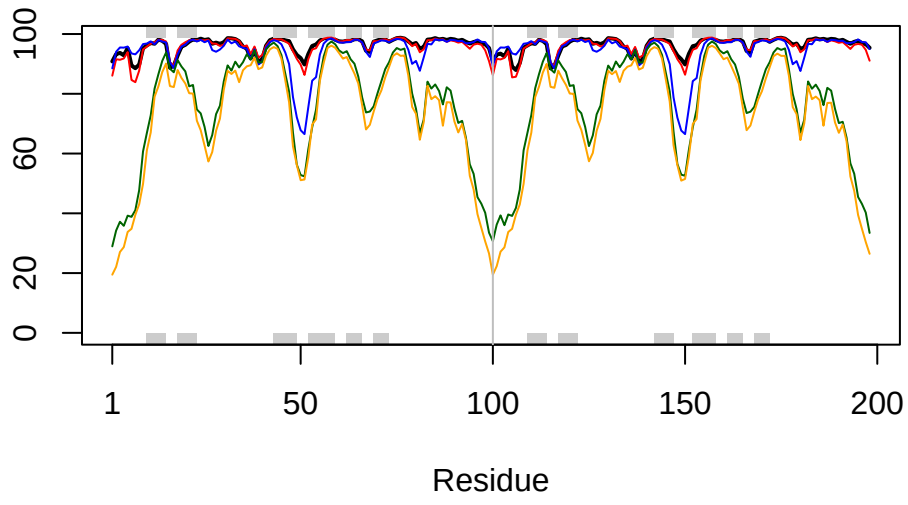
plDDT

Now lets plot the pLDDT values across all models. Recall that this information is in the B-factor column of each model and that this is stored in our aligned pdbs object as pdbs$b with a row per structure/model.

```
# Read a reference PDB structure
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 = 9886.027
 core size 196 of 198  vol = 6910.101
 core size 195 of 198  vol = 1339.391
 core size 194 of 198  vol = 1040.703
 core size 193 of 198  vol = 951.872
 core size 192 of 198  vol = 899.097
 core size 191 of 198  vol = 834.744
```

```
core size 190 of 198  vol = 771.349
core size 189 of 198  vol = 733.076
core size 188 of 198  vol = 697.291
core size 187 of 198  vol = 659.753
core size 186 of 198  vol = 625.285
core size 185 of 198  vol = 589.553
core size 184 of 198  vol = 568.266
core size 183 of 198  vol = 545.027
core size 182 of 198  vol = 512.902
core size 181 of 198  vol = 490.736
core size 180 of 198  vol = 470.28
core size 179 of 198  vol = 450.744
core size 178 of 198  vol = 434.749
core size 177 of 198  vol = 420.351
core size 176 of 198  vol = 406.672
core size 175 of 198  vol = 393.347
core size 174 of 198  vol = 382.408
core size 173 of 198  vol = 372.872
core size 172 of 198  vol = 357.007
core size 171 of 198  vol = 346.581
core size 170 of 198  vol = 337.46
core size 169 of 198  vol = 326.673
core size 168 of 198  vol = 314.965
core size 167 of 198  vol = 304.142
core size 166 of 198  vol = 294.567
core size 165 of 198  vol = 285.663
core size 164 of 198  vol = 278.9
core size 163 of 198  vol = 266.78
core size 162 of 198  vol = 259.01
core size 161 of 198  vol = 247.737
core size 160 of 198  vol = 239.855
core size 159 of 198  vol = 234.978
core size 158 of 198  vol = 230.077
core size 157 of 198  vol = 222
core size 156 of 198  vol = 215.636
core size 155 of 198  vol = 206.808
core size 154 of 198  vol = 196.999
core size 153 of 198  vol = 188.554
core size 152 of 198  vol = 182.276
core size 151 of 198  vol = 176.967
core size 150 of 198  vol = 170.725
core size 149 of 198  vol = 166.134
core size 148 of 198  vol = 159.81
```

```
core size 147 of 198  vol = 153.78
core size 146 of 198  vol = 149.107
core size 145 of 198  vol = 143.672
core size 144 of 198  vol = 137.152
core size 143 of 198  vol = 132.531
core size 142 of 198  vol = 127.244
core size 141 of 198  vol = 121.587
core size 140 of 198  vol = 116.787
core size 139 of 198  vol = 112.583
core size 138 of 198  vol = 108.182
core size 137 of 198  vol = 105.145
core size 136 of 198  vol = 101.261
core size 135 of 198  vol = 97.387
core size 134 of 198  vol = 92.986
core size 133 of 198  vol = 88.195
core size 132 of 198  vol = 84.039
core size 131 of 198  vol = 81.908
core size 130 of 198  vol = 78.029
core size 129 of 198  vol = 75.282
core size 128 of 198  vol = 73.063
core size 127 of 198  vol = 70.706
core size 126 of 198  vol = 68.984
core size 125 of 198  vol = 66.715
core size 124 of 198  vol = 64.384
core size 123 of 198  vol = 61.153
core size 122 of 198  vol = 59.037
core size 121 of 198  vol = 56.633
core size 120 of 198  vol = 54.022
core size 119 of 198  vol = 51.805
core size 118 of 198  vol = 49.652
core size 117 of 198  vol = 48.193
core size 116 of 198  vol = 46.648
core size 115 of 198  vol = 44.752
core size 114 of 198  vol = 43.292
core size 113 of 198  vol = 41.093
core size 112 of 198  vol = 39.147
core size 111 of 198  vol = 36.472
core size 110 of 198  vol = 34.117
core size 109 of 198  vol = 31.47
core size 108 of 198  vol = 29.448
core size 107 of 198  vol = 27.325
core size 106 of 198  vol = 25.822
core size 105 of 198  vol = 24.15
```

