Alpha Fold Analysis

Snehita Vallumchetla (Al6853399)

#Custom analysis of resulting models

Here we will analyze our AlphaFold structure prediction models. The input directory/folder comes form the ColabFolder server:

```
results_dir <- "HHIV_23119"
```

```
# create an object for all of the pdb files in this project space, and filter for them
pdb_files <- list.files(path = results_dir,
                        pattern = "*.pdb",
                        full.names = TRUE)

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

```
[1] "HHIV_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb"
[2] "HHIV_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb"
[3] "HHIV_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb"
[4] "HHIV_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb"
[5] "HHIV_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb"
```

Now we will install the Bio3D package to analyze our HHIV data!

```
library(bio3d)

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

```
Reading PDB files:
HHIV_23119/HHIV_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_seed_000.pdb
HHIV_23119/HHIV_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_seed_000.pdb
HHIV_23119/HHIV_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_seed_000.pdb
```

```
HHIV_23119/HHIV_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_seed_000.pdb
HHIV_23119/HHIV_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_seed_000.pdb
.....

Extracting sequences

pdb/seq: 1   name: HHIV_23119/HHIV_23119_unrelaxed_rank_001_alphafold2_multimer_v3_model_1_se
pdb/seq: 2   name: HHIV_23119/HHIV_23119_unrelaxed_rank_002_alphafold2_multimer_v3_model_5_se
pdb/seq: 3   name: HHIV_23119/HHIV_23119_unrelaxed_rank_003_alphafold2_multimer_v3_model_4_se
pdb/seq: 4   name: HHIV_23119/HHIV_23119_unrelaxed_rank_004_alphafold2_multimer_v3_model_2_se
pdb/seq: 5   name: HHIV_23119/HHIV_23119_unrelaxed_rank_005_alphafold2_multimer_v3_model_3_se
```

```
#let us view what pdbs will output, looks pretty boring, so we will use RSMD to better visual
pdbs
```

```
                               1         .         .         .         .         50
[Truncated_Name:1]HHIV_23119   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:2]HHIV_23119   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:3]HHIV_23119   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:4]HHIV_23119   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
[Truncated_Name:5]HHIV_23119   PQITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGI
                               **************************************************
                               1         .         .         .         .         50

                              51         .         .         .         .         100
[Truncated_Name:1]HHIV_23119   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:2]HHIV_23119   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:3]HHIV_23119   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:4]HHIV_23119   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
[Truncated_Name:5]HHIV_23119   GGFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNFP
                               **************************************************
                              51         .         .         .         .         100

                             101         .         .         .         .         150
[Truncated_Name:1]HHIV_23119   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:2]HHIV_23119   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:3]HHIV_23119   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:4]HHIV_23119   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
[Truncated_Name:5]HHIV_23119   QITLWQRPLVTIKIGGQLKEALLDTGADDTVLEEMSLPGRWKPKMIGGIG
                               **************************************************
                             101         .         .         .         .         150
                             151       .         .         .         .      198
[Truncated_Name:1]HHIV_23119   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:2]HHIV_23119   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:3]HHIV_23119   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:4]HHIV_23119   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
[Truncated_Name:5]HHIV_23119   GFIKVRQYDQILIEICGHKAIGTVLVGPTPVNIIGRNLLTQIGCTLNF
                               ************************************************
                             151       .         .         .         .      198

Call:
  pdbaln(files = pdb_files, fit = T, 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
```

Let use the `rmsd()` function which will allow us to calculate the RMSD values between all the pairs model!