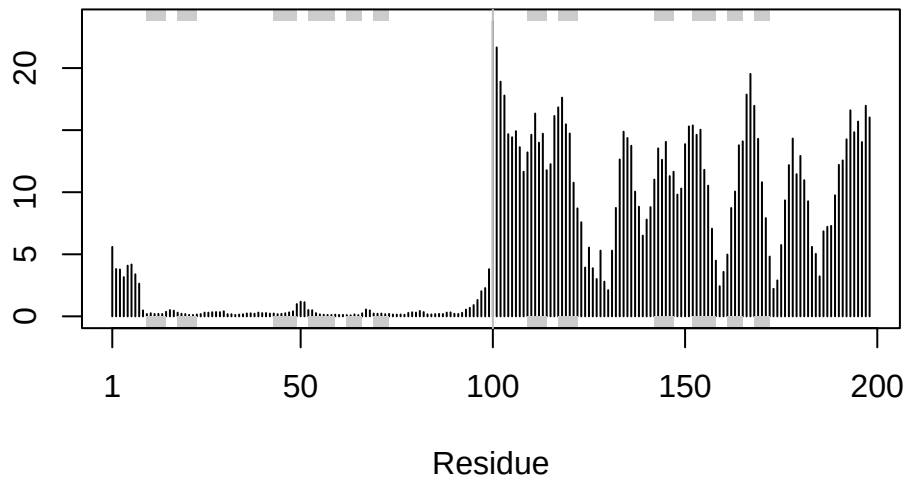
```
 core size 104 of 198  vol = 22.648
 core size 103 of 198  vol = 21.069
 core size 102 of 198  vol = 19.953
 core size 101 of 198  vol = 18.3
 core size 100 of 198  vol = 15.723
 core size 99 of 198  vol = 14.841
 core size 98 of 198  vol = 11.646
 core size 97 of 198  vol = 9.434
 core size 96 of 198  vol = 7.354
 core size 95 of 198  vol = 6.179
 core size 94 of 198  vol = 5.666
 core size 93 of 198  vol = 4.705
 core size 92 of 198  vol = 3.665
 core size 91 of 198  vol = 2.77
 core size 90 of 198  vol = 2.151
 core size 89 of 198  vol = 1.715
 core size 88 of 198  vol = 1.15
 core size 87 of 198  vol = 0.874
 core size 86 of 198  vol = 0.685
 core size 85 of 198  vol = 0.528
 core size 84 of 198  vol = 0.37
 FINISHED: Min vol ( 0.5 ) reached
```

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

```
# 85 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1     9  49     41
2    52  95     44
```

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

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



PAE

```
library(jsonlite)
results_dir <- "hivpr_23119.result/hivpr_23119"

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] 90.81 93.25 93.69 92.88 95.25 89.44
```

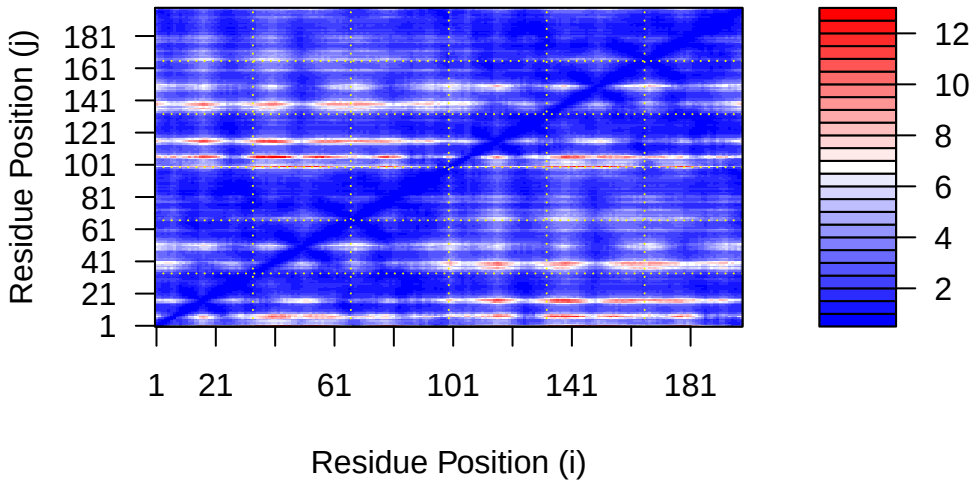
```
pae1$max_pae
```

```
[1] 12.84375
```