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

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

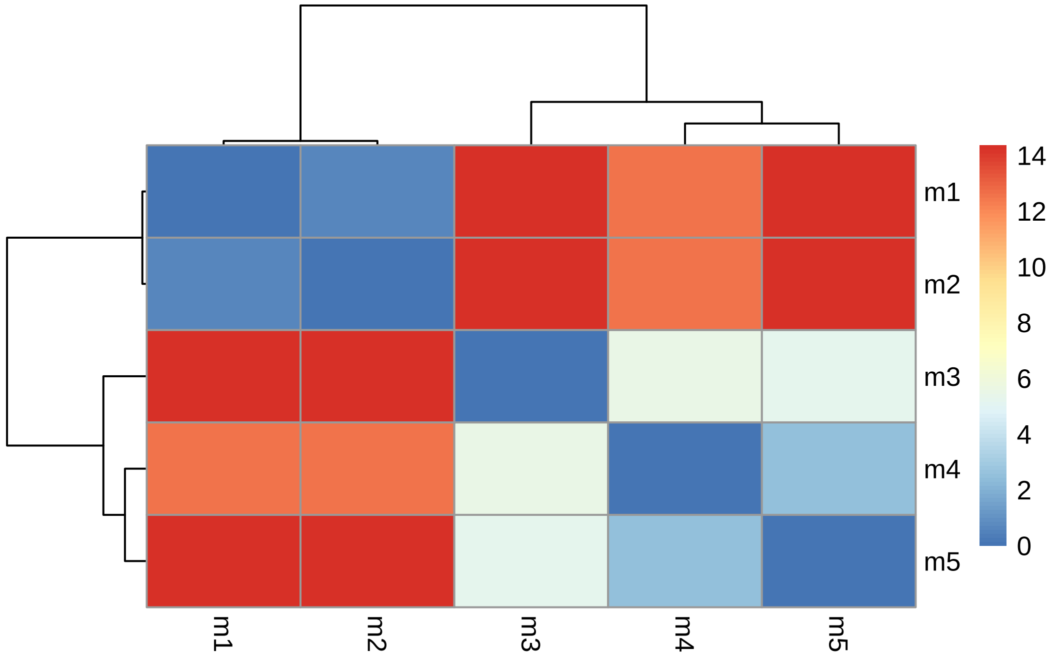
```r
range(rd)
```

```
[1]  0.000 14.361
```

Let us now make a heatmap of the RMSD values:

```r
library(pheatmap)

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

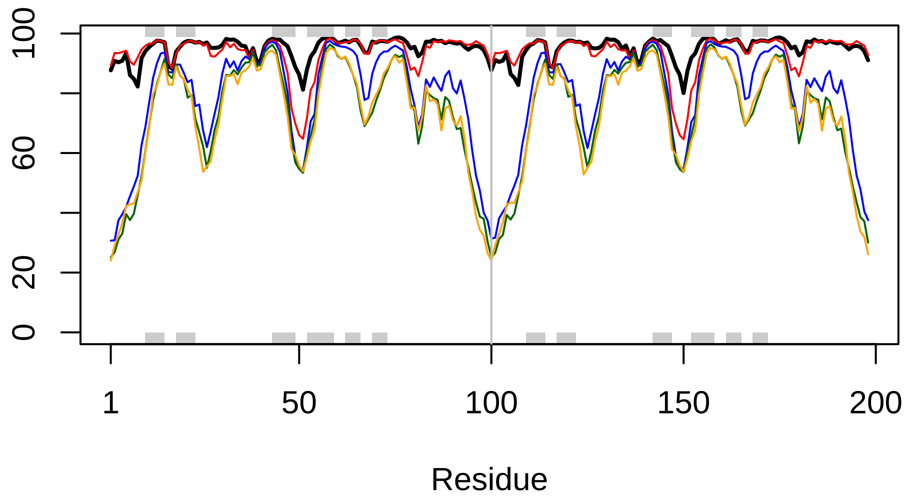


Reading in a reference PDB structure

```
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")
```



We can improve this model by finding the core/rigid core that is common between all of the models:

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

```
 core size 197 of 198  vol = 4922.84
 core size 196 of 198  vol = 4313.857
 core size 195 of 198  vol = 4106.08
 core size 194 of 198  vol = 3900.782
 core size 193 of 198  vol = 3716.54
 core size 192 of 198  vol = 3536.496
 core size 191 of 198  vol = 3429.534
 core size 190 of 198  vol = 3311.647
 core size 189 of 198  vol = 3225.074
 core size 188 of 198  vol = 3152.333
 core size 187 of 198  vol = 3072.545
 core size 186 of 198  vol = 3002.633
 core size 185 of 198  vol = 2961.727
 core size 184 of 198  vol = 2917.627
 core size 183 of 198  vol = 2885.566
 core size 182 of 198  vol = 2852.644
 core size 181 of 198  vol = 2861.997
 core size 180 of 198  vol = 2914.742
```

```
core size 179 of 198  vol = 2971.668
core size 178 of 198  vol = 3005.304
core size 177 of 198  vol = 3040.64
core size 176 of 198  vol = 3075.337
core size 175 of 198  vol = 3100.223
core size 174 of 198  vol = 3124.446
core size 173 of 198  vol = 3133.315
core size 172 of 198  vol = 3091.11
core size 171 of 198  vol = 3033.785
core size 170 of 198  vol = 2945.289
core size 169 of 198  vol = 2883.995
core size 168 of 198  vol = 2825.968
core size 167 of 198  vol = 2741.87
core size 166 of 198  vol = 2666.217
core size 165 of 198  vol = 2595.745
core size 164 of 198  vol = 2529.347
core size 163 of 198  vol = 2458.67
core size 162 of 198  vol = 2384.543
core size 161 of 198  vol = 2318.879
core size 160 of 198  vol = 2231.728
core size 159 of 198  vol = 2155
core size 158 of 198  vol = 2071.293
core size 157 of 198  vol = 1997.488
core size 156 of 198  vol = 1935.278
core size 155 of 198  vol = 1855.378
core size 154 of 198  vol = 1774.751
core size 153 of 198  vol = 1692.853
core size 152 of 198  vol = 1617.43
core size 151 of 198  vol = 1541.23
core size 150 of 198  vol = 1466.916
core size 149 of 198  vol = 1406.176
core size 148 of 198  vol = 1344.36
core size 147 of 198  vol = 1287.433
core size 146 of 198  vol = 1239.608
core size 145 of 198  vol = 1196.508
core size 144 of 198  vol = 1156.067
core size 143 of 198  vol = 1102.579
core size 142 of 198  vol = 1055.75
core size 141 of 198  vol = 1019.539
core size 140 of 198  vol = 977.252
core size 139 of 198  vol = 935.522
core size 138 of 198  vol = 886.34
core size 137 of 198  vol = 843.442
core size 136 of 198  vol = 817.741
core size 135 of 198  vol = 787.008
core size 134 of 198  vol = 757.163
core size 133 of 198  vol = 734.269
core size 132 of 198  vol = 705.121
core size 131 of 198  vol = 670.492
core size 130 of 198  vol = 632.368
core size 129 of 198  vol = 599.159
core size 128 of 198  vol = 559.582
core size 127 of 198  vol = 524.938
core size 126 of 198  vol = 492.781
core size 125 of 198  vol = 466.874
core size 124 of 198  vol = 434.022
core size 123 of 198  vol = 405.499
core size 122 of 198  vol = 396.118
core size 121 of 198  vol = 388.098
core size 120 of 198  vol = 370.742
core size 119 of 198  vol = 342.169
core size 118 of 198  vol = 319.401
core size 117 of 198  vol = 294.51
core size 116 of 198  vol = 267.29
core size 115 of 198  vol = 249.087
core size 114 of 198  vol = 227.847
core size 113 of 198  vol = 209.488
core size 112 of 198  vol = 187.703
core size 111 of 198  vol = 167.142
core size 110 of 198  vol = 152.52
core size 109 of 198  vol = 139.574
core size 108 of 198  vol = 129.304
core size 107 of 198  vol = 116.792
core size 106 of 198  vol = 104.543
core size 105 of 198  vol = 96.677
core size 104 of 198  vol = 89.198
core size 103 of 198  vol = 82.087
core size 102 of 198  vol = 73.854
core size 101 of 198  vol = 67.997
core size 100 of 198  vol = 64.3
core size 99 of 198  vol = 58.842
core size 98 of 198  vol = 52.074
core size 97 of 198  vol = 46.78
core size 96 of 198  vol = 41.964
core size 95 of 198  vol = 35.757
core size 94 of 198  vol = 30.817
```