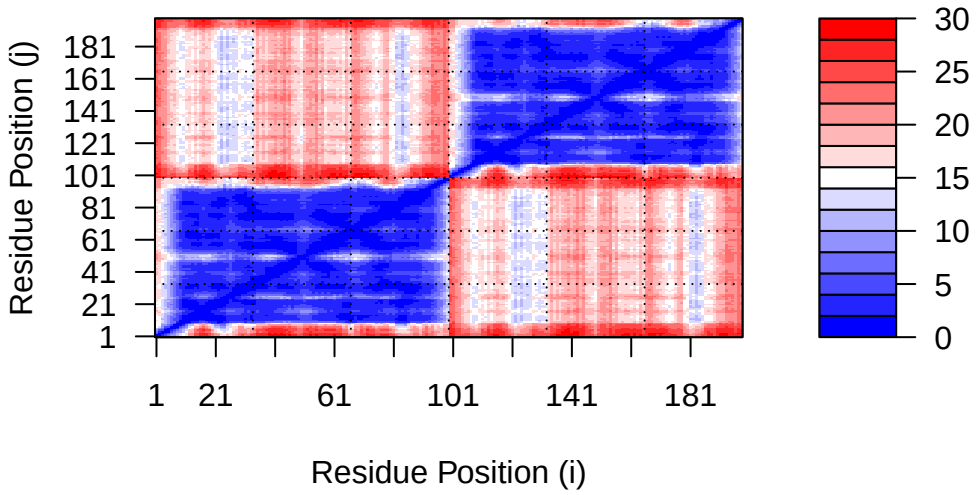
```
pae5$max_pae
```

```
[1] 29.59375
```

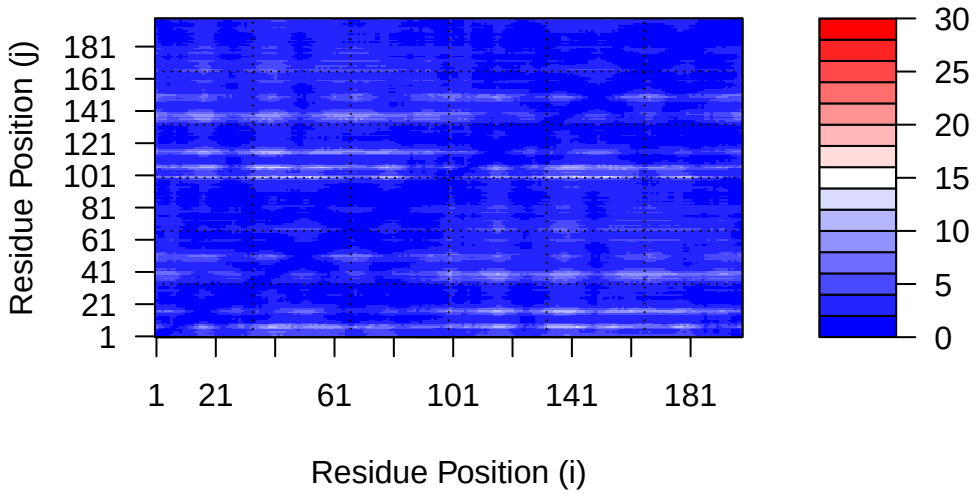
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```




```
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] "hivpr_23119.result/hivpr_23119/hivpr_23119.a3m"
```

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

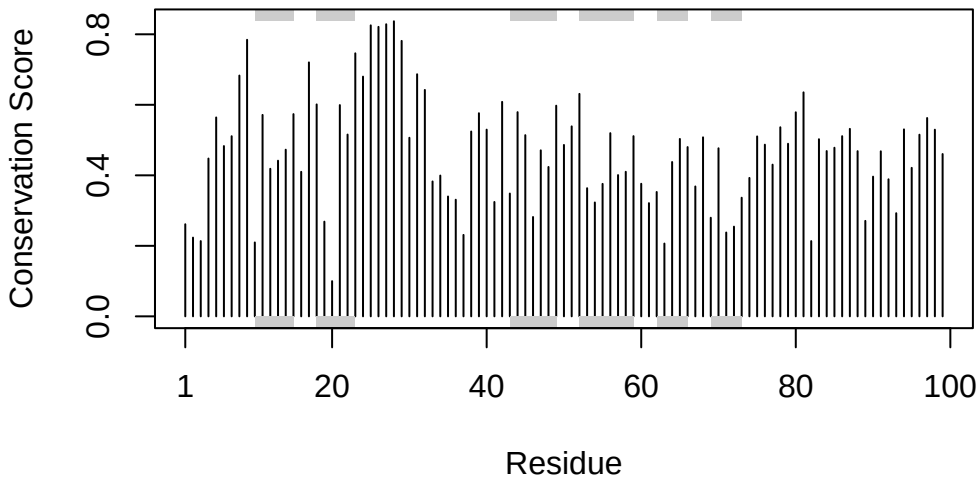
```
[1] " ** Duplicated sequence id's: 101 **"
[2] " ** Duplicated sequence id's: 101 **"
```

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

```
[1] 5397  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")
```