```
 core size 93 of 198  vol = 23.114
 core size 92 of 198  vol = 16.702
 core size 91 of 198  vol = 9.431
 core size 90 of 198  vol = 4.606
 core size 89 of 198  vol = 3.121
 core size 88 of 198  vol = 2.639
 core size 87 of 198  vol = 2.248
 core size 86 of 198  vol = 1.905
 core size 85 of 198  vol = 1.551
 core size 84 of 198  vol = 1.279
 core size 83 of 198  vol = 1.059
 core size 82 of 198  vol = 0.893
 core size 81 of 198  vol = 0.753
 core size 80 of 198  vol = 0.638
 core size 79 of 198  vol = 0.589
 core size 78 of 198  vol = 0.525
 core size 77 of 198  vol = 0.483
 FINISHED: Min vol ( 0.5 ) reached
```

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

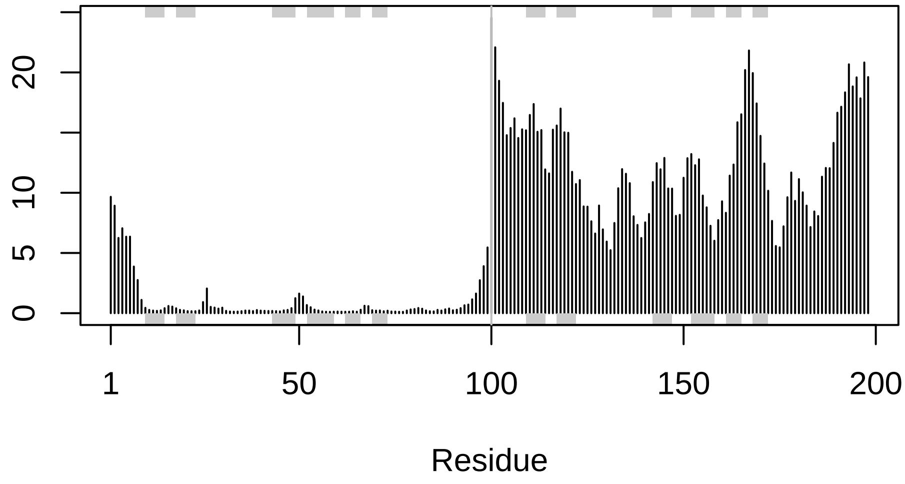
```
# 78 positions (cumulative volume <= 0.5 Angstrom^3)
  start end length
1    10  25     16
2    28  48     21
3    53  93     41
```

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

Let's examine the RMSF between positions of the structure, showing conformational variance in the structure.

```r
rf <- rmsf(xyz)

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



#Predicted alignment for error domains:

Prediction alignment for model 1

```
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"
```

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

```
[1] 87.69 90.81 90.38 90.88 93.44 86.06
```

Let’s look at the maxPAE scores for pae1 and pae5:

```
pae1$max_pae
```

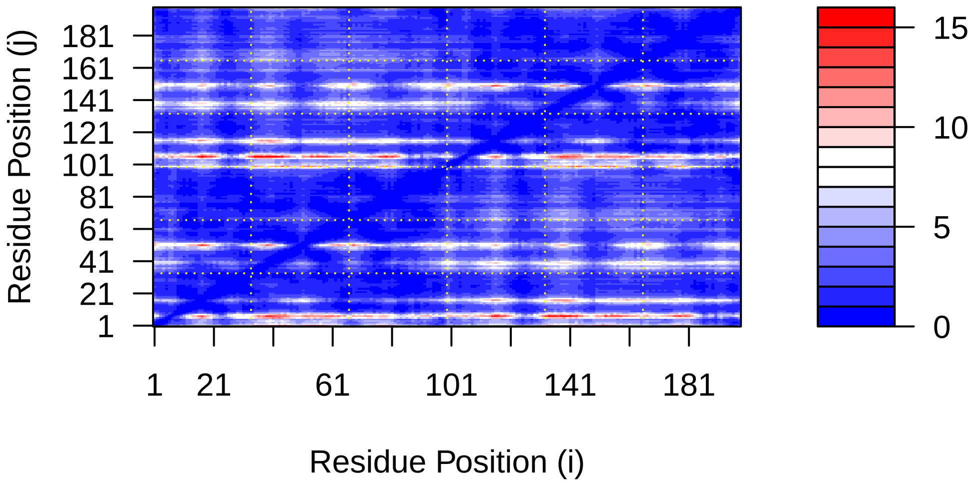
```
[1] 15.47656
```

```
pae5$max_pae
```

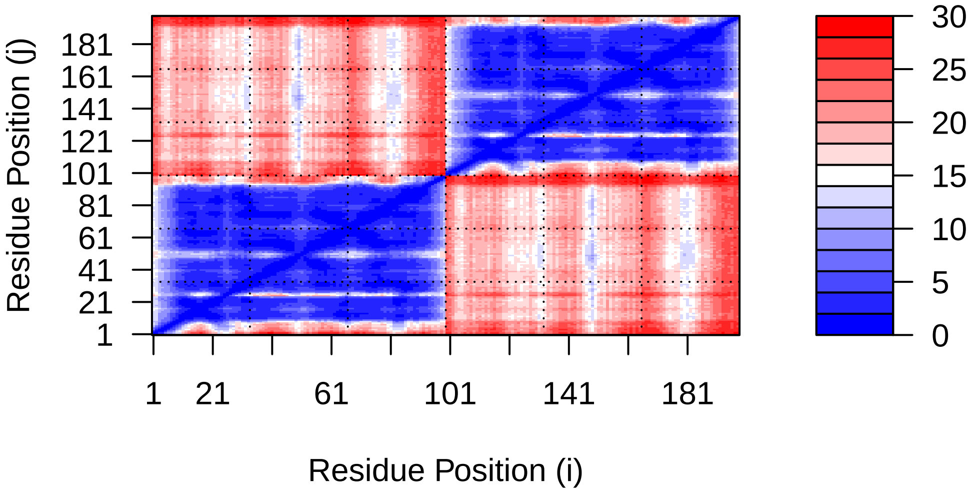
```
[1] 29.32812
```

Plotting the N by N PAE scores using functions from the Bio3D packages:

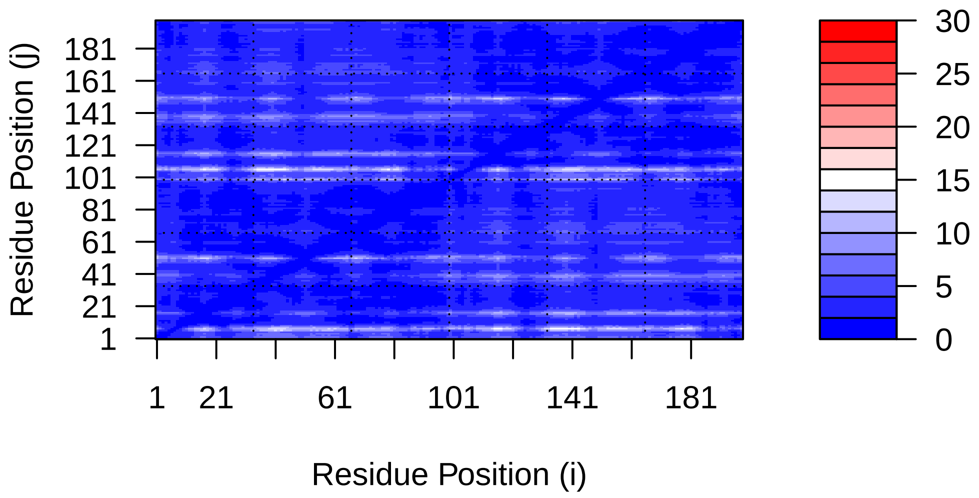
```
plot.dmat(pae1$pae,
          xlab="Residue Position (i)",
          ylab="Residue Position (j)")
```



```
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] "HHIV_23119/HHIV_23119.a3m"
```

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

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

how many sequences are in the alignment:

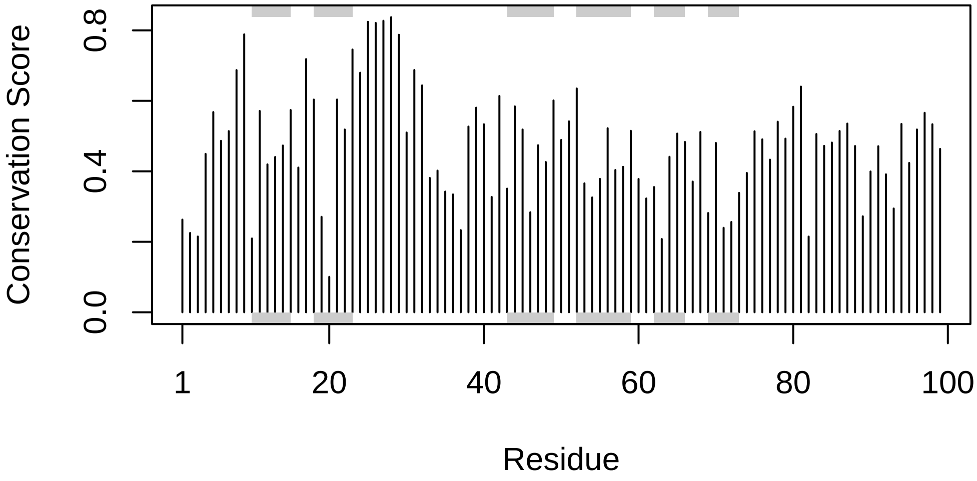
```
dim(aln$ali)
```

```
[1] 5378  132
```

using the `residue()` function to score the 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] "-" "-" "-" "-" "-" "-"
```

```
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")
